SECTION 231-045-215
Issue 1, September 1977

AUDITS

SOFTWARE SUBSYSTEM DESCRIPTION

2-WIRE NO. 1 AND NO. 1A ELECTRONIC SWITCHING SYSTEMS

CONTENTS PAGE

CONTENTS PAGE

1. GENERAL

INTRODUCTION

1.01 This secton provides an operational description of the Audit Software Subsystem operating in a No. 1/1A Electronic Switching System (ESS) central office.

1.02 When this section is reissued, the reason for reissue will be given in this paragraph.

1.03 Part 5 of this document provides a list of abbreviations and acronyms as used herein.

PURPOSE OF THE AUDIT SUBSYSTEM

1.04 The audit subsystem (AS) provides a method of protecting the ESS from the effects of errors in temporary and permanently stored data (Fig. 1). The AS also provides a means of system initialization or reinitialization.

SCOPE OF SECTION

1.05 This section provides an operational description of the AS at the software subsystem and control program level. A large part of the information applies to both the No. 1 and No. 1A ESS systems. However, differences do exist: information unique to No. 1A ESS is noted as such; information unique to No. 1 ESS is not provided.

AUDIT SUBSYSTEM PIDENTS

1.06 Table A provides a PIDENT-program listing cross-reference for the No. 1A ESS audit subsystem PIDENTs.

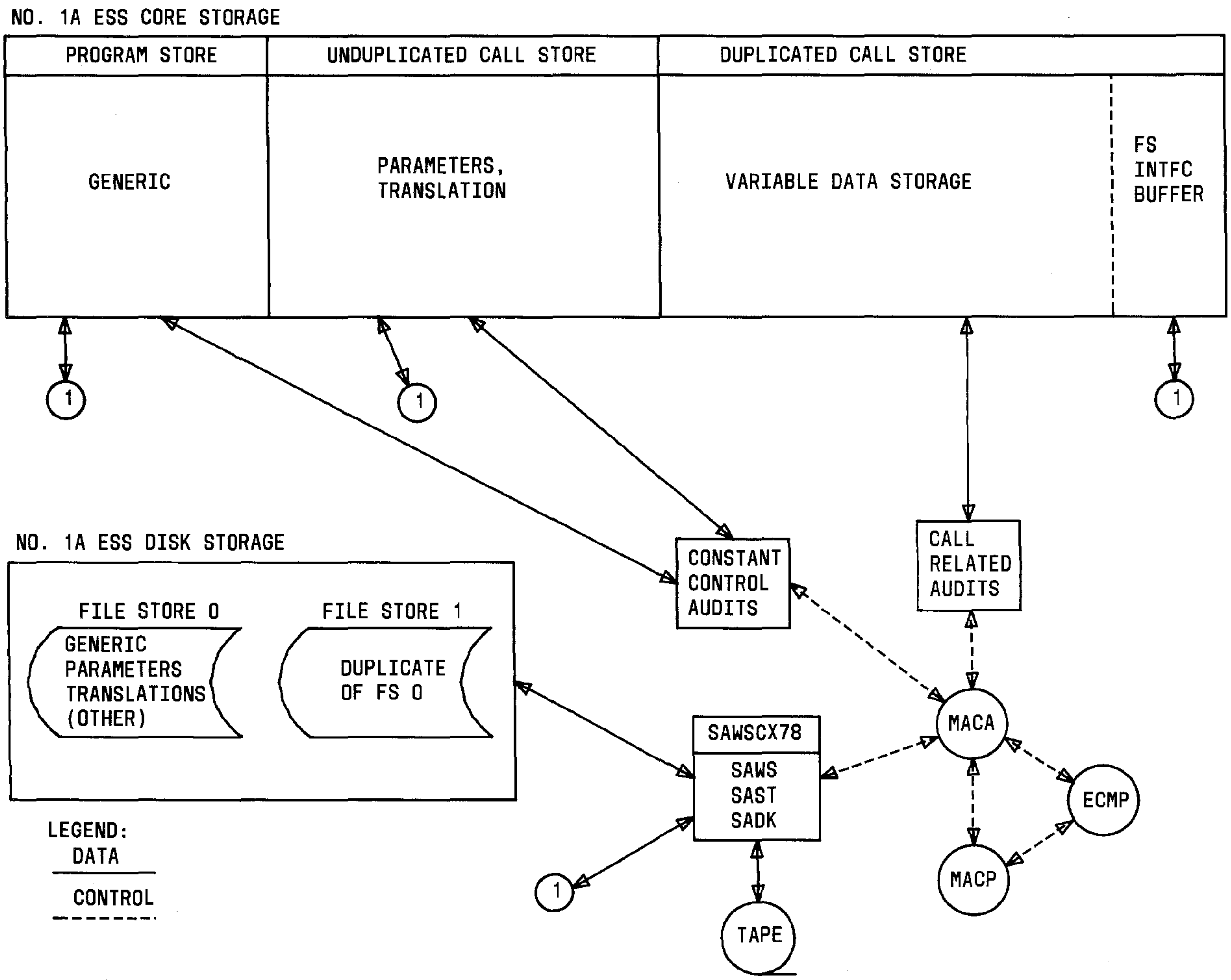

Fig. 1—No. 1A ESS Audit System Access

TABLE A

AUDIT SUBSYSTEM PIDENTS

PIDENT	TITLE	AUDIT NUMBERS	NO. 1 PR/PD-	NO. 1A PR-
MACA	No. 1A ESS Audit Scheduler		—	6A221
NMDT	Line Bit Audit	50	1A012	6A012
PØMC	PØB Audit	11	1A012	6A012
SARG	Call Register Audit	43, 44, 45	1A012	6A012
SACX	Centrex Register Audit	38	1A012	6A012
SAHØ	Hopper and Fixed Length Queue Audit	40	1A012	6A012
SAMP	Network and Map Audit	60, 61, 62, 63	1A012	6A012
SAQU	Audit Variable Length Queues	41	1A012	6A012
SADT	System Audit Programs	5, 14, 16, 20, 22, 24, 26, 30, 34, 36, 46, 48, 52, 59	1A012	6A102
SANK	Linkage Audit of Junior Registers	42	1A012	6A012
NSUP	Enable Table Maintenance Routines	3	1A044	6A044
SAWS	Writable Store Audit	2, 23, 55	—	5A239
CXSR	Centrex Call Register Audit	43, 44, 45	1A012	6A012
NEGN	Network Head Cell and Junctor List Audit	5, 6	1A012	6A012
SADA	Regenerated Constant Audit	1, 3, 65	1A012	6A012
SACV	Receiver Scan Audit	19	1A012	6A012
SATS	Expanded Enable Audit	—	1A012	6A012
SURT	Ring Trip Supervisory Scan Initialization	18	1A012	6A012
SALT	Translation Audit	4	1A012	6A012
SASU	Supervisory Scan Data Table Audit	7, 8, 9, 10	1A012	6A012
SACT	Cutover Program	64	1A098	6A098
AUDSMDIØ	Common Input/Output Audit	—	—	5A216
TNLS[1]	Trunk List Maintenance	12, 54	1A047	6A047

[1] Refer to Section 231-045-230.

2. AUDIT SUBSYSTEM FUNCTIONAL DESCRIPTION

AUDIT ERROR DETECTION

2.01 The AS is tasked with both ***detecting*** and ***correcting*** (if at all possible) software data structure errors which may occur from a great many sources. In general, an audit consists of an evaluation of a data item, ie, bit(s), word(s), etc, and based on the evaluation (actual value or data credibility), a pass/fail decision is made. If errors are detected, the audit in control or other audits may attempt to correct the faulty data. Data structures which can be audited exhibit one or more of the following properties:

- ***Constant Data***—The simplest auditable memory is that which is known to contain constant information. Errors can be detected by simply comparing the backup or redundant data with the actual data.

- ***Timed Memory***—Certain types of data structures are known to have short holding times relative to the length of a call. Other structures are not allowed to be in a transient state for a long period of time. Therefore, it is possible to monitor these structures and if a certain threshold time limit is exceeded, the memory is deemed to be in error. The outpulsing register is a good example of a timeable piece of software since no call should be in the outpulsing state for more than a few seconds.

- ***Redundancy***—If a data structure contains redundancy, then all additional pieces of information may be compared by an audit to determine if they agree. If total agreement is not found, then the data structure is assumed to be in error. In many cases this redundancy is not essential to the normal processing of the data structure and must be added at a considerable expense of real time or memory to provide auditability.

CORRECTION STRATEGY

2.02 On detection of an error in the constant data, an audit will correct the erroneous information from the backup data. An error in a more sophisticated data structure or a time-out will cause all identifiable memory associated with the structure to be idled and maintenance to be requested on all identifiable hardware. In general terms, then, no attempt is made to return the incorrect structure to its proper operational state. Any incorrect guess by the audit as to the proper busy state could be fatal to system sanity. This strategy was chosen as the safest way to guarantee a valid software state.

2.03 In some rare cases where there is an overwhelming N-to-1 vote for the state of a piece of software, the one dissenting piece of information will be changed to agree with the majority.

2.04 After the error is cleared, an audit message is printed on the TTY describing the error. Other audit messages will be printed for each additional piece of equipment restored or memory idled. These messages are printed to notify the craftsman that an error has been detected and are to be used in correcting the source of the error.

2.05 Finally, other audits are requested at high priority to examine all memory functionally related to the data structure in error.

AUDIT SUBSYSTEM REQUIREMENTS

2.06 Audit subsystem requirements pertain to all audits. Due to the differences between the No. 1 and 1A systems, certain system modifications have been made even though they necessitated changes in the translated program.

A. Audit Segments

2.07 An audit segment is defined to be that amount of base level time in which an audit runs without giving up control or taking a time break. The actual time allotted is a theoretical figure for maintenance programs determined by systems engineering. This figure is considered to be the amount of time in which base level call processing can be suspended without unduly influencing the system. In No. 1 the nominal segment length is 10 ms with an actual segment length being probably 12 or 13 ms due to the inaccurate timing methods used. The audit segment length in No. 1A is 2.4 ms. This figure was obtained by taking the 10 ms available in No. 1 and dividing it by the 4 to 1 speedup factor in No. 1A.

Short Segment

2.08 A short segment is an untimed segment that is known to run less than 2.4 ms under error-free conditions. A short segment audit usually has a very simple job whose execution time can be computed by counting cycles.

Timed Segment

2.09 This segment uses a timing method external to the audit to determine elapsed segment time. Once the proper segment time has passed, a flag is set by the external timing program. This flag is consulted by the audit at convenient points in the program where a real-time break would be allowed. If the flag is set, then a segment break is taken. Routines are provided in the PIDENT SADT (System Audit) to do this timing for the client audit. In No. 1 the 10-ms timing was done by counting J-level interrupts; however, it would be difficult to time the 2.4 ms for No. 1A using a 5-ms interrupt as a base. Therefore, the timing for No. 1A is done using the maintenance timer. Since this timer measures elapsed real time and J level consumes 30 to 50 percent of that time, the timer must be set to a value equal to 2.4 ms plus the time consumed by the average J. By selecting an average J of 2 ms or 40 percent of all time, the audit will get less time at peak traffic and more time in an idle system, thus making segment times somewhat dynamic in response to load. The SADT timing subroutines set the clock with the G-level interrupt inhibited so that time-out can be detected by simply reading the clock to determine if it is zero. It is not necessary to zero the clock at the end of an audit since time-out causes no system action.

Interject Long Segments

2.10 Audits that must run for long periods of time without a real-time break must run an interject long segment. This is necessary because certain call configurations cannot tolerate delays of this length without base level action. The interject segment warns the system not to start any delay sensitive jobs and then waits six peripheral order buffer (POB) cycles to allow any delay sensitive jobs that have been started to complete. The audit runs its long segment in interject and then takes five "do nothing" segments to allow the system to recover call processing ability. During the POB cycle wait period the normal maintenance control program (MAC) entry is shut off.

