

AUDIT PROGRAMS
SOFTWARE SUBSYSTEM DESCRIPTION
NO. 3 ELECTRONIC SWITCHING SYSTEM

CONTENTS	PAGE	CONTENTS	PAGE
1. GENERAL	2	Power Alarm and Miscellaneous Scan Point Ignore Bit Audit	9
2. AUDIT FUNCTIONS	3	Equipped Peripheral Decoder (PD) Map Audit	10
A. General	3	Plug-Up and Line Timing Table Audit	10
B. Audit Monitor (AUDMON)	3		
C. Stable TMR Audit (STABLEAUD)	4	3. GLOSSARY	10
Call Forwarding Audit (CL 4WARD)	5		
D. Transient Call Audit (TRANSAUD)	5		
Three-Way Audit (3WAY MON)	5		
E. Busy Line Audit (BLAUDIT)	5		
F. Busy Service Circuit Audit (BSCAUD)	6		
G. Busy Trunk Audit (BTRKAUD)	6		
H. Test Vertical Status Bits Audit (TVAUD)	6		
Interrupt Test Vertical Audit (intvaud)	7		
I. Network Audit (AUDNET)	7		
J. Main Store Audit (MASAUD)	8		
K. Maintenance Audits (MAUD)	8		
Program Organization	8		
Trunk and Service Circuit Out-of-Service Counters Audit	8		
		Figures	
		1. Audit Control Block (AUDCNTL), SO-2 Issue 4A Generic	12
		2. Audit Control Block (AUDCNTL), 3E3 Generic	13
		3. Audit Table (AUDTABLE)	13
		4. Relationship of Audits to Base Level Loop, SO-2 Issue 4A Generic	13
		5. Relationship of Audits to Base Level Loop, 3E3 Generic	14
		6. Audits Functional Sequence	14
		7. Stable Terminal Memory Record (TMR)	15
		8. Transient Call Record (TCR) Layout for Audits	17
		9. Transient Memory Record (TMR)	19

NOTICE

Not for use or disclosure outside the
Bell System except under written agreement

Table	PAGE
A. Maintenance Audits	20

1. GENERAL

1.01 This section describes the audit programs used in the No. 3 Electronic Switching System (ESS) to maintain consistency between the various call processing records in temporary storage. The audit programs check all resources marked busy to ensure that they are actually busy.

1.02 This section is being reissued to provide information concerning the 3E3 generic. Since this is a general revision, no revision arrows have been used to denote significant changes.

1.03 The following Bell System Practices provide background information related to No. 3 ESS audit programs.

SECTION	TITLE
233-151-125	Input Processing and Scanning, Software Subsystem Description, No. 3 Electronic Switching System
233-151-130	Basic Call Processing, Software Subsystem Description, No. 3 Electronic Switching System
233-151-150	Translations, Software Subsystem Description, No. 3 Electronic Switching System
233-152-125	System Control, Software Subsystem Description, No. 3 Electronic Switching System
233-153-105	Programmed Maintenance Aids, Software Subsystem Description, No. 3 Electronic Switching System

1.04 The following programs contain the codes and comments that detail the audit functions in the No. 3 ESS:

- (a) The Audit Programs (AUDB), PR-3H001-01, audit lines, service circuits, trunks, and test verticals.

- (b) The Audit Programs (AUDITS), PR-3H002-01, schedule audits at the end of each base level loop, maintain the minimum time consumed in the base level loop (only for SO-2 Issue 4A generic), and include the stable and transient calls audits.

- (c) The Network Audit Program (AUDNET), PR-3H003-01, performs a routine network map audit as well as regeneration on the network map.

- (d) The Custom Calling Audit Program (CAUDIT), PR-3H098-02, contains audits for call forwarding, call waiting (only for SO-2 Issue 4A generic), and 3-way calling.

- (e) The Common System Maintenance Monitor Program (CMMON), PR-1C963, consists of a set of ESS Common System routines which control and monitor maintenance programs.

- (f) The Input Monitor Program (INPUT), PR-3H160-01, dispenses system inputs detected by scanning programs to programs that service inputs.

- (g) The Maintenance Audit Program (MAUD), PR-3H314-01, performs audits on various out-of-service items and alarm counts.

- (h) The Maintenance Monitor Application Program (MMONA), PR-3H203, works in conjunction with the CMMON program and contains tables, control parameters, and routines necessary for a particular application.