B. Base Level Scratch

2.11 The AS is allocated a large block of scratch area in call store (CS) extending from MA2ASCP to M4GS (No. 1A ESS only).

C. Audit Holding Time

2.12 Audit holding time is that interval of real time in which an audit is holding the MAC scratch pad. Audit cycle time is the sum of the holding times for all audits in the system. It is generally measured as the time between one pass of audit 30 and the next pass. This ensures that all audits have been run at least once. Determination of exact audit cycle and phase times is a very complex process depending on a large number of variables. Better figures are obtainable by manual timing in the field under actual conditions and even these results can vary widely from office to office, or within a single office at different times during the day. Keeping this in mind, the rest of this section attempts to formulate some guidelines for predicting audit times for a new office by comparing its parameters to base offices whose audit times are known.

Audit Cycle Time

2.13 Audits are run primarily out of MACRFILL, ie, as fill work when the main program has nothing else to do. The scheduled routine maintenance entry to audits is from main program job class E, a relatively infrequent entry as compared to the fill entry. Since most auditing is done in spare real-time, and spare real-time is a function of traffic in the system, then audit cycle time (ie, time required for one complete pass of all routinely scheduled audits) increases with traffic in the system.

Traffic

2.14 Individual audit holding times are also a function of traffic. Exact relationships are not clear since each audit responds differently to traffic stimulation. For instance, the map audit takes only a few cycles to process an idle path, but a few thousand to process a busy path. On the other hand a constant audit is unaffected by traffic, and an idle link list audit is inversely proportional to traffic since there are fewer idle structures to process during busy periods.

Number of Networks

2.15 Audit cycle time is almost directly proportional to the number of networks in an office. Increasing the number of networks also increases the number of trunks and lines to be audited, and it indirectly increases the amount of auxiliary memory to be audited, such as registers, POBs, and number of call stores. By making audit cycle time a function of the size of an office, as expressed by the number of networks, it is obvious that a large office has more audit work than a small office.

D. Phase Times

2.16 Phase times (refer to 2.20) are responsive to many of the same factors that govern audit cycle times but in different degress. Obviously available real-time is not a factor in phases because stitched audits do not run out of class E or MACRFILL and do not take real-time breaks for call processing. Traffic is not a variable in phases 1, 2, or 6 but is important in a phase 4 where additional work must be done for each busy or transient connection. The number of networks is the primary factor governing the length of phase 2 with the phase 2 time being almost directly related to network size. The length of phases 4 and 6 seems to be constant and not a factor of office size.

E. Audit Scheduling

2.17 Audits are run by MAC on a demand basis if software errors have been detected in the system or if a teletypewriter (TTY) audit request has been made. The audits are run on a routine basis (class E main program entry to MAC) if no higher priority maintenance work exists in the system. This routine mode serves as the systems chief protective defense against unknown software errors. Additionally, MACRFILL runs audits as fill work when the main program job scheduler has available real-time with no scheduled work to do. All other audit functions, such as phases, demand audits, and interrupt level audits, are after the fact and serve to clean up known errors.

Request Tables

2.18 In No. 1A ESS, audits are requested out of five request tables (audit control blocks):

(a) Phase (runs audits in a phase)

(b) Demand high priority

(c) Demand low priority

(d) Routine high priority

(e) Routine low priority

Each request table consists of three call store words. Each bit in the request table corresponds to a single one of all the possible audits. Requests are served by scanning the tables in order of priority (highest to lowest) from right to left until a 1 is detected. Once a 1 has been found, its corresponding bit position is zeroed in all tables to clear any other requests for that audit, and its bit position is used to index a vector table to run the correct audit. The priority structure ensures that certain critical audits will have execution priority over the last critical ones.

Routine Scheduling

2.19 Audits running out of MACRFILL get a smaller and smaller percentage of real time as traffic increases. This design provides a valuable dynamic audit response to load conditions. The system is simply trading some of its audit protection for increased call capacity during times when extra call capacity is needed. This technique is useful, but it has the disadvantages of providing the least protection at a time when the probability of errors and vulnerability of the system is greatest. Conversely, it provides the most protection in an idle system where the error rate is very low. In addition, relying solely on the class E visits to audits greatly increases the audit cycle time while the higher traffic causes a greatly increased error propagation rate. These two diverging trends cause a consequent decrease in the probability that the audit needed to detect a specific error will run within a short enough time span of the occurrences of the error to prevent further call degradation, interrupts, or phases. This routine protection function is also shared with the less detailed and faster checks of the data validation. The data validation, since it has a fixed entry rate, is not subject to traffic variations and is thus more likely to detect catastrophic errors during peak traffic situations.

F. Audit Acitivity During a Phase

2.20 During a system reinitialization or "phase," audits are executed in a "stitch" mode; ie, the audits comprising the phase are stitched together one after another with no real-time breaks taken. All other cental control (CC) processing activities are suspended until the phase is completed. (Refer to TOP 231-369-001.)

Audit Output Messages

2.21 The AS provides a set of SA03 type output messages (OMs) to keep the office maintenance personnel aware of AS activities. (Refer to OM-6A001.)

3. PIDENT DESCRIPTIONS

GENERAL

3.01 The AS consists of a collection of PIDENTs which form a set of audits and a control structure. Most of No. 1A ESS audit PIDENTs are converted versions of the No. 1 ESS audit PIDENTs. As such, the following descriptions, although oriented toward No. 1A ESS, generally apply to No. 1 ESS also. Exceptions are so noted.

AUDIT PIDENTS

A. NMDT

3.02 The line bit audit (NMDT) ensures that lines do not lose supervision, ie., having line bits in an invalid state. In order to verify that a line which is marked busy should be marked busy, it might be necessary to do a very long search. First, all the path memory for lines (PML) words for the line network would have to be examined for having this line in a path. If not found there, the line could be on the high-and-wet list, permanent signal queue, blocked dial tone queue, customer DP receiver queue, TOUCH-TONE® queue, or in one of two different states in a disconnect register. Finally, the line could be involved in a dial tone speed test. Since this is a very time-consuming search, it cannot be done on a per-line basis. Instead, the same areas are examined but an image table is formed for many line bits simultaneously.

3.03 When the line bit audit is first entered, checks are made to determine if the normal audit can be done. In emergency action (EA) phases other than 4 or 5 only, the no-test vertical check is done because the audit would have no way to restore the ferrod to a line, if the line bit was in error. In EA phases 3, 4, and 5 ferrod restoring is done at the end of the phase to all lines which are marked idle. Hence, in these phases, the audit is run but it must not be run until audit 40 has run if it is to avoid looking into unaudited hopper memory. Therefore, if the low priority audit flag is still set for audit 40 in an EA phase, then the line bit audit re-requests itself at low priority and terminates. When the line bit audit is entered outside an EA phase, it still has to see if it can run. The audit in this case must try to avoid restoring the ferrod to a line which could still be connected through the network but which must be restored to return normal service to the line. To minimize this possibility, which can eventually destroy the line cutoff, the line bit audit will delay itself for up to 256 segments in MAC until both the trunk maintenance list addendum (TMLA) and the junctor maintenance list (JML) are empty. When the audit finds that it can run, each pass of the audit is run as a long segment.

3.04 Each pass of the audit begins by marking all bits idle in an image table. The audit then examines all PML words associated with the line network to be audited. As busy lines are found, their image bits are marked busy. When the audit finds a PML word containing a pseudo line indication, it marks a time-out bit in the PML word. If the time-out bit had already been marked, the PML word is zeroed and an error print is given. The routine SADMPJ is not used because of scratch conflicts, so another error print from an audit finding the zeroed PML word can be expected. When the audit has looked at all the PML words, it then uses the routine NMLIST to search the other areas where lines can be found. As lines are found by this routine, their image bits and line bits are marked busy and, in an office with 2:1 line frames, the mate bit in the line equipment number (LEN) is verified. If the mate bit is incorrect, the LEN is zeroed out wherever it was found and an error print is given. After all areas have been examined, the image bits are matched with their corresponding line bits. When a mismatch occurs, the line bit is either corrected in an EA phase or the LEN involved is stored in "BLENS." In the latter case, an error message is also printed unless the line involved is an unassigned line. In addition, if the predicted state of the line bit is busy and the audit is not in an EA phase, the LEN is also stored in the CS location BBLEN. A delayed request is made to audit this network if more mismatches occur than room remains to store the LENs involved. When the matching is finished, the audit either segments if not in a phase, or proceeds to the next pass.

3.05 When control is returned to the audit after it segments again, location BBLEN is checked to see if the preceding long segment found a busy bad line. If it did, the audit will change the line bit to busy and trace the line with the routine NETRCL (Call Trace). The result of the trace is used to restore any call register (CR) or junctor linked to this LEN. The result of the trace is also printed in an SA03 message. Independent of BBLEN, the audit restores/verifies the lines that were stored in MA2ASCP. After each line is restored, a direct scan is done on the line. If it is on-hook, the L bit is set to idle, and when the cutover program SACT exists and is active, the cutoff contacts are opened on inactive lines to remove supervision. When it is found off-hook, it is placed on the high-and-wet (HAW) list. When all lines have been restored, the audit segments seven more times to let the system recover some of the time taken by the long segment, and then proceeds to the next pass.

3.06 After the audit has finished checking all the line bits that were requested, it proceeds to a different phase of the audit. This last phase of the audit verifies that no no-test vertical is in a limbo state. Once the no-test verticals are checked, the audit terminates.

B. POMC

3.07 The peripheral order buffer audit program (POMC) functions both in the stitched mode for initialization purposes and as a routine audit. POMC checks for entries in the POB queue. An entry indicates that a client has been queued and is waiting for a POB. If there is an entry, the queue flag should be set. If the flag is not set, POMC zeros the queue head cell. POMC also checks to see if a POB has been held too long by a client. Each time POMC is entered, the time-out flags of all nonidle POBs are set. Before the next scheduled POMC audit entry, the busy POBs should have been idled by their respective clients. If the same POB had been released and then seized again, its time-out flag would have been cleared when the POB was hunted. Each time the audit is entered, each nonidle POB with its time-out flag set is automatically idled. If a client register is associated with a timed-out POB, and if the program tag (PT) is not 0, the register identifier (RI) is set to 0 and the PT is set to 7 (indicating that the client register should be taken down). Should the PT equal 0, the RI and PT are not touched because the path memory is idle.

C. SARG and CXSR

3.08 PIDENTS SARG and CXSR comprise the call register audit. The CR audit performs validity checks on the active CRs, idles those found in trouble, and attempts to idle any equipment associated with the troubled CRs. The audit is divided into three phases with a five-minute interval between the end of phase I and the start of phase II, and a one-minute interval between the end of phase II and the start of phase III. After an EA phase, the five-minute interval is shortened to two minutes.

3.09 The CR audit recognizes three types of CRs (Table B). Group I registers [with the exception of regular ringing registers or originating registers used for local test desk (LTD)] which stay busy on one call for more than five minutes are in trouble, having lost control of the call. Group II registers can be busy with a particular call for extended periods, but cannot be found transient by two inspections one minute apart. Group III registers can be busy on the same call for extended periods of time and display transient or non-steady state conditions, eg, collecting coins, etc.

TABLE B

CALL REGISTERS

NAME	SYMBOL	GROUP 1 TØ	GROUP 2 TØ AND NSS	GROUP 3 TØ, NSS, AND TØA
Originating	ØR	X		
Disconnect	DR	X		
Regular Ringing	RR1	X		
Special Ringing	RR2	X		
Flash Timing (Senior Register)	FL	X		
Reverting Call	RVC	X		
SXS Incoming	ISR	X		
Incoming Trunk Test	ITT	X		
Auto. Idend Out Dial	AIØD	X		
Permanent Signal	PS		X	
Auto. Message Accounting	AMA		X	
Hotel-Motel	HMØ		X	
Temporary Transfer	TPT		X	
Simulated Facilities	SF		X	
Trunk Preemption	TRRG		X	
Operator	ØP			X
Coin Charge	CNC			X
Conference — 3 Port	CF			X
Conference — 6 Port	C6			X
Centrex Loop	LP			X
Supplementary DDD*	DDS		X	
Basic DDD*	DDB			

* The direct distance dialing service observing registers are specially audited. The supplementary register is audited as a group 2 type, and the basic is only processed if an error is found in the supplementary register.

3.10 Each CR can have up to three bits, depending on its group, reserved for audit usage. All registers have a Y4TO time-out bit in bit 8 of the register state word. This bit is maintained by the CR audit and release register routines. Y4TO is set (to 1) in all busy registers during phase I of the CR audit. It is cleared (to 0) by the release register routines when the register is idled as part of normal call processing and by the CR audit phase II or III. A nonsteady-state activity bit Y4NSS in bit 16 of the state word is provided for registers in groups II and III. This bit is set by the call program when the register is in a nonsteady-state condition and reset when the condition terminates. Group III registers have a time-out activity bit, Y4TOA, in bit 15 of the state word. This bit is cleared by the CR audit in phase I and must be set by the call program when the call enters a nonsteady-state condition (Y4NSS=1). Call programs may zero the nonsteady-state bit but must not zero the time-out activity bit.

3.11 In addition to the audit bits described above, an audit code, defined as the combination of register identifier = 0 and program tag = 7 (RI, PT = 0, 7), can be placed in a register. The audit code indicates that discrepancies have been found in associated CS memory. The audit code for a register may also be set by a call program when it encounters an invalid program condition while processing the register and transfers to YRUNEX or SHAEP1. Registers containing the audit code have all reports of disconnect, time-out, and answer deadened by the SADIS5 or SADIS7 subroutines, and will eventually be idled by the CR audit.

3.12 When a bad CR is idled, special checks are made to determine if the register is on a call processing list or queue. If the audit finds it on a queue or list, the register is removed. This avoids linking the call processing list or queue into the register idle list. While searching the list or queue, the audit makes limited sanity checks and destroys the list by zeroing the head cells when the checks fail.

3.13 Certain auditing functions (eg, rebuilding the idle list) performed by the audit require that ***all*** call registers of one type be investigated in one segment. Since large offices may have a large number of CRs of one type, the audit determines the number that actually exists and arranges for doing the auditing in a long segment in interject, in lieu of running the segment in the normal manner. Since the function of checking a busy register can consume varying amounts of time, depending on the particular state of the register, busy register checking is segmented by means of the audit timing control routines in MACS25.

D. SACX

3.14 SACX, the centrex register audit, verifies and updates the information in the centrex input/output (I/O) blocks, centrex console registers, and centrex console group number. SACX also rebuilds the idle console register linked list for each centrex console group and checks the state of each console group queue.

3.15 Upon initial entry, SACX checks first to see if an EA phase 2 or higher is in progress. If so, the CS locations serving as lamp order link list head cells are initialized by the centrex lamp order (CNLP) program. Also, subroutine CXEACI (PIDENT CXMS) is used to initialize the hardware for the data links and consoles. If SACX is entered on a routine basis, the lamp order link lists are audited by routine CNDALL and the verification program CXMSVR (PIDENT CNLP), which verifies the headset status, night service lamp, and position busy lamp on all the centrex consoles, is called.

3.16 Regardless of how the audit was called, it next compares two sets of data in the data link status tables, ie, the out-of-service (OS) bits and the trouble (T) bits. On a mismatch, if the T bit is set, subroutine CXMVIO examines the I/O block and if it decides that the link is out of service, then a main program flag (135) is set to call CXQUCL in the data link diagnostic I/O program (PIDENT CXMC). This routine removes all the lamp blocks in the lamp order link lists for the data link to be taken out of service and sets the OS and diagnostic (D) bits. If the T bit is not set and CXMVIO indicates that the link is OS, the T bit is set. If CXMVIO indicates the link is in service, the OS and T bits are zeroed. The constant data necessary for the remainder of this audit is stored primarily in the centrex data link auxiliary (AUX) blocks and the centrex common blocks, eg, data used to describe the equipment interface for each centrex attendant console that is controlled by that data link.

3.17 The AUX block data also yields the information necessary to scan the data link frame for reception of the key signals and the enables to transmit lamp orders to the remote console terminal. The computed data result is compared to the corresponding present data in the I/O block. If a mismatch is detected, the I/O block word is updated and an audit error print of the CC registers is requested.

3.18 Next, using data found in the I/O block, SACX determines the in- or out-of-service state of each data link. For an in-service link the corresponding bits in the data link status are reset. Processing of an out-of-service link is handled by routine CXMBIO in PIDENT CXMC. For an inconclusive result, the in- or out-of-service decision is based on the state of the T bit and the corresponding I/O block items are made to agree with it. All of the I/O blocks associated with all the data links on the first frame are examined and updated, and then the process is repeated for the other frames in the office. The data link status tables are updated to reflect any frames, data links, and consoles not equipped or assigned.

3.19 Next, the console registers controlled by the data link are audited. AUX block data, such as console group number, centrex group number, etc, are compared to the contents in the console register. If a mismatch is found, the register contents are corrected and an error report is given.

3.20 The routine also ascertains that any console or pseudo-console register, having its maintenance busy bit (MBT) indicator set to 1, is either associated with a data link or a console that is out-of-service. If it is not, the MBT is reset to 0 and the console register is restored to the idle link list. The verification of other console register data includes:

(a) The console number

(b) The audit code

(c) The status of certain lamp groups under control of the console

(d) The correspondence of the loop busy-idle bits to the number of loop circuits assigned to the console and the loop registers linked to the console

(e) The index (PLOP) to identify the call which the attendant is presently controlling.

An error printout is requested whenever any of the above comparisons results in a mismatch. In addition to the error printout, the blind idle control bit is set if the console number, the audit code, the route index (PRI), or the miscellaneous trunk distributor number (MTDN) was incorrect, or if the present call that is being processed by the register is in some invalid state. The audit segments after processing each console register.

3.21 After all console registers and I/O blocks are updated, the next step is to regenerate the linked list of console registers. This section of the audit does not segment and can be regenerated separately if there is dependable information in the registers. This portion of SACX accomplishes the following:

(a) Zeros out all the head cells of the linked list in the console group number (CGN) blocks and the centrex maintenance console control register

(b) Regenerates the linked lists of console registers

(c) Regenerates the blind idle link list of console registers that have their blind idle control bit set

(d) Counts and stores the number of console registers in the group whether or not they are placed on the linked list

(e) Ensures that the link word of the last console register on the list contains zeros

(f) The centrex number (CTXN) stored in the CGN block corresponds to that of the console register. If there is a mismatch, the error printout occurs.

3.22 Console group queues are audited by checking the queue usage count as it appears in the CGN block to be sure that it does not exceed two times the number of consoles. If the maximum is exceeded, an error printout is generated. Additionaly, all loop registers on a queue are checked for address range limits, RI=4YILP, and that the queue number stored in loop subregister 2 is identical to the number of the queue currently being audited. If the first register on the queue is invalid for any of the above reasons, the console group head cell is cleaned up by zeroing the words containing the addresses of the first and last registers on the queue and by zeroing the queue usage count. And, an error message is printed. If any other register is found to be invalid, the address of the last good register is made the end pointer for the queue and the call register audit is requested. All console queues are checked similarly.

3.23 SACX also audits the following:

(a) The console register conference bit, indicating whether the console has attendant conference, is checked against data in the data link auxiliary block.

(b) The console register conference indexes are audited in conjunction with the conference lamp items.

(c) The loop register associated with the conference call is audited as is the associated conference register.

(d) The up-down count for simulated facility registers is audited.

E. SAHO

3.24 The hopper and fixed length queue audit program SAHO verifies that the load and unload pointers for hoppers and fixed length queues (FLQs) are within valid limits and makes corrections when they are not. Also, SAHO makes limited checks on hopper contents and corrects any errors found.

Format of Hoppers and FLQs

3.25 Hoppers are blocks of CS memory used to store various data items for communication between I/O and call processing programs. FLQs are blocks of CS memory which store information on customers unable to be served due to the unavailability of equipment. Since both hoppers and FLQs are organized and processed in similar ways, they are audited by a common program. Blocks of CS are assigned to each hopper and FLQ, and contain an engineered number of data words, followed by two link words.

3.26 Each hopper and FLQ has two words for loading information (load pointer, load bottom pointer) and two words for unloading (unload pointer and unload bottom pointer). The first word of each specifies the point to load (or unload) information; the second word specifies the end of the current hopper or FLQ. Hoppers must always have their load pointers pointing to an all-zero CS word and their unload pointer pointing to a nonzero word, unless the hopper is all zero.

SAHO Operation

3.27 SAHO operates in the normal audit environment with segmentation based on completion of the audit of a single hopper of FLQ.

3.28 Since hoppers are normally loaded by I/O programs running on J-level, it is possible that the load pointers could change while the audit is auditing the hopper. SAHO makes special checks to determine if this has happened before deciding that an error exists. When errors in hoppers are detected, they are corrected with J-levels inhibited, thus avoiding complicated interaction problems. The period of time required to correct hopper errors is sufficiently short to permit this suspension of I/O activity.

3.29 Certain hoppers and FLQs may not be present in particular offices. Accordingly, SAHO requires parameter information to determine the number of two-word hoppers, the total number of hoppers, and the total number of FLQs. The necessary information is stored in parameters. Table C provides a list of the hoppers and FLQs.

TABLE C

HOPPERS AND FIXED LENGTH QUEUES

NAME	TYPE
MULTI-FREQUENCY DIGIT	2-Word Entry Hopper
TOUCH-TONE DIGIT	2-Word Entry Hopper
HIT SCAN RESULT	2-Word Entry Hopper
CENTREX KEY	2-Word Entry Hopper
RING TRIP	1-Word Entry Hopper
ABANDON, DIAL PULSE DIGIT	1-Word Entry Hopper
RELEASE DIAL TONE	1-Word Entry Hopper
REVERTIVE PULSE DIGIT RECEPTION	2-Word Entry Hopper
TRUNK SEIZURE + ANSWER	1-Word Entry Hopper
MISCELLANEOUS SCAN	1-Word Entry Hopper
LINE FERROD	1-Word Entry Hopper
STEP-BY-STEP (SXS) REORDER, ABANDON, DIGIT	1-Word Entry Hopper
TRUNK DIAL PULSE TRANSMISSION	1-Word Entry Hopper
REVERTIVE PULSE DIGIT TRANSMISSION	1-Word Entry Hopper
TRUNK DIAL PULSE RECEPTION	1-Word Entry Hopper
TOUCH-TONE RECEIVER	Fixed Length Queue
CUSTOMER DIAL PULSE RECEIVER	Fixed Length Queue
BLOCKED DIAL TONE	Fixed Length Queue

Range Checks

3.30 The range check for each hopper or FLQ verifies that the load and unload pointers contain addresses which fall within allowable address limits. The address of each pointer is obtained from parameter data. In the range check of an FLQ, the load pointer may point to the word at the bottom of an FLQ. For hoppers, however, this would be a range check failure.

3.31 If the pointer range check fails, a message is printed on the TTY and the pointer changed as follows:

(a) For hoppers, the unload pointer is set to the first nonzero entry in the hopper and the load pointer to the first zero entry following this nonzero entry.

(b) If the hopper is empty (no nonzero entries), the pointers are both set to the first word. For FLQs, the pointers are also set to the first word.

Following the range check, further consistency checks on the pointers are made for hoppers only. The program checks that the hopper load pointers point to zero entries and that the unload pointers point to nonzero entries. However, when the hopper is empty, the load and unload pointer must point to the same zero entry. When the load and unload pointers point to the same zero entry, the program verifies that the next entry is also zero, indicating a truly empty hopper. If a failure occurs, an error message is printed on the TTY. The pointers are corrected so that the unload pointer points to a nonzero entry and the load pointer points to a zero entry.