- (i) The Test Vertical Administration Program (TVADM), PR-3H178-01, performs a test vertical audit at interrupt level.



Any symbolic name not defined within a figure is defined in the glossary.

1.05 A resource may be marked false-busy due to a program error, a hardware fault, or a combination of both. The audits attempt to verify that there is a record pertaining to all resources being used with an active call. When a resource without a record is found, the audit idles the resource and prints a teletypewriter (TTY) message so that the program error that caused the trouble

can be located. In many cases, audits will not completely clear a call record or an invalid situation; therefore, the situation is corrected by a sequence of audits, which may result in several TTY messages for a single failure.

1.06 All audit programs operate out of the audit control block (AUDCNTL). This 8-word block of dedicated temporary memory is used to select the next audit to be performed and to allow spanning of real-time breaks by providing a dedicated work area and a progress mark (applies for both generics). In the SO-2 Issue 4A generic, the audit control block is also used to force audits that have been blocked too long. In the 3E3 generic, the block is used to indicate to the MMON program whether the audits are being initiated for the first time in the present base level loop. The audit control block for the SO-2 Issue 4A generic is shown in Fig. 1 and the block for the 3E3 generic is shown in Fig. 2.

1.07 When an audit is to be initiated, the entry point is defined by the audit table (AUDTABLE), Fig. 3. This entry point is entered in the audit progress mark (AUDPM) of AUDCNTL by the audit monitor (AUDMON). Subsequent entries to the audit are made at the address recorded in AUDPM. An audit program terminates by branching to the audit monitor at AUDDONE.

1.08 For the SO-2 Issue 4A generic, individual audits are scheduled at the end of each base level loop. The base level loop is a series of programs that are run sequentially over a period of 100 milliseconds. The audit monitor actually controls the timing of a base level loop, and audits are scheduled when a portion of the 100-millisecond period remains. If no time remains, audits are not scheduled; however, a counter is incremented. When too much time passes without an audit (approximately 7 minutes), the monitor will force an audit and set up for a time-out 1 minute later. The functional relationship of the audits to the programs in the base level loop is shown in Fig. 4 (for the SO-2 Issue 4A generic).

1.09 For the 3E3 generic, audits are scheduled by the dispatcher subroutine (DSPTCHER) in the program MMON. The dispatcher is entered from the CBLM program at the end of the nondeferrable work associated with the base level loop. The dispatcher will give control to one of 16 possible deferrable clients of which call processing

audits and main store audits are included. The deferrable jobs are defined in a table called JOBADDR in MMONA together with associated parameters which are used to run the job according to its defined priority and minimum execution interval. Normally, only "waste time" (time remaining before the start of a new base level loop) is used to execute the routines under the control of the dispatcher and any given client may receive control several times per base level loop. However, jobs which fall behind a performance schedule are automatically given priority and/or forced regardless of system condition. The relationship of audits to the base level loop is shown in Fig. 5.

2. AUDIT FUNCTIONS

A. General

2.01 The audit function is composed of a number of independent programs and subroutines which are scheduled sequentially by a common monitor as follows:

- (a) Stable Terminal Memory Record (TMR) Audit (No. 1) in AUDITS
- (b) Transient Call Audit (No. 2) in AUDITS
- (c) Busy Line Audit (No. 3) in AUDB
- (d) Busy Service Circuit Audit (No. 4) in AUDB
- (e) Busy Trunk Audit (No. 5) in AUDB
- (f) Test Vertical Audit (No. 6) in AUDB
- (g) Routine Network Audit (No. 7) in AUDNET
- (h) Main Store Audit Control (No. 8) in AUDITS (only for SO-2 Issue 4A generic)

The functional relationship of the individual audits is shown in Fig. 6.

B. Audit Monitor (AUDMON)

2.02 The audit monitor performs the common processing for all the audit functions and provides the entry and exit points for the programs when the audit is in progress, completed, or for error returns.

2.03 A delay service, which is required by the audits when corrective action is performed, is also provided by AUDMON. This function provides a delay period of 16-base level loops which allows the calling program to take corrective action. The audit control block (AUDCNTL) and the audit table are the primary data elements used by AUDMON during its scheduling function.

2.04 For the SO-2 Issue 4A generic, the primary function of AUDMON is to maintain the time for the base level loop. This is accomplished by computing the time since the start of the base level loop and determining if sufficient time remains to schedule an audit. When insufficient time remains, AUDMON will increment an audit-forcing timer blocking further audit programs during that loop. When too long a period passes without an audit (approximately 7 minutes), a time-out will be set and audits are forced. When sufficient time does remain, AUDMON will schedule the next sequential audit. When this audit is complete, AUDMON is notified and schedules the next audit on the list within remaining time.

C. Stable TMR Audit (STABLEAUD)

2.05 This audit is divided into 15 segments, each processing a group of 32 terminal memory records (TMR), Fig. 7. Each group of TMRs corresponds to the junctors whose scan points are in the same network concentrator group (scan points on the same logical scanner). The audit inspects and verifies the following conditions:

- (a) That the *ignore bits*, which inform the scanning programs to ignore the status changes of the junctor scan points, are 0.
- (b) All junctor scan points associated with lines are off-hook.
- (c) The line(s) in the TMR is in the busy state.
- (d) Incoming trunk scan point (calling party) is unmasked and off-hook.
- (e) Outgoing trunk scan point (called party) is unmasked.
- (f) Junctors corresponding to idle TMRs are idle.

These tests are repeated until all 32 TMRs have been completed. Once this is accomplished, the next group (logical scanner) is identified and the process is repeated until all 15 segments have been completed.

2.06 The first task of the audit program is to identify the proper (scanner) segment and 32 associated TMRs. A stable TMR, eg, one in which the active (ACT) and stable (STAB) bits are set, is then located. The *ignore bits* for the junctor associated with the TMR are tested to ensure they are 0. In the 3E3 generic, the split bit in the TMR is checked. If set, a call waiting audit is performed (the function of doing the call waiting audit is performed by the stable TMR audit). The call waiting audit verifies the call waiting group and checks to see that the split connection is not left by itself. If call waiting is verified, the call forwarding bit in the TMR is checked. If set, the call forwarding audit (CL 4WARD) in the program CAUDIT is invoked (paragraphs 2.09 and 2.10).

2.07 Various checks are performed to determine which party is the calling party and whether the party is a line or a trunk. When the calling party is a line, the line status bits are located and checks are performed to determine that the junctor scan point status shows off-hook and that the line status bits are marked busy. When the calling party is identified as a trunk, the *ignore bits* are checked for a 0. A check is then made to ensure that the trunk scan point status is unmasked and off-hook. For both line or trunk status checks, incorrect *ignore bits* are corrected and an output message, RCOVRY AU, is printed which identifies the audit which was performed, the test in which the error occurred, and the TMR number.

2.08 The stable TMR also contains a called party. When the called party is identified as a line, the audit is performed exactly as for a calling party. When the called party is an outgoing trunk, the *ignore bits* are checked and corrected as required, and the scan point status is checked to verify that the scan points are unmasked. The off-hook status is not checked since on an outgoing trunk the status can be in either state, depending on signaling conditions at the moment of the check. As before, all erroneous *ignore bits* are corrected as required, and a report is printed out on the TTY.

Call Forwarding Audit (CL 4WARD)

2.09 The call forwarding audit (3E3 generic only) is performed by the subroutine CL 4WARD in the program CAUDIT. The audit is directly under the control of the stable TMR audit. After the stable TMR audit has been performed, the TMR is tested for call forwarding status. If call forwarding exist, the call forwarding audit is invoked; otherwise the stable TMR audit continues in the normal fashion. The call forwarding audit may also be invoked after the call waiting audit portion of the stable TMR audit (see paragraph 2.06).

2.10 The call forwarding audit has two main purposes: (1) to check the linkage between the TMR and auxiliary TMR which have been chained together and (2) to verify the validity of the last link (indicated by the call forwarding bit being 0). Upon completion of the audit, the registers are restored to their entry status. When the last link indicating bit is erroneously set or there is erroneous linkage in the TMR chain, a RCOVRY AU message is printed indicating the error code, the number of the TMR associated with the error, and the contents of the four TMR words (in octal).

D. Transient Call Audit (TRANSAUD)

2.11 The transient call audit is performed by the subroutine TRANSAUD in the program AUDITS. This audit consists of two parts; the transient call record (TCR), Fig. 8, audit and the TMR audit. The TCR audit verifies the linkages between TCRs and TMRs starting with all active TCRs. The transient TMR audit is basically the same except the linkage is verified from the TMR to a TCR.