F. SURT

3.32 PIDENT SURT, the ring trip supervisory scan initialization program, is used to update the ring trip table. The table consists of a word containing five times the number of ring trip rows followed by pairs of words which contain an expanded master scanner number (EMSN) and the address of the corresponding T bits.

3.33 For each ring trip row, the audit regenerates the EMSN without checking for errors, calculates the address of the associated T2 bits, and zeroes the T2 bits. SURT determines if the stored T1 address is equal to the calculated value, and, if an error is detected, the calculated T1 address is stored and a transfer is made to SAERCT to increment the error count and print the contents of the CC registers.

3.34 SURT may also be called by the maintenance recovery program after an F-level or out-of-range D- or E-level interrupt has occurred in the ring trip supervisory scan program to update the EMSNs and audit the memory data.

G. SACV

3.35 The receiver scan audit, SAVC, provides the office constant data used by digit reception, dialing connection, revertive pulse detection, and revertive pulse digit generator programs. SACV is a multilevel program, being requested on interrupt whenever an F-level interrupt or an out of range D- or E-level interrupt has occurred in the digit reception program, and on base level as a routine audit.

3.36 In addition to the constant data (determined by office parameters) SACV also provides certain variable data, ie, data dependent on real-time hardware configuration eg, active bus, central pulse distribution controller, etc.

3.37 Whenever the computed data base does not agree with the actual data being used, an audit error printout is requested if SACV is running on base level. If auditing or interrupt level, the errors are expected and no printouts are requested.

H. SAQU

3.38 This audit checks the accuracy of the cell information, linkage, and states of queued registers in variable length queues and timing lists.

Variable Length Queues and Timing Lists

3.39 Queues and timing lists are lists of call registers which have been linked together waiting for release of equipment, release of CS memory, a time-out, etc. The linkage may be 1-way beginning at a head cell and pointing only forward to the next register, or it may be 2-way, with linkage pointing in both forward and reverse directions. The head cell points to the first register on the list. The next register address (forward link) is always stored in the queue (second) word of the CR. For 2-way lists, the previous register address (backward link) is normally stored in the scan (fourth) word of the CR. The end of the list is indicated by all zeroes in the queue word of the last register or head cell.

3.40 A variable length queue is a 1-way list associated with a 2-word head cell. Word 0 of the cell points to the first register on the list, while word 1 points to the last register, which has the 0 end indication in its queue word. Certain queues may have registers of only one type (a given RI), in which case, the maximum number of registers allowed on the queue equals the number of this type in the office. If registers on a queue may be of any type, the allowable maximum is obtained from parameter word I4QMAX. Each register on a queue must have its queue indicator (QI) bit set to 1.

3.41 A 1-way timing list is associated with a 2-word head-cell; word 0 points to the first register on the list. The last register on the list points to the head cell, whose queue word contains the 0 end indication. Since registers may be of any type, the maximum number of registers allowed on a list is obtained from I4QMAX. Each register on a list must have its QI bit set to 1.

3.42 A 2-way general-purpose timing list is associated with a 4-word head cell; word 0 points to the first register on the list. The last register points to the head cell, whose second word contains the 0 end indication. The scan (fourth) word of the head cell contains the address of the last register on the list. Reverse linkage is found in the scan word of each register. The scan word of the first register contains the head cell address minus 1. Since registers may be of any type, the maximum number of registers allowed on a list is obtained from I4QMAX. Each register on a list must have its QI bit set to 1.

3.43 A 2-way special-purpose timing list has a 1-word head cell (see 3.44 for exception), which points to the first register on the list. The queue word of the last register contains the 0 end indication. Reverse linkage is found in the fourth word (word 14 if 100-ms timing list) of each register. The first register points to the head cell. Registers on a list may be of one type (a given RI); thus, the maximum of registers allowed on a list equals the number of this type in the office. The register QI bit is not audited.

3.44 The trunk answer TA (on centrex only) special-purpose timing lists are 2-way lists associated with 2-word head cells. Word 0 of the head cell points to the first register on the list, while word 1 points to the last register, which has the 0 end indication in its queue word. Reverse linkage is found in the scan word of each register. The scan word of the first register contains 0. Registers on a list may be of only one type (loop); thus, the maximum number of registers allowed on the list equals the number of this type in the office. The register QI bit must be set to 1.

3.45 An idle variable length queue or 2-way special-purpose timing list has an all-zero head cell word 0. The first word of an idle 1-way or 2-way general-purpose timing list head cell points to itself while the queue word has the 0 end code. In addition, the scan word of the 2-way general-purpose timing list head cell points to the head cell address minus 1.

General Operation

3.46 This audit functions in the general audit environment and is segmented, taking a real-time break after auditing each variable length queue or timing list. The audit uses an index (in the Z register) to identify the type of queue or list being audited and to obtain data on the type being audited, such as the address of the head cell. The order of auditing is 2-way special-purpose timing lists, variable length queues, 1-way timing, and 2-way general-purpose timing lists. See Table D.

TABLE D

QUEUES AND LISTS AUDITED (IN ORDER AUDITED)

NAME	PERMITTED REGISTER*
2-WAY SPECIAL PURPOSE TIMING LISTS	
1-min. Timing — Ground Start PBX	Originating
Partial Dial Timing	Permanent Signal
40-sec Timing	Permanent Signal
60-sec Timing	Permanent Signal
Permanent Signal Timing	Permanent Signal
100-ms Timing	Operator
TA Request List (Centrex only)	Loop
VARIABLE LENGTH QUEUES	
Coin Zone	Coin Charge
Local Coin	Coin Charge
AMA Output	AMA
Class of Service Tone 1	Operator
Calss of Service Tone 2	Operator
Ringing Trunk (Service link network only)	Regular Ringing
Low Priority Receiver [Automatic Identified Outward Dialing (AIOD) only]	AIOD
High Priority Receiver [Automatic Identified Outward Dialing (AIOD) only]	AIOD
Coin Control	All*
Regular Ringing	All*
Special Ringing	All*
POB	All*
Trunk-to-Trunk Path Memory	All*
Long Job	All*
ONE-WAY TIMING LISTS	
200-ms One-Way Timing	All*
1-sec One-Way Timing	All*
6-sec One-Way Timing	All*
2-WAY GENERAL-PURPOSE TIMING LISTS	
200-ms 2-Way Timing	All*
1-sec 2-Way Timing	All*
6-sec 2-Way Timing	All*

* All — does not imply that all registers actually use the queue or list.

3.47 On the first entrance from audit control, the program is initialized to begin auditing the first queue. On reentrances from the audit control after real-time breaks (by exiting to MACS25), the next queue or list is audited until all lists have been audited and the audit makes a final exit to the audit control (MACS24).

Linkage and Head Cell Validity Checks

3.48 For each queue or list, an index is used to obtain the address of the head cell, the maximum number of registers allowed on a list, and the specific RI permitted. The list of registers is traced from the head cell through each of the registers until an end code or error is found. For 2-way timing lists, the reverse linkage is also checked by verifying that each register on a list is linked back to the previous register on the list.

3.49 While tracing through the queue or list, a count is kept of the number of registers to protect a runaway queue or list in which the end code is destroyed or missing.

3.50 If the queue or list does not end properly, the linkage or the head cell information has been destroyed. In this case, the audit marks the queue or list empty and calls for the register audit by going to SAHELP. The CR audit will idle any register left stranded and busy when the queue or list was marked empty by audit SAQU. An SA03 error printout also occurs.

Register State Checks

3.51 Each CR found on a queue or list, on certain special purpose timing lists, is checked to verify that its QI bit is set to a 1, meaning that it expects to be on a queue or certain timing list. Every register is also checked to ensure that another audit has not found it in error and marked it in trouble by placing the audit code in it (RI = 0 and PT = 7). If a queue or list being audited is restricted to being used by one type of register, each register is checked for the permitted RI. A check is made to determine whether the maximum number of permitted registers has been exceeded, an idle register is on the list, or that each next register on a 2-way list is linked back to the previous register on the list. If any of the previous checks fail, it is destroyed by forcing it to be empty. The CC registers are printed out via the SA03 message and the CR audit is requested.

I. SADA

3.52 The regenerated constant audit, SADA, audits critical data areas in call store. With the exception of permanent signal and partial dial timing, private scratch used by the permanent signal timing program, and H- and J-level interrupt counters, the existing contents of the memory locations are ignored and blindly overwritten in all sections of the program. For these exceptions, the data is initialized only if the existing values are outside the valid range. The terms initialize and regenerate both mean to write constant values into memory. The former is generally used when there is likelihood that the previous contents are wrong; the latter, when the values being written are presumably the same as those being overwritten.

SAPUNT

3.53 SAPUNT is entered only when an emergency action (EA) phase 4 or greater has occurred and the EA control program has deemed it necessary to zero the CS. In this section, critical data necessary to start the office is initialized. Diagnosis of incoming and 2-way trunks during automatic progression is inhibited. The regulator and shifted diagonal tables of ones are generated. The central pulse distributor (CPD) status control word enable table and scan head cells, line load control, and automatic message accounting (AMA) pointers are initialized. In an EA phase greater than 5, all relays and cutoffs are restored to their normal initial states.

SAECIN

3.54 SAECIN is entered when EA phase 2 or greater has occurred, or when requested by PIDENT TNMC. The trunk and junctor busy-idle words are marked busy and the memory associated with the ECMP is zeroed (with the exception of the traffic timetable) and then initialized.

SACONS

3.55 SACONS is entered if an EA phase of 1 or greater has occurred and also on a routine basis. In this section, the following data which is used by a variety of programs is audited, initialized, or regenerated:

(a) Initialize ECMP constants

(b) Initialize permanent signal and partial dial timing

(c) Audit H- and J-level interrupt counters

(d) Initialize EA software and hardware (only in EA phases)

(e) Initialize CPD status control words

(f) Initialize enable table and scan head cells

(g) Regenerate expanded register identification (RI) copies in CS.

(h) Regenerate table of scan modes for start pulse signal detection

(i) Initialize PMTO to invalid code

(j) Regenerate recorded announcement control words

(k) Initialize Y4GTIM to nonzero data

(l) Regenerate receiver queue control words

(m) Regenerate the CS register identification table which for each RI points to its program tag (PT) table.

(n) Initialize timed scan junior register (TSJR), flash timing, and multibit scan head cells

(o) Initialize parameter hopper

(p) Initialize receiver scan parameter data

(q) Initialize queue execution

(r) Initialize centrex traffic measurements

(s) Regenerate screen words 04USCN, 04PSCN.

If an EA phase is not in process, this program calls the enable and status word audit subroutine NAMUD, and then terminates. If not in a phase 1, 2, or 3, the audit terminates. If an EA phase 1 or 2 is in process, the enable tables are updated before terminating the audit. If in a phase 1 (centrex offices only), the centrex enables are also updated before terminating the audit. If an EA phase 3 is in process, network memory is cleared and the audit is terminated.

J. NSUP

3.56 The enable table maintenance routines, NSUP, maintain the enable tables which provide preformed peripheral unit enables. The enable tables are maintained by NSUP as an integral part of fault recognition. Whenever a controller, bus, or CPD becomes suspect, the enable table entries affected are altered to specify the use of alternate equipment. The user would ordinarily not even be aware of the change. In the case, for example, of a quarantined controller, the fault recognition program replaces the normal enable table entry with an enable word for the mate controller, using a known working bus and CPD. A client desiring some peripheral action in the part of the fabric under control of the quarantined unit picks up the enable word for the mate. Since the mate is now in the combined mode, it is capable of controlling both halves of the fabric.

3.57 The master enable tables comprise one 4-word block for each network or signal distributor (SD) unit. Each 4-word block (enable block) is formed as shown in Table E.

TABLE E

ENABLE BLOCK

AEA0	Enable word controller 0
	Busy-idle word for 0
AEA1	Enable word controller 1
	Busy-idle word for 1

3.58 The AEA of a particular controller is the address of the enable word of that controller in the enable tables. The CS address corresponding the AEA0 always has 00 in its least two significant bits.

3.59 The busy-idled word is used in conjunction with the network cycle counter (Q6RING) to determine whether or not a controller is busy. Q6RING is essentially a ring counter containing a single 1 in a field of 0. This 1 is rotated to the right every network cycle until it reaches bit 0, from which it moves back to bit 15. Any 1 in bits 0 to 15 of the busy-idle word means that the corresponding controller is declared busy whenever Q6RING has a 1 in the same bit.

3.60 The enable word contains the enable placed in the F register to access the frame. It also contains some status information. The enable word layout is shown in Table F.

TABLE F

ENABLE WORD LAYOUT

BIT NO.	SIGNIFICANCE
22 21 20 19 18 17 16 15	CPD point
14	Bus choice
13 12 11	CPD pair
10	CPD choice
9 8 7	Translator
6	Enable verify
5	All Seems Well (ASW)
4	Primary trouble
3	Fault recognition active
2	This unit is route sensitive
1	Controller out of service
0	Busy-idle word to be used

3.61 In order to derive the address of the enable word of a network controller, the head cell system is used. Briefly, the process of deriving an AEA is as follows:

(a) The unit type of the circuit to be accessed determines which head-head cell to use.

(b) The address contained in the head-head cell directs the client to a block of head cells.

(c) The client divides the member number of his circuit by 2n where n is a function of unit type. Results of the division are a quotient (Q) and remainder (R).

(d) Indexing down the head cell block by the number Q provides a pointer to the proper group of enable blocks.

3.62 There are two routines included within the framework of NSUP to regenerate and update the enable tables. The first of these is a rapid, unsegmented regeneration program (NMENUP) which is intended to be run on F or J level or during emergency action. The other is the enable table and status table audit (NMAUD).

3.63 The demand update program is essentially a control routine for an execute table included in the parameter module. Sequential instructions of the table are executed, one for each loop, through the control program. These instructions load the base enables in the order that the enable blocks appear in CS. Legal routes for the two controllers are determined with reference to the bus and CPD status tables. The full enable (base enable routes, enable verify bit, all-seems-well bit, and busy-idle indicator) is inserted for each controller unless there is some indication that the unit has off-normal status. If it is off-normal, only the base enable is inserted.

3.64 In many cases, enable blocks are adjacent in memory. In order to take advantage of this fact, the control program assumes contiguity of enable blocks and the execute table contains transfers to a special repositioning program whenever subsequent blocks are not adjacent. This rather complicated scheme has been adopted to minimize F- and J-level holding time, which otherwise would be quite long whenever enables had to be regenerated in a large office. The demand update provides two other services:

(a) It updates the head cells and the head-head cells.

(b) It keeps track of whether or not any units are off-normal and so informs the client.

3.65 The enable table and status table audit program (NMAUD-system audit No. 3) is run routinely on base level through the MAC program. The audit serves the following functions:

(a) Regenerates base enables.

(b) Inserts routes, enable-verify, all-seems-well, and busy-idle indicators unless a unit is marked off-normal (use active CPD in routes—also randomizes bus selection for scanners).

(c) Corrects obvious inconsistencies in enable tables, such as:

(1) Both quarantine bits set.

(2) Unit marked permanently busy.

(3) Redundant status indicators in conflict.

(d) Keeps the status tables up to date.

(e) Zeroes the frame error counts hourly.

(f) Makes sure that maintenance (T2 bit) has not been turned off any frame by accident.

(g) Looks for network and SD controllers left illegally in the test point access mode and releases them.

(h) Keeps MCC status lamps up to date.

(i) Regenerates network F-scan buried enables.

3.66 The audit is changed in an EA phase. In order to run more quickly, the demand exercise is used. Then the buried enables are updated. If the phase is 3 or larger, the T2 bits are regenerated to restart maintenance on all equipped units. All network and SD controllers are released from maintenance modes and the routes are normalized.

K. SANK and SATS

3.67 PIDENTs SANK, the linkage audit of junior registers, and SATS, the expanded enable audit, are for the most part identical. The major difference is that SANK runs as a routine audit while SATS is run only as a result of an EA phase or an interrupt.

3.68 The SALINK subroutine of SANK audits the following junior registers:

(a) Timed scan junior register (TSJR)

(b) Flash scan and timing junior registers (FSTR)

(c) Multiple bit scan junior register (MBJR)

(d) Line ferrod disconnect junior register (LFJR)

(e) Multifrequency outpulsing junior register (MFJ)

(f) Trunk dial pulse outpulsing junior register (DPJ)

(g) Panel call indicator outpulsing junior register (PCJ)

(h) Step-by-step junior register (SXJ).

3.69 For each type of junior register, all members are checked by the audit. The address of each member is obtained from parameter information. The idle list for each type is initially made empty, and each idle junior register found is reidled, thus returning potentially lost idle registers to service. Busy registers are subjected to additional checks to ensure that they are validly busy. If the junior register is in a transient state not requiring it to have a senior register, then the state of the register is altered. If the audit sees a junior register in the altered state, the register is considered lost and is idled. Junior registers in states requiring senior registers to be associated with them are checked to ensure that a call register is in fact associated with it. This is done by seeing if the linkage from the junior to the senior and from the senior back to the junior is circular. In addition, the senior register must be busy (PT not 0) and it must not contain the audit code (RI = 0, PT = 7).

3.70 Whenever errors are detected, an SA03 error message is printed followed by a printout of the bad junior register. Bad junior registers are put in an abort state. This will result in termination of additional processing by call programs, and eventual idling either by the processing program or by the audit.

3.71 When errors are detected in the outpulsing junior registers (MFJ, DPJ, PCJ), an additional auditing routine is used. This routine checks various processing linking lists for sanity.

3.72 Since junior registers are processed by the I/O program operating on an interrupt level and the audit is operating on base level, there exists potential interaction problems between the audit and the processing progam. Where this exists, the audit sets the J-interrupt pest flip-flop, resets the H-interrupt pest flip-flop, and takes short segments to reduce the severity of suspending the low priority I/O processing.

3.73 When all the members of one type have been completely audited but other types remain, the audit transfers to MACS25 with a return address so that, when this program is reentered, the next type of registers will be audited. If, however, there are no other registers remaining to be audited, control is returned to MACS24, except that in a phase it passes control to SAEXP3 in SATS.

L. SASU

3.74 SASU, the supervisory scan data table audit, audits the next row addresses of the line bit tables for the line supervisory scan, the junctor bit tables for the junctor supervisory scan, and the T1-T2 bit tables for both the universal trunk and master scanners. SASU, comprising audits 07, 08, 09, and 10, has four GLOBAL entry points from MAC (No. 1A ESS only):

- SASU_LI (Audit 07)
- SASU_JS (Audit 08)
- SASU_UT (Audit 09)
- SASU_MS (Audit 10)

Each entry is a distinct audit and as such may be executed independently.

3.75 ***Audit 07,*** the line bit table audit, sets up the next row to scan in the line bit table for the line scan program. Each row is verified and an error message is printed if an incorrect row is found. Audit 07 also checks the next frame to scan word and if found to be too large, it is zeroed.

3.76 ***Audit 08,*** the junctor bit table audit, sets up the next row to scan in the junctor bit table for the junctor scan program. Each row is verified and an error message is printed if an error is found.

3.77 ***Audit 09,*** the universal bit table audit, checks the T1-T2 bit table associated with universal frames for correct row information. It uses part of the junctor program since both frames are identical except for the two-word data blocks.

3.78 ***Audit 10,*** the master scanner table check, checks both supervisory fields on a master scanner (SXS and non-SXS) for proper row data. Non-SXS supervisory rows are audited first.

3.79 The SASU program is multilevel, operating in D, E, and F levels as well as in base level.

M. SACT

3.80 For various reasons, groups of subscriber lines are sometimes transferred from one central office to another. The primary instance of line transfer occurs when a new central office replaces an existing office. In order to maintain uninterrupted telephone service when such a transfer occurs, subscriber lines must be connected to both the old and the new offices (where the ***old*** office is the one being transferred from, and the ***new*** office the one being transferred to) for a period prior to cutover, so that line connections to the new office can be tested while switching functions for these lines are performed by the old office. These lines must therefore be kept functionally isolated from the new office and serviced by the old. SACT, the cutover program for growth, provides this isolation. It also provides the mechanism for effecting immediate change of state of lines from isolated to supervised or vice versa. In addition, the cutover program provides the capability of accessing isolated lines from certain test facilities.

3.81 In the most general cases, there will be lines in one or both offices which are not involved in the transfer, and these may require different treatment from the lines being transferred. For example, the new office may already be providing service to some subscriber lines; then, when lines are bridged during the period before cutover, the new office must keep some lines isolated and provide service to others. It must also recognize that the calls to isolated lines are given interoffice treatment, while calls to normal lines are given intraoffice treatment. The cutover program provides the capability for making all these distinctions.

3.82 The cutover program may be present in either the old or new office; that is, it can handle line transfers into or out of an office (or both). When both offices involved in the transfer have the cutover program, it is possible to "undo" the transfer if desired, for example, if the new office has serious trouble during or immediately after the transfer. Any number of lines may be transferred, although the basic unit of transfer is 1000 directory numbers.

3.83 After all transfers have been performed, the cutover program is no longer necessary. The program is then inactivated.

3.84 The cutover audit assures that inactive lines are kept isolated from the switching machine by maintaining cutoff contacts of all lines in the office in the proper state—open for inactive lines, closed for active lines—according to translation information. The audit can be called in three ways. First, it is called by maintenance control during the normal audit cycle. Second, it can be requested by the message SA-AUDIT-64777777.

> ***Note:*** TTY messages shown here are for explanation only. Refer to IM-6A001.

Third, it can perform the actual cutover by using the message ESS-CUT-0 or ESS-CUT-1 with the CUT 0 and CUT 1 keys set properly.

3.85 There are also a number of interfaces with the cutover program for maintaining cutoff contacts in the proper state and for denying termination to inactive lines, except from selected facilities. The majority of these interfaces are accomplished through the standard originating and terminating translation routines, which indicate to their clients whether a line is active or inactive. For example, during a phase 4, translation information is interrogated to determine whether to open or close cutoff contacts. The effect of these interfaces can be controlled by means of a master control center key, CUT 1. When this key is set to ON, the cutover audit will not run and the interfaces are in effect eliminated, allowing cutoff contacts to be closed on inactive lines. Terminating translations are not affected by the CUT 1 key.

Cutover Audit

3.86 The purpose of the cutover audit is to maintain cutoff contacts of all lines in the proper state, according to rate center table and directory number exception list entries. The audit has two modes, routine and demand; demand mode is activated by performing the line transfer, and the audit runs only once.

3.87 ***Routine Mode.*** The routine audit (short for routine mode of cutover audit) can be requested in two ways: by TTY with the message

SA-AUDIT-64777777 (Sample)

or by maintenance control during the routine audit cycle. Every time the routine audit runs, it audits a single thousands group. Lines are audited from highest to lowest number group. Two peripheral order buffers (POBs) are used to hold orders to operate (open or close) cutoff contacts, each POB holding orders for a maximum of 15 lines. If failures occur in both POBs, the audit aborts and starts over at the highest number group the next time it is run. In addition to the standard SA01 SUCC 64 message, there are two messages associated with routine running of the cutover audit:

SA04 NORM (Sample)

indicates aborting of routine audit and

SA04 ERR (Sample)

indicates a POB execution failure occurred.

3.88 If the MCC CUT 1 key is set to ON, the routine audit will not run in the routine audit cycle, and cannot be requested via maintenance TTY. If the CUT 2 key is set to ON, the routine audit again will not run in the routine audit cycle, but can be requested via TTY. Entry point to the routine audit is SAROUT.