2.12 The TCR audit searches the TCRs until an active TCR is located. The audit code (ACODE) is then located and a special audit code bit is tested. If the bit is set, a branch is made to the SPECIAL ACODE subroutines. By testing the base level progress mark (BASEPM), SPECIAL ACODE determines if the TCR is associated with 3-way calling. If so, the 3-way audit subroutine 3WAY MON in the program CAUDIT is invoked. (See paragraph 2.14.) The audit code is then used to determine which of three possible (A, B, or T) junctors is in use. The junctor identity specifies which TMR is in use. The linkage from the TCR to the TMR and back to the TCR is verified by assuring that the A and B party terminal equipment numbers in

the TCR are matched in the specified TMR. When a mismatch, eg, a bad link is located, the message RCOVRY AU is printed specifying the TCR and the party involved in the bad link; then the TCR is cleared. The TCR audit returns to AUDMON after processing each active TCR. Therefore a complete TCR audit is spread over many cycles of the base level loop.

2.13 The transient TMR audit is started by a check of all TMRs marked as transient, eg, stable bit not set. When a transient TMR (Fig. 9) is located, it is checked for the presence of a valid TCR number. The TCR is then checked for the associated junctor switch number, which should identify the proper TMR, thus verifying correct linkage. When the correct linkage is found, the function will continue cycling through the TMRs until all are checked. When a bad linkage is found, the TMR involved is removed from the transient category by marking it stable. The RCOVRY AU message is printed giving a failure code for an erroneously idle TCR or that the proper junctor switch number was not in the TCR.

Three-Way Audit (3WAY MON)

2.14 The subroutine 3WAY MON in the program CAUDIT is the main monitor for 3-way calling. The subroutine receives control directly from the TCR audit portion of the transient call audit TRANSAUD. The primary function of 3WAY MON is to perform a series of party verification test. These tests are not conventional party verification tests as performed in other audits but are actually tests to determine that the correct party is associated with the correct port and conference circuit and to determine which is the 3-way customer. The junctor switch number (JSN) is used to determine each TMR number. A crosscheck is then performed to assure that the TMR and TCR information agrees. Should the TCR be found in error, an error message is printed and the TCR is cleared.

E. Busy Line Audit (BLAUDIT)

2.15 The busy line audit is performed by the subroutine BLAUDIT in the program AUBD. The audit searches for lines that are marked normal busy but not actually in use. At least one line assigned to one concentrator group (logical scanner) is audited during each audit period.

2.16 The audit accomplishes the check by first locating a busy status in the line status bits and then ensuring that the equivalent scan point number (SPN) is in the input hopper or a TMR. When no record of the busy line is found, the RCOVRY AU message specifying the incorrect line is printed. The terminal equipment number (TEN) of the incorrect line is placed in an idle TCR and a failure subroutine will fail the associated TCR and line. Due to the time required for the record search, only one line is audited for each entry to BLAUDIT, thus many base level loops are required before all busy lines are checked.

F. Busy Service Circuit Audit (BSCAUD)

2.17 The busy service circuit audit is performed by the subroutine BSCAUD in the program AUIDB. The audit looks for service circuits whose selection status bits indicate busy but which are not in any TCR. One busy service circuit in each group is verified each audit cycle. A circuit is selected by generating a pseudo-random member number. The system timing register contents are used as the source of the random number.

2.18 The audit is started by selecting the group number of the circuit to be audited. The ALL CIRCUIT BUSY bit is then checked. If the bit is set, a check is made to verify the busy conditions. If a circuit is found that is not busy, the bit is cleared and the RCOVRY AUO AB message is printed. The random member number is then generated and the selection status data for that member is located. The selection status data is searched for a busy indication. If busy is indicated, a search of the TCRs is started to determine if the service circuit number is in either service circuit slot in a TCR. When the number is found in a TCR, a return is made to AUDMON and the process is repeated until all service circuits have been audited. When the specified busy service circuit number from the selection status word is not found in a TCR, the RCOVRY AU BS message is printed specifying the group number and member number of the failed service circuit. An idle TCR is then located and the circuit number is held in this TCR for 2 minutes, after which the service circuit is idled, and the TCR cleared. This is done since there is a chance that a bad TMR has picked up the falsely busy circuit during the transient TMR audit. In this case, the circuit will be cleared when the associated TCR is idled. When it is not, the circuit is idled after the 2-minute wait.