3.89 ***Demand Mode.*** Demand mode of the cutover audit is requested on a high priority basis when message

ESS-CUT-0 or ESS-CUT-1 (Sample)

is typed (with CUT 0 and CUT 1 keys set properly), requesting that line transfer be performed. Demand audit works only on number groups which are being transferred. It remains in control of maintenance control scratch storage until it has completed running, but it segments so that normal call processing can go on if the office contains working lines. The number of POBs used by demand audit is a variable supplied by the operating company (default value is 2). There are two messages associated with demand audit:

SA04 SUCC (Sample)

indicates demand audit ran successfully and

SA04 REQ (Sample)

indicates demand audit was aborted. Entry point to the demand audit is SADMND.

Interfaces for Maintaining Isolation of Inactive Lines

3.90 ***Maintaining Call in Progress During Cutover.*** In a line transfer where both offices isolate lines by removing battery, the cutover program maintains calls in progress. Lines involved in a call are then cut after disconnect. This is handled in the following way:

(a) If a line being cut ***out*** is off-hook when the transfer takes place, the cutover audit ignores it. Following disconnect, it is restored to normal (cutoff contact closed). On its next origination or when the routine audit is run, the cutoff contact is opened. In this case, in order to maintain a conversation the new office must be an ESS since electromechanical systems cannot withhold dial tone from off-hook lines.

(b) If a line being cut ***in*** is off-hook, its line bit is marked busy and its cutoff contact is closed, causing a drop in transmission on the line. When cutover audit terminates, it calls audit 50. If the line has disconnected by this time, audit 50 will put it into service by idling the line bit. Otherwise, audit 50 will place it on the high-and-wet list and the line will be restored to normal on disconnect. If the high and wet list is full, the line will be examined by audit 50 every audit cycle until it disconnects or until room exists on the high and wet list. In this case, the old office need not be an ESS as long as the line is cut out by removal of battery from the line relay.

N. SALT

3.91 PIDENT SALT audits translation and recent change (RC) area of CS. Entered at SATRHC, SALT first audits the translation head-head table, F4HHTC, which contains addresses that are pointers to translation and recent change data blocks (Fig. 2).

3.92 The correct addresses are computed from backup data and then compared with the contents of the head-head table. If a mismatch is found, an audit error message is printed and the pointer in the head-head table is corrected. Constant data in the trunk group head cells are audited for all types of trunk groups; and, for the 2-way trunk groups, the pointer to the activity block and the data in the activity block are audited. A real-time break is taken after every 50 trunk group head cells have been processed. The activity bits associated with a 2-way trunk group are contained in blocks of CS words with a maximum of 16-trunk activity bits in each word. All 2-way trunk group activity blocks are in a contiguous block. The maximum size of the block is determined by a parameter set card value (F4TWAY). Deletions or additions of trunks to one 2-way trunk group may require moving activity blocks from one location to another.

3.93 After auditing of the trunk group data, the audit calls for various equipment queues to be unloaded and will automatically set the queue bits located in the associated trunk group head cells to the proper state.

3.94 The toll network protection (TNP) bits located in the trunk group head cells are all zeroed in case they were accidently set. If a scan of the toll network control key located at the master control center (MCC) indicates it is operated, the subroutine LLCTNP (PIDENT LLOD) is used

```
                  ***************************************************
F4HHTC+0 ------>*                                                   *
                *        TRUNK GROUP HEAD CELLS                     *
                *                                                   *
                ***************************************************
F4HHTC+1 ------>*                                                   *
                *        MULTILINE HUNT GROUP HEAD CELLS            *
                *                                                   *
                ***************************************************
F4HHTC+2 ------>*        7   7   7   7   7   7   7   7              *
                *        0   0   0   0   0   0   0   0              *
                *        0   0   0   0   0   0   0   0              *
                *                                                   *
                *---------------------------------------------------*
                *        SERVICE &/ COMPLAINT OBSERVING &           *
                *        CALL FORWARDING TEMP RC AREA               *
                *                                                   *
                ***************************************************
F4HHTC+5 ------>*                                                   *
                *        DELAYED RC AREA                            *
                *                                                   *
                ***************************************************
F4HHTC+8 ------>
                 ---------------------------------------------------
F4HHTC+9 ------>
                ***************************************************
                *                                                   *
                *        MULTILINE HUNT GROUP ACTIVITY BLOCKS       *
F4HHTC+11------>*                                                   *
                ***************************************************
F4HHTC+10------>*                                                   *
                *        2 WAY TRUNK GROUP ACTIVITY BLOCKS          *
                *                                                   *
                ***************************************************
F4HHTC+12------>
```

Fig. 2—Sample Head-Head Table

to set the TNP bit in the proper trunk group head cells.

O. NEGN

3.95 NEGN (GLOBAL NECJGA) is the network head cell and juntor list audit. The purpose of NEGN is to regenerate the junctor subgroup (JSG) linked lists and their head cells using information from the translation data is unduplicated CS. NEGN also audits the network switch frame out-of-service tables.

3.96 The number of JSGs in a junctor group that connect any pair of networks is a function of the size of the office and the traffic between the link networks. Once the number is determined, the JSGs are arranged in a fixed linked list which may be changed when the office grows or the junctors are rearranged for some reason. The head cell associated with each group contains a JSG in the group, and it is a pointer to the next available JSG in the linked list.

3.97 The auditing of the JSG linked lists of the junctor groups follows a set pattern. Starting with LLN0 the first JSG linked list consists of those JSGs which come back on LLN0 (ie, the intralink network). The second linked lists will consist of those from LLN0 to LLN1, while the third will be from LLN1 to LLN0. Continuing with this pattern, LLN0-LLN2, LLN2-LLN0, ..., LLN0-LLN15, LLN15-LLN0, when all the linked lists are formed between LLN0 (to itself and all other LLNs), then the routine starts with the linked lists between LLN0 and all the TLNs, ie, TLN0 through TLN15. At the end of this, a 1 is added to the LLN, now making it LLN1 and the whole process is repeated. When the succeeding LLNs are completed, the linked lists are formed between the trunks in the same manner. However, no additional linked lists are formed for the JSGs going from TLN to LLN for the following reason. There is only one head cell assigned to a junctor group: between a TLN and LLN, and for an intralink network, ie LLNi-LLNi or TLNi-TLNi (where i = 0 through 15). On the other hand, there are two head cells per junctor group for interlinked networks, one for each link network (LN). Thus, the JSG linked lists are formed and inserted in the appropriate places for pairs for LNs (eg, LNi to LNi+1 and LN+1 to LNi). Only when this is accomplished, and before continuing with LNi to LNi+2, does segmentation take place.

3.98 In the PS, there is a 64-word block associated with each LN. The block is commonly referred to as the backup table. Each word of the 64-word block has a one-to-one correspondence with a JSG, and the word includes the following information: L, U, and NET. A 1 in L indicates that the linked list is associated with an LLN, and a 0 indicates it is a TLN; NET identifies which LN, 0 through 15. A 1 in U indicates that the JSG is unequipped. Using this backup table, the routine compiles the linked list but the order of the JSG in the linked list is dependent upon the slip associated with it. Two of the requirements regarding the order of the JSG are that a JSG of a given slip should not be preceded nor followed by another JSG of the same slip, nor of a complementary slip. Slips are numbered 0, 1, 3, 5, 7, 9, 11, 13, and 15. If the sum of the two slips is equal to 16, then they are complementary (eg, slip 1 is the complement of slip 15).

3.99 As the backup tables are examined, a check is made on each word to determine if there is any RC. If there is, it is the new information that is used; but, if not, the contents of each word are compared to the L and N of the particular LN being processed. The index of the address where a match occurs is the JSG. Before putting this JSG in its assigned place in a special table, a check is made to ascertain that it is not of a shelf that is to be rewired. (If the shelf number is greater than 7, it is invalid and the routine writes a 0 in it.) If the JSG is on the shelf, it is ignored and the examination of the table is continued. When the list is compiled if it contains no JSG, a 1 is written in bit 6 of the head cell word to indicate this. If there is only one JSG, it will be placed in the head cell and it will point to itself. For all unassigned LLNs and TLNs, an invalid code is placed in bit positions 21 through 18 of the associated head cell words and the no junctor (NJ) bit is set to 1.

3.100 In general, the JSGs are inserted in the network link map in their preassigned order. Before each JSG is inserted, it is compared to the JSG already there. If they are the same, then no changes are made, but if they are different, the new one is inserted and an error message is printed. In either case, the next JSG in the list is processed. If the linked list is not for JSGs on the same LN, or for LLN to TLN, then each JSG in the special table is replaced by the corresponding JSG of the other terminal of the junctors on the terminating LN. This new JSG is obtained by reading a junctor table (JT) indexed by the former JSG. The new linked list is inserted in the same manner as before.

3.101 After the JSG ring lists have been formed, control is passed to the routine NEWGRO (PD-1A113 for No. 1 ESS or PR-6A113 for No. 1A ESS). It audits the network switch frame out-of-service tables.

P. SAMP

3.102 PIDENT SAMP, the network and map audit program, comprises audits 60, 61, 62, and 63. These audits keep the busy-idle bits in the network map correct. The audits are forced to segment on a network basis:

LLN—A-links in one segment, B- , C- , and J-links in another

TLN—A- and B-links in one segment, C- and J-links in another.

The total number of busy-idle words audited at one time is 256. Accordingly, each audit may be requested on a per network basis. Since the four audits are very similar to each other, most of the program to perform the audits is common to all four audits. Tests are made in the program when necessary to find out exactly which audit is being executed. The basic procedure used by these audits is the same. The differences are because a different one of the four kinds of network map subtables is audited in each audit.

3.103 The audit method employed consists of first building a scratch memory image (in network scratch, L-level (T) scratch, and maintenance scratch) of the busy-idle words for the particular type of link and then comparing this image with the actual link memory. If any errors are found, they are corrected and an appropriate error message is printed. Busy-idle bits found to be idle in error are more serious than if found to be busy in error, ie, cutoffs may occur.

3.104 SAMP is entered at GLOBAL SAMAP from PIDENT SADT. The entry is provided with a data word specifying which SAMP audit is to run:

(a) TNBM—audit 60 (MAPBT)

(b) TNAM—audit 61 (MAPAT)

(c) LNBM—audit 62 (MAPBL)

(d) LNAM—audit 63 (MAPAL)

For example if audit 60 (MAPBT audits the C- and J-links of the TLN) were to be requested, data word TNBM would be used.

3.105 If SAMP is entered during a phase, the COPY subroutine performs an unconditional update of the network map (ie, no matching is done).

Q. MACA (No. 1A ESS)

3.106 MACA is the No. 1A ESS audit scheduler. The audit control routine program unit provides the interface between the ECMP, the audit package, and MACP, ie, the common portion of the maintenance program.

Audit Related MACA GLOBALS

3.107 The following GLOBAL entries in MACA are used for audit execution/control:

(a) MACRFILL—sets the fill entry flag

(b) MACRMAIN—initializes MACR to be equivalent to No. 1 ESS

(c) MACRAMAC—entry from ECMP on every class E main program (MP) cycle

(d) MACAPHAS—entry to run audits in stitched mode during a phase

(e) MACKSADT—checks to see if audit activity flag is set

(f) MACRREQA—sets the request high/low priority flag

(g) MACRCAIP—entry to change the current MACR job in progress

(h) MACACNLA—requests MACA to cancel a requested audit

(i) MACASTIM—sets the MAC timing flag for audits

(j) MACARTIM—resets the MAC timing flag for audits.

R. SADT

3.108 PIDENT SADT consists of a collection of routines which perform timing and scheduling for the audit control system, routines for saving stable state calls during a phase, and routines which comprise audits 24, 26, 30, 36, 46, 48, 52, and 59. SADT also contains the system audit vector table (SATV) which is a map of pointers to all of the audits.

Primary SADT Control Routines

- SASTRT—GLOBAL entry to start an audit request via the TTY.
- SAAEXM—GLOBAL entry used by all of the network audits to get the next PMT or PML to be audited when auditing PMTs or PMLs.
- SASSSC—routine to save stable state calls during a phase 3, 4, or 5.
- SANETX—GLOBAL entry to initialize the network audits.
- SAPROB—GLOBAL entry to the subroutine which drives the audits called by a call processing program or another audit which has found a problem. SAPROB decides which audit has the best chance of correcting the problem.
- SAROT—GLOBAL entry to the audit cycle driver routine. SAROT is entered from MACA to call the next sequence of audits.

- SAR0T7—audit slot 30 selects one of the map audits (60, 61, 62, 63) to run every eight audit cycles. Each map audit is requested (routinely) every 32 audit cycles.

SADT Audits

Audits 24 and 26 (TGA3 and JGA3)

3.109 These audits perform analogous checks examining the path memory for trunks (PMT) or PML words, respectively, for codes indicating that path memory information for this word is in a client register or a trunk to trunk (TTM) block. Audit 26 performs an additional check. Here, the LNTCHK subroutine is used to check the validity of the line network tag (LNT) in the displaced PML word. For both audits, redundant information in the client register or TTM block is then checked by appropriate conversion to verify that the information points back to the PMT or PML word being tested. If a client register is involved, the test also fails when the client register has an RI, PT of 0, 7 or a PT equal to 0. When the test fails, the TTN or junctor network number (JNN) involved is given to the SADMPT or SADMPJ routine, respectively, to idle. If a client register is involved, then the address is supplied to the SARLF routine to be marked with an RI, PT of 0, 7. These two audits have the standard length 2.4 ms segments (10 ms for No. 1 ESS).

Audit 30 (SAROT7)

3.110 This audit is used for control purposes only. It is called with other audits by the routine SAROT. Its function is to reduce the frequency of the map audits relative to the other routine audits called by SAROT, and to zero inhibit flags set by the SAPROB routine. Every eighth time SAR0T7 is entered, it requests one of the four map audits. In this way, the routine requests for individual map audits, which have long segments, are reduced to one thirty-second of the requests for other audits. Audit 2 also runs once every 32 passes. The SAPROB flags are zeroed each time audit 30 runs, ie, each time a routine audit cycle is completed.

Audit 34 (SATG2A)

3.111 This audit program audits the linked lists for hunting trunk groups. The technique is to examine the trunk group head cells starting at 0 and stopping at the maximum trunk group number (TGN). For each hunting trunk group, its idle linked list is checked. A trunk network number (TNN) to TGN translation is made for each trunk on the idle linked list, and the resulting TGN is matched with the TGN of the head cell. After sixty trunks on this linked list have been checked, the audit segments. If the list does not terminate after 1024 trunks have been checked or if the proper code does not terminate the list, the linked list is in error. If any errors are found in the list, the head cell links are zeroed, audits 4 and 36 are requested, and an error message is printed.

3.112 After all trunk groups have been checked and if the system is not in a phase 4 or 5, the audit terminates to MACS24. If the system is in a phase 4 or 5, action is taken to permit trunk and junctor restoral to begin. MACS24 is placed in E4RETN; I/O job 52 is requested; ECJHON turns on J-level work; and the audit exits to the routine TPHASE.

Audit 36 (TG2B)

3.113 This audit zeroes the head cells of the TMLA trunk linked list and then examines all the PMT words. All TNNs corresponding to PMT words with a nonpath code are checked for the TMLA code, the trunk maintenance list (TML), carrier group alarm (CGA) code, trunk make busy (TMB) code, or other code. Since this list was zeroed at the beginning of the audit, each trunk found with the TMLA code is placed on the TMLA with no error print. Each trunk with TML, CGA, or TMB code is supplied to the SADMPT routine, but the J register is negative. All other negative coded trunks are treated as if they had been given to SADMPT with a negative register, except that the check made for CGA or TMB is bypassed. This is done to save time in the audit since audit 58 will see that all trunks that should be in the CGA or TMB state are placed in that state. When SADMPT verifies that the TNN is actually in the state that is indicated by the code in the PMT word, it simply returns. If the code in the PMT word does not verify, this TNN is treated as a TNN supplied to the SADMPT routine with a positive J. This audit has standard length segments.

3.114 Audit 36 performs an additional function in an EA phase 4 or 5. Each trunk that appears to be in a path is checked to see if it can be allowed to remain in a path. The expanded trunk class code is used to determine this. If the code indicates that word 3 applies, or if the type of trunk circuit equals 3 and the circuit program index (CPI) is not equal to 4FIOTK, then the trunk is given to SADMPT. If the PMT was linked to a register, audit 16 will be requested. One of the implications of this action, that no path to a service circuit, an operator trunk, or a multiport trunk other than an intraoffice trunk (IOT) will remain connected in an EA phase 4 or 5.

Audit 46 (TGA1)

3.115 This audit, along with audit 48, protects against a limbo state in a line-to-trunk path memory. This limbo state would be a PML word for a line-to-trunk junctor (LTJ) pointing to a line without having a PMT word pointing to this junctor. If there is no PMT word, there is no supervision of any kind on the line. To protect against this, audit 46 checks the line-to-trunk junctor busy-idle (LTJBI) bits by examining all of the PMT words for a particular TLN to create an image table of these bits in scratch memory. Each word in the image table corresponds to a junctor subgroup (JSG). This word is then matched with the corresponding word in the proper LLN map. During this entire process, the audit cannot segment because a change in network (CIN) could be performed during the real-time break. If this was allowed to happen, the audits could tear down the paths involved in this CIN. Since the audit cannot segment except by TLN, which implies a long segment, it is provided with a data word (TNJR) so that a specific network may be audited without auditing the others. When the audit is run, any TNN, which appears to be in a line-to-trunk path whose LTJBI bit is found idle, is in error and is given to the SADMPT routine to idle. If any LTJBI image bits do not match with the LTJBI bits, the JNN corresponding to the mismatched bit is given to the SADMPJ routine to idle.

3.116 Two additional tests are made by the audit. First, as each PMT word that contains a LTJ is found, the JNN is checked for validity. If it is invalid, then the TNN associated with the PMT word is given to the routine SADMPT. Further, if the PMT was linked to a client register, then the client register address (CRA) is given to the routine SARALF (PIDENT SARG). The second test is to check if two or more PMT words point to the same LTJ. This situation should not occur because junctors are never shared in paths. The check is performed as the image table is built by testing whether the image busy-idle bit is already marked busy. If it is busy, the TNN associated with the PMT word is given to the routine SADMPT. Also, if the PMT was linked to a CR, the CRA is given to the routine SARALF.

Audit 48 (JGA1)

3.117 This audit, along with audit 46, protects against a limbo state in a line-to-trunk path. This audit checks the junctor busy-idle (JBI) bits with the PML words for a LLN. As in audit 46, an image table of JBI bits is formed and then compared with the JBI bits. Since this must be done without segmenting (except by network), this is another long segment audit. Because of this, a data word (LNJR) is provided to this audit which specifies which LLN is to be audited. When the audit is run, any JNN whose JBI bit is idle and appears to be traffic busy is in error and is given to the SADMPJ routine to idle. Any JBI bit which is busy but does not have a busy PML word corresponding to it is marked idle at the end of the segment and an SA03 print is given.

Audit 52 (JGA2)

3.118 The primary purpose of audit 52 (JGA2) is to check supervision of the junctors which are idle or in paths not linked to a CR. A secondary purpose is to verify that unequipped junctors have their PML words marked with the invalid code, and vice versa. This audit looks through all the PML words. It ignores PML words whose code indicates a path in a CR or a member of one of the JMLs. For all others, a JNN to JSG and channel translation is made on the JNN corresponding to the PML word. If the JSG is a line-to-trunk JSG, the PML is checked to see that it does not contain the invalid code. If the JSG is an unassigned JSG, the PML is checked for the invalid code. If either of the checks fail, the JNN is supplied to the SADMPJ routine and an error message is printed. For either a line-to-trunk or a line-to-line JSG, if the PML word is busy, the LNTCHK subroutine is used to check the validity of the LNT in this nondisplaced PML word. Finally, if the JSG is a line-to-line JSG, the supervision test is performed. If the PML word is marked with the idle code, its junctor scan bit is tested for the ignore state. If not, the JNN is supplied to the SADMPJ routine and an error message is printed.

3.119 The only remaining case is a line-to-line JNN whose PML word is not marked idle. The junctor scan bit is checked for the busy state in this case. If it checks, no further work is necessary, but failure of the check is ambiguous. The junctor may have just disconnected. To allow for this when the busy test fails, both the contents of the PML word and the JNN are stored in a 64-word table. When this table is filled, or when the audit finishes with all the junctors and there are entries in the table, 4-second timing is done. After the 4-second timing, the old contents of the PML words from the table are compared with the present PML words. If they have changed, the audit ignores them. If they have not, the junctor scan bit is checked once more. If it is still not in the busy state, then indicators are marked which will cause the associated JNN to be checked later. In this case and only this case, 4-second timing is done again. After this second 4-second timing interval the PML, JNN pairs left in the table are those which failed the second junctor scan bit check. For these, the old PML words from the table are again compared with the present PML words. If they have changed, the audit ignores them. If not, the junctor scan bit is again checked. Finally on this third check, if it is still not in the busy state, the audit changes it to the busy state and prints an error message. Except for 4-second delays when timing, this audit has standard length segments.

Audit 59 (TGA4)

3.120 This audit has three major functions: to check the state of the trunk T2 bits, to verify the state of trunks that appear to be in the trunk high-and-wet (THAW) state, and to modify the manner in which the first two functions are performed when the system is in an EA phase 4 or 5. The modifications enable the functions to do a more thorough check, but this type of check is only possible when the allowable call states are restricted as in a phase 4.

Check Trunk T2 Bits

3.121 The audit looks through all the PMT words and does one of the two checks. (The THAW check will be described later.) The other test is made for all trunks which are idle or in paths not linked to a CR. For each PMT word in this category, translations are made on the associated TNN to find all its scan points, and the special program index (SPI) or trunk program index (TPI) for each scan point. Each MS point for a trunk has a TPI and each pair of universal trunk scan points have a single SPI. The audit has one PS table associated with TPIs and a pair of PS tables associated with SPIs. One of the tables associated with SPIs is for the first of a pair of trunk scan points (point 0) and the other table is for the remaining scan point (point 1).

3.122 The information in the left-half of each table contains the expected value of the T2 bit for trunks in the idle state and the first eight words of the right-half contain the same information for trunks in steady state line-to-trunk paths without a CR. The next eight contain this information for steady state trunk-to-trunk paths in a TTM. Generally, only trunk side scan point information will be a one (1) in this last subtable because the line side T2 in a trunk-to-trunk connection is ignored.

3.123 For each span point associated with a non-bylink TNN being audited, the appropriate table is referenced and the expected value of the T2 is compared with the actual value. For bylink scan points the expected value is found in the SXS-trunk-word. In this case, the T2 is not checked if the seizure bit of this word is set to one; otherwise, it is checked for the accept state. If the T2 bits for all the scan points of the TNN agree with their expected values, the audit proceeds with the next TNN. If any T2 bit does not agree, both the TNN and associated PMT word are stored in a 64-word table. The TNN may be in some transition state. Entries in the same table are also made by the other portion of the audit which is checking HAW trunks.

Check Trunks in the HAW State

3.124 The audit has three distinct functions which may be performed for HAW trunks. They are:

(a) Time out HAW trunks when scheduled.

(b) Print all trunks in the HAW state upon request.

(c) Check the T2 and T1 bit of the supervisory scan point of the trunk in this state.