G. Busy Trunk Audit (BTRKAUD)

2.19 The busy trunk audit is performed by the subroutine BTRKAUD in program AUIDB. The purpose of the audit is to search for trunks with selection status bits that are marked busy, but are actually idle. Trunks are audited one trunk per group on a group by group basis.

2.20 The audit is started by performing a translation to locate the trunk group data for the selected group. The select circuit subroutine (SEL CKT) is then invoked. SEL CKT accesses the groups selection status register and searches until a busy circuit indication is located. A check is then made to see if the circuit is marked "maintenance busy". If so, a high and wet audit (HWTEST) is performed to check if the trunk is actually in a high and wet state. If the trunk is in a false high and wet state and is not in processing, a high and wet audit message is printed and the HAW CLR routine is called to clear the high and wet condition.

2.21 When SEL CKT locates a busy circuit and it is not marked maintenance busy, control is returned to BTRKAUD. The selection status register for this group is then accessed and searched until a busy circuit indication is located. Active TMRs are then searched to determine whether or not the trunk is recorded in a TMR by way of a scan point. When the trunk is recorded, a return is made to AUDMON and the audit continues by accessing the next group until all groups are completed. One busy trunk per group is audited during each audit cycle.

2.22 When the trunk cannot be found in a TMR, a search is made of the TCRs. This is done because in some cases (dial pulse trunk receiving digits and during disconnect) a trunk may be used only from a TCR. Therefore a search must be made of the TCRs to try and find the TEN of the A-PARTY or B-PARTY position. When it is found, a return is made to AUDMON to continue the audit. When the trunk is still not found, a RCOVRY AU BT message is printed indicated the trunk group and member numbers. The trunk which failed the audit is idled and a return is made to AUDMON to continue the cycle.

H. Test Vertical Status Bits Audit (TVAUD)

2.23 The test vertical status bits audit is performed by the TVAUD subroutine in the AUIDB program.

The audit consist of two parts: (1) the auditing of the test multiple (TM) status bits and (2) the auditing of the test vertical (TV) status bits.

2.24 In the TM portion of the audit, a check is made of the TCR currently claiming each busy TM to ensure that it is active. The TM is first located and the status bits are accessed. The bits are then searched until all bits have been checked or a busy bit is located. When a busy bit is not found, a return is made and the cycle continues until all TMs have been checked (at which time the audit stops). When a busy status bit is found, the TCR claiming the TM is checked to see if it is active. If so, a return is made and the cycle continues. If the TCR is not active, the TM is idled and the TCR ownership pointer is cleared. The RCOVRY AU TV message is printed identifying the TM falsely marked busy and the number of the TCR incorrectly owning the TM.

2.25 In the TV portion of the audit, the TV group is located and the status bits are accessed. The status bits are then searched until all have been checked or a busy bit is located. When a busy bit is not found, a return is made and the cycle continues to the next group until all groups have been checked. When a busy bit is found, the corresponding TM is obtained and the status bits checked. A TV may be operator busy only if the corresponding TM is normal or maintenance busy. If this is not the case, the TV is idled and the RCOVRY AU TV message is printed identifying the error and the erroneously set TV.

Interrupt Test Vertical Audit (intvaud)

2.26 Because TV status information is manipulated both at base and interrupt levels, additional auditing of the TV is performed at interrupt level when the fast trunk scanning program (FASTTK) passes control to the test vertical administration program (TVADM). The interrupt TV audit (intvaud) in TVADM audits the maintenance busy state of the TVs to ensure against the possibility of an aborted diagnostic leaving a TV maintenance busy for a long period of time. The maintenance busy state is defined by the maintenance busy bit (MBUSY) in the TV status register. The MBUSY bit is set whenever a false cross and ground, power cross, or a restore-verify test attempts to seize a TV and finds it maintenance or no-test busy. TVAUD schedules the audit every 640 ms. The audit checks that a legal blockage exists and then resets the

MBUSY bit when the blockage disappears. The audit also scans all the TV test circuits for a service busy indication. If a service busy is found, the test circuit is audited by checking the owning TCR to determine if it is active and performing a peripheral action. If the check fails, the test circuit is idled.

2.27 In the 3E3 generic, two additional functions are performed by TVADM. First, the TV reserve bits are audited. The bits are checked to see if they are set. If so, the appropriate test circuit and TM are checked to verify the set condition. If it is found that the TV reserve bit should not be set, the bit is cleared. Secondly, TVs found in the test circuit busy state are audited against the test circuit. The audit checks to see if the TV should be test circuit busy. When a discrepancy is found, the TV is idled.

I. Network Audit (AUDNET)

2.28 The network audit is performed by the program AUDNET and provides the means to:

- (a) Perform a routine network map (NETMAP) audit.
- (b) Perform a NETMAP regeneration. The audit program basically regenerates an image of what should be in the map area to be audited. The two main sources of data used in the NETMAP audit are the TMRs and the link out-of-service list (OSLST). For routine audits, real-time breaks are allowed to return control to the base level loop. During map regeneration, no interrupts are permitted.

2.29 The network map audit is performed by first copying an image of NETMAP into a location titled AUDMAP. A flag is set up for use by the path hunt (PATHNT) program. A scan is then performed on all TMRs. When a TMR is active, PATHNT is entered and will idle all A-links in AUDMAP specified by the TMR. NETMAP is checked to ensure that the proper bits are set. When a TMR is inactive, a test is made for a junctor out-of-service condition. When one is found out of service, the appropriate B-link status bit is idled in AUDMAP and verified as being set in NETMAP. A scan of the link out-of-service list is then made and the proper bits are verified as being set in NETMAP and idled in AUDMAP.

When these steps are completed, AUDMAP is scanned and should contain all 0s. Any logic 1 found is considered an error in NETMAP so the corresponding bit in NETMAP is cleared. The RCOVRY AU NW message is printed out identifying the corrected location. The flag for PATHNT is cleared and return is made to base level processing.

2.30 The network map regeneration process starts by clearing any routine audit progress flags and inhibiting real time breaks. The NETMAP is then set to all 1s. A scan of TMRs is initiated and all path status bits attributable to active TMRs are idled in NETMAP by PATHNT. The out-of-service circuit B-link status bits from any out-of-service TMRs are also idled. The out-of-service list is then scanned and the appropriate status bits attributable to entries in the list are idled in NETMAP. When this is complete, the correct bits in NETMAP should be 0 and all others set to 1. The words in NETMAP are now complemented resulting in the correct bits being set and the regeneration being complete.

J. Main Store Audit (MASAUD)

2.31 The auditing of the main store is performed by common system programs. In the SO-2 Issue 4A generic, the subroutine MASAUD in the program AUDITS acts as a control point to speed up the process. The program sets up for a burst of 60 calls to the audit subroutines. This is accomplished by setting up for six calls per cycle over ten cycles. In the 3E3 generic, control for main store audits comes from the DSPTCHER subroutine in CMMON.

2.32 The actual audit is performed by a call to the audit update of main store (AUUPDMAS) routine in the common base level monitor (CBLM) program which checks if the audit is allowed and sets it in-progress when appropriate. Control is then passed to the audit main store block (AUMASBLK) subroutine in the main store access (MASACS) program. This subroutine scans a block of addresses both on-line and off-line for parity and data match and reports errors back to AUUPDMAS as correctable or noncorrectable. AUUPDMAS will update the error record, get any uncorrectable errors and analyze them, and print out an error report which identifies the incorrect data. When all audits are complete, the system automatically begins the cycle over.

K. Maintenance Audits (MAUD)

2.33 In addition to the audits described in this document, there are maintenance audits (MAUD) which may be performed as required. The purpose of MAUD is to provide a check on those counters, bits, and maps that may, through normal maintenance functions, become incorrect as a result of errors encountered. If a counter, bit, or map being checked by MAUD is found to be incorrect, the audit performing the check will correct the counter, bit, or map in error and/or issue a TTY message describing the problem found or corrected. A list of the counters, bits, and maps that are audited by MAUD is found in Table A.

2.34 The MAUD routine can be invoked automatically, under the control of the multiscan function (MSF), or manually by a TTY input message. When invoked automatically, MAUD will perform all of the maintenance audits.

Program Organization

2.35 MAUD is divided into four separate and distinct audits:

- (1) Trunk and service circuit out-of-service counters audit
- (2) Power alarm and miscellaneous scan point ignore bit audit
- (3) Equipped peripheral decoder (PD) map audit
- (4) Plug-up and line timing table audits.