The first two functions are under control of the trunk maintenance programs. When it is time to try to time out HAW trunks (once per hour), they set CS word HWTRT0 to zero. When a TTY message requests that the HAW trunks be printed, they set item PRT to 1. In either case, they request audit 59. To avoid problems which could occur if these flags were set when audit 59 was already running, the audit takes a snapshot of them at the start of the audit by moving them to other locations. In the event that HAW trunk are to be printed, the audit prints TN14 THAW when it begins and TN14 THAW END when it is finished.

3.125 As mentioned previously, the audit looks through all PMT words. If a PMT word is not idle or in a path not linked to a CR, a test is made to see if the THAW code is in the PMT word. If it is not, the audit proceeds to the next PMT. For each THAW coded PMT, the audit first checks to see if time-out action is required and if the trunk is one of the trunks which can be timed out. If the trunk is to be timed out, bit 8 in the PMT is tested for 1. If it is a 1 and the audit succeeds in placing the trunk on the TML, the audit proceeds to the next PMT. If not, bit 8 is set to a 1 in the PMT so that the time-out will take place in the next hour, if the trunk does not change states in the meantime. Next, the audit checks to see if the HAW trunks are to be printed. If they are, a TN13 print is made giving the TNN and TGN of the trunk. Finally, the audit checks that the T2 is accept and the T1 bit is in the off-hook state for the scan point which is to supervise the trunk when it is in the THAW state. This check is bypassed, however, when the HAW trunk is a bylink trunk that has its seizure bit marked in the SXS-trunk-word. In all other cases, if either bit is in the wrong state, both the TNN and associated PMT word are stored in the 64-word timing table along with the other entries from the normal T2 check. This is done because the trunk may be in a transition from the HAW state to the idle state.

Processing the Timing List

3.126 When the table is filled or when the audit finishes with all the trunks and there are entries in the table, 4-second timing is done. After 4-second timing, the old contents of the PMT words from the table are compared with the present PMT words. If they have changed, the audit ignores them. If not, the appropriate THAW or non-THAW check is made. On this second check, if any trunk bit to be checked does not agree with its expected value, indicators are marked which will cause the associated TNN to be checked later. In this case and only this case, 4-second timing is done again. After this second 4-second timing interval, the PMT TNN pairs left in the table are those which failed the second check. For these pairs, the old PMT words from the table are again compared with the present PMT words. If they have changed, the audit ignores them. If not, the appropriate trunk bits for HAW or non-HAW trunks are again matched with their predicted states. Finally, on this third check, if any trunk bit does not agree with its expected value, then the trunk bit is set to its expected value and an error message is printed. Except for the 4-second delays when timing, this audit has standard length 2.4 ms (10 ms for No. 1 ESS) segments.

Modified Actions in an EA Phase 4 or 5

3.127 An EA phase 4 or 5 requires audit 59 to take several special actions. The most trivial of these is that the audit will not attempt to print any TN14 or TN13 messages for HAW trunks when it is in the EA phase. An important action taken is to set each T1 bit either off-hook or on-hook when the corresponding T2 bit should be accept and the PMT word is either path busy or idle, respectively. Another special action is to set certain disconnect guide bits in the PMT words to safe states. When a PMT word is involved in a TTM path, the MB bit is set to zero by the audit. When the PMT word is in a line-to-trunk path, both the MB and F bits are set to zero.

3.128 In those generics with trunk preemption, the following is done during an EA phase 4 or 5 for trunks linked to a TPC register (these are the only trunks which are linked to any CR and which are saved in an EA phase 4 or 5). Guide bit, T1/T2 bit, and SXS state word auditing is provided. The strategy is:

(a) Mark the guide bits.

(b) Set up the "CLOCK" word so that TG4TST correctly decides whether L-T, T-T, or T-T with TPC register exists.

(c) Cause the audit to look at these trunks during its first pass only.

3.129 The rest of the special actions are taken only when the audit is dealing with a bylink trunk. In an EA phase 4 or 5, the SXS-trunk-words are not zeroed and audit 59 audits them as it finds the corresponding PMT words. There is one of these SXS-trunk-words assigned to each bylink trunk and its format is shown in Fig. 3.

3.130 If the bylink PMT word is idle, the audit zeroes the SXS-trunk-word and proceeds with its normal work. Otherwise the audit requires the bits to be as indicated in the following cases:

(a) PMT in trunk-to-trunk path:

D = 0, C = 1, B = 1, A = TS, S = 0

(b) PMT in line-to-trunk path:

D = 0, C = 0, B = 1, A = complement of O bit, S = 0

(c) PMT in THAW state:

D = 0, C = 0, B = 0, A = O, S = don't care.

3.131 TS is the trunk state bit in the TTM and O is the originating bit in the PMT word. If any bit mismatches, the TNN is given to SADMPT to idle the trunk. If all bits match, the MTDN is regenerated and the S and AD bits zeroed. Finally, if all bits match, the A relay bit is examined again. If it is a one (1), then the audit continues with its normal work. When a path busy or HAW bylink trunk has the A bit equal to zero, special action is taken to ensure that the A relay is not operated. If the A bit is zero when a disconnect occurs on this trunk, the A relay would not be released if it were actually operated. Once a bylink trunk gets in this state, no seizure on the trunk can be detected by its scan points. To avoid the possibility of this limbo state after a phase 4 or 5, the audit loads a POB within the phase to blindly release the A relay when the PMT and A bit are as described above.

S. SAWS Audits

3.132 The three SAWS audits for the No. 1A ESS are:

(a) SAWSFHSH—(audit 2) a fast hash of the permanent core resident data, ie, the merge structures (maps, head tables, etc), run out of SAROT7

(b) SAWSHSH—(audit 23) a hash of core and disk resident data, run hourly

(c) SAWSMSH—(audit 55) a word-by-word matching operation of the core resident and duplicate data in file store. Both copies of file store are matched against core, run daily.

3.133 The SAWSFHSH, SAWSHSH, and SAWSMSH GLOBALS in PIDENT SAWSCX78 initialize entries in the audit control block (ACB) to provide a controlled entry into SAWSCMMN which controls the SAWS audit. Refer to Section 254-280-260.

22	21	20	19	18	17	16 — 0
D	C	B	A	S	AD	MTDN

NOTE:
D = DISCONNECT BIT
C = C RELAY BIT
B = B RELAY BIT
A = RELAY BIT
S = RESEIZURE OR SEIZURE BIT
AD = AUDIT TIME-OUT BIT
MTDN = SPECIALLY RESTRICTED MTDN OF SD-1A220-01 TRUNK

Fig. 3—SXS Trunk Word

T. AUDSMDIO—Common I/O Audit

Purpose

3.134 This is a common (1A Processor) program which audits the various data structures of the I/O system. The purpose of the I/O audit is to maintain the software integrity of the I/O system. Entry to this PIDENT is at global entry point AUDSIOMD which is the global entry into mutilation detection.

AUDSMDIO Program Units

3.135 This PIDENT consists of the ten program units listed below. Refer to this PIDENTs current program listing for details.

PU	NAME	GLOBAL ENTRY POINT
PU1	Output Message Register (OMR) Link List Audit	None
PU2	IO Channel Status and Channel Memory Block Audit	None
PU3	Output Message Register Audit	None
PU4	Idle CMR Occupancy Counter Audit	None
PU5	Verify OMR Address Audit (AUDS_VERIFY_OMRA)	AUDSVOMR
PU6	Put OMR In Audit State (AUDS_OMR_TO_AUDIT_STATE)	AUDSAOMR
PU7	Recover OMRs (AUDS_RECOVER_OMRS)	AUDSMC07
PU8	Idle OMR Scan	None
PU9	Unit Activity Audit	None
PU10	Reassign Table Audit	None

4. GLOSSARY

Block	An assigned area in memory.
Buffer	A storage area used to compensate for a difference in data flow rate
Class (of execution)	Refers to main program job classification from A (highest) through E (lowest).
Client	Refers to the program that is currently requesting service from other programs or routines.
Hopper	A CS area containing information being referred from an I/O program to a call control program.
Phase	Refers to a stage or "phase" of system reinitializaton.
Queue	A waiting "list" in CS of jobs to be done.
Register (software)	A specially allocated memory area where data is operated upon according to job progress.

5. ABBREVIATIONS AND ACRONYMS

ACB	Audit Control Block
AEA	Address of Enable Address
AIOD	Automatic Identified Outward Dialing
AMA	Automatic Message Accounting
AS	Audit Subsystem
ASW	All Seems Well

AUDSMDIO	Common I/O Audit	JBI	Junctor Busy-Idle
AUX	Auxiliary	JML	Junctor Maintenance List
CC	Central Control	JNN	Junctor Network Number
CGA	Carrier Group Alarm	JSG	Junctor Subgroup
CGN	Console Group Number	JT	Junctor Table
CIC	Change in Circuit	LEN	Line Equipment Number
CIN	Change in Network	LFJR	Line Ferrod Disconnect Junior Register
CNLP	Centrex Lamp Order Program	LNN	Line Link Network
CPD	Central Pulse Distributor	LN	Link Network
CPI	Circuit Program Index	LNT	Line Network Tag
CR	Call Register	LTD	Local Test Desk
CRA	Client Register Address	LTJ	Line-to-Trunk Junctor
CS	Call Store	MAC	Maintenance Control Program
CTXN	Centrex Number	MACA	No. 1A ESS Audit Scheduler
CXSR	Centrex Call Register Audit	MBJR	Multiple Bit Scan Junior Register
D (bit)	Diagnostic	MBT	Maintenance Busy Bit
DPJ	Dial Pulse Outpulsing Junior Register	MCC	Maintenance Control Center
EA	Emergency Action	MF	Multifrequency
ECMP	Executive Control Main Program	MFJ	Multifrequency Outpulsing Junior Register
EMSN	Expanded Master Scanner Number	MP	Main Program, ie, ECMP
ESS	Electronic Switching System	MTDN	Miscellaneous Trunk Distributor Number
FLQ	Fixed Length Queue	NEGN	Network Head Cell and Junctor List Audit
FSTR	Flash Scan and Timing Junior Register	NMDT	Line Bit Audit
HAW	High and Wet	NSUP	Enable Table Maintenance Routine
I/O	Input/Output	OM	Output Message
IOT	Intraoffice Trunk		

OMR	Output Message Register
OS (bit)	Out of Service
PCJ	Panel Call Indicator Outpulsing Junior Register
PML	Path Memory for Lines
PMT	Path Memory for Trunks
POB	Peripheral Order Buffer
POMC	Peripheral Order Buffer Audit
PRI	Route Index
PS	Program Store
PT	Program Tag
QI	Queue Indicator Bit
RC	Recent Change
RI	Register Identifier
SACT	Cutover Audit Program
SACV	Receiver Scan Audit
SACX	Centrex Register Audit
SADA	Regenerated Constant Audit
SADT	System Audit Programs
SAHO	Hopper and Fixed Length Queue Audit
SALT	Translation Audit
SAMP	Network and Map Audit
SANK	Linkage Audit of Junior Registers
SAQU	Audit for Variable Length Queues
SARG	Call Register Audit
SASU	Supervisory Scan Data Table Audit
SATS	Expanded Enable Audit
SAWS	Writable Store Audit
SD	Signal Distributor
SPI	Special Program Index
SURT	Ring Trip Supervisory Scan Initialization
SXJ	Step-by-Step Junior Register
SXS	Step-by-Step
T (bit)	Trouble
TA	Trunk Answer
TGN	Trunk Group Number
THAW	Trunk High and Wet
TLN	Trunk Link Network
TMB	Trunk Make Busy
TML	Trunk Maintenance List
TMLA	Trunk Maintenance List Addendum
TNN	Trunk Network Number
TNP	Toll Network Protection
TPI	Trunk Program Index
TSJR	Timed Scan Junior Register
TTM	Trunk-to-Trunk Memory
TTY	Teletypewriter

6. REFERENCES

A. TOP 231-369-001

B. OM-6A001—Output Message Manual

C. Section 231-045-230

D. IM-6A001—Input Message Manual.