Trunk and Service Circuit Out-of-Service Counters Audit

2.36 The entry point for this routine is MAUDPE1. If this routine is requested manually, the TTY input message AU:MAINT TSV! is used.

2.37 This audit determines whether the maintenance trunk and service circuit out-of-service counters are correct or incorrect. This routine performs an audit on the following counters.

- (a) TSVLIM—These are the trunk and service circuit out-of-service limit counters. These counters hold the current number of trunk and service circuit groups that have reached or exceeded their circuit out-of-service limit.

(b) TRKOOSC—This is the trunk out-of-service counters. It holds the current total number of trunks that are marked out-of-service.

(c) SVCOOSC—This is the service circuit out-of-service counter. It holds the current total number of service circuits that are marked out-of-service.

(d) MBC—This is the maintenance busy counter. This counter has the correct number of circuits in a group that is out-of-service. There is a maintenance busy counter for each group.

(e) NIS—This is the not-in-service bit for a group. It indicates that all the circuits of a group are out of service.

2.38 To determine whether the counters stated in paragraph 2.37 are correct, the audit must redetermine each count and compare it to the current count of each counter. This is accomplished by examining the state (paragraph 2.39) of each trunk and service circuit. If the circuit is in one of the out-of-service states, the appropriate audit count is incremented. Once all the trunks and service circuits have been checked and appropriate audit counts incremented, the audit counts are compared to the current counter count. Then, if any of the current counts differ from the audit counts, the current count is changed and a TTY output message describing the counter in error is issued. See the No. 3 ESS output manual (OM-3H300) for a description and explanation of the TTY output messages generated by MAUD.

2.39 The state of a circuit is defined by the values of the two terminal status bits, the selection bit, and the ignore and last look bits. The circuit states are:

- (1) Invalid
- (2) HAW—High and Wet
- (3) DSA—Disabled
- (4) LKO Busy—Locked-Out Busy
- (5) LKO Idle—Locked-Out Idle
- (6) ACT Busy—Active Busy
- (7) ACT Idle—Active Idle

States 1 through 5 are the out-of-service states.

2.40 The invalid state indicates that the bits used to define a circuit state cannot be resolved. If this state is encountered when the state of a circuit is being checked, the audit will issue a TTY output message describing the circuit in the invalid state and attempt to put that circuit in the active idle state. If the attempt to idle the circuit fails, a TTY output message describing the failure is issued.

Power Alarm and Miscellaneous Scan Point Ignore Bit Audit

2.41 The entry point for this routine is MAUDPE2. If this routine is requested manually, the TTY input message AU:MAINT PAIG! is used.

2.42 This audit determines if the maintenance power alarm counters and miscellaneous scan point ignore bits are correct. This routine performs an audit on the following counters.

(a) P24VPWRS—Plus 24 volts power state.

(b) MISPWR—Maintenance power alarm counters.

Word 0 = Major power alarm count;

Word 1 = Minor power alarm count;

Word 2 = Fuse alarm count.

(c) MSCAN—Miscellaneous scan point ignore bits.

2.43 To verify the counts of the power alarm counters and +24 volt power state, this audit must redetermine the +24 volt power state and each of the power alarm counts. Then the audit counts and power state are compared to the current power alarm counts and power state. If any of the current power alarm counts or the +24 volt power state is found to be incorrect, the audit will correct that counter or power state and issue a TTY output message describing the counter or power state in error.

2.44 To redetermine the power alarm counts and +24 volt power state, this audit compares the last-look bit of each power alarm scan point to its normal state. If the last-look bit is not equal to the normal state, an audit power alarm count

is incremented that reflects the type of power alarm that has been found. A TTY output message is also issued that describes the power alarm scan point which was found to be off normal.

2.45 Because the power alarm scan points needed to determine the power alarm counts are part of the miscellaneous scan points and the ignore bits of all the miscellaneous scan points are also being audited, the audit checks the ignore bit of a scan point before determining if that scan point is a power alarm scan point. This check is performed to conduct the ignore bit audit and to ensure that the power alarm count audit will be correct. If the ignore bit of a power alarm scan point is incorrect, it is possible that the power alarm scan point being checked was not checked by normal maintenance functions which count power alarms. Therefore, if any ignore bit error is encountered, the power alarm audit is inhibited, and only the ignore bits are checked thereafter. For each ignore bit error found, a TTY output message describing the scan point will be issued. When all ignore bits have been checked, a real-time break is taken so that the scan points can be updated. Upon reentry to this audit, the power alarm counter audit is restarted.

Equipped Peripheral Decoder (PD) Map Audit

2.46 MAUDPE3 is the entry point for this routine. If this routine is requested manually, the TTY input message AU:MAINT PDM! is used.

2.47 This audit performs a check of the equipped PD map to determine if the map reflects the hardware supplied. Specifically, the number of vertical matrix packs (FC206) supplied in the peripheral control frame are counted.

2.48 The equipped PD map (EQPT_PD) is 32 words long with 16 bits per word. Each word represents a vertical (matrix pack) of the peripheral pulse distributor (PPD) selection matrix and each bit of a vertical represents a horizontal. Therefore, if a vertical word has one or more bits set, the corresponding matrix pack must be supplied in the counter frame. Words 0 through 15 represent verticals 0 through 15 (matrix packs) on control frame 0, and words 16 through 31 represent verticals 0 through 15 on control frame 1.

2.49 To verify the equipped PD map this audit examines each vertical word in the map

and tests to determine if the vertical is zero. If the vertical is zero, the audit sends a zero to a PD selected by the vertical in question. If the response from the PPD is a valid reply, a vertical pack is present that is not reflected in the map. The audit issues a TTY output message describing the vertical word of the map in error. If the response is not a valid reply, there is no vertical pack present and the map is correct.

Plug-Up and Line Timing Table Audit

2.50 MAUDPE4 is the extra point for this audit. If this routine is requested manually, the TTY input message (AU:MAINT PULT!) is used.

2.51 This audit performs a check on each entry in both the plug-up and line timing tables to ensure each entry (TEN) is in the maintenance busy (MB) state. If an entry is found that is not in the MB state, it is removed from the table and a TTY message is printed indicating the TEN removed.

3. GLOSSARY

3.01 The following terms, abbreviations, and definitions are used in this section to describe audits.

ACODE—Audit code bits in TCR used for verifying TCR contents.

ACT—The active bit in the terminal memory record (TMR). Indicates an active TMR when set.

AMAINIT—Bit in 3E3 generic TMR that indicates when automatic message accounting (AMA) has been initialized.

ANI—Automatic number identification.

ANSW—Bit in 3E3 generic TMR that indicates if an answer has occurred.

APORT—Bits in TMR that identify the port of the calling party.

ASPLIT—Bit in TMR that indicates if the calling party is not connected to a junctor.

AUDCNTL—The audit control data block.

AUDCOUNT—Denotes the number of the audit currently in progress.

AUDFORCE—The field in the audit control blocks which contains the count of the number of base level loops without an audit (SO-2 Issue 4A generic).

AUDIDENT—The audit identifier is the block in the audit control block which contains the two character identifier of the current audit.

AUDMON—The audit monitor subroutine in the AUDITS program which controls and schedules the audits.

AUDPM—The audit progress mark field in the audit control block used to keep track of the audit and furnish return points from real-time breaks.

AUDTABLE—The data table which defines all audits in numerical sequence.

BACTION—Bit in TCR which invokes base level.

BASEPM—Base level progress mark.

BPORT—Bits in TMR that identify the port of the called party.

BSPLIT—Bit in TMR that indicates if the called party is not connected to a junctor.

CALLFOR—Bit in TMR that indicates if the connection was forwarded to.

CBLM—The common base level monitor program.

CCPM—Custom calling progress mark.

CONHOOK—Bit, in 3E3 generic TMR, used to notify the local coin overtime that a call waiting customer is on-hook.

CRB—Bit in 3E3 generic AUDCNTL that returns control to MMONA.

CWA—Bit in TMR that indicates the calling party is involved in call waiting.

CWB—Bit in TMR that indicates the called party is involved in call waiting.

DSA—Disabled.

ESS—Electronic Switching System.

HAW—High and wet.

JOBADDR—A table in the MMONA program that list the deferrable jobs (call processing audits, main store audits, etc) with associated parameters, ie, priority and minimum rate of repetition.

JSN—Junctor switch number.

LINE A—Line bit set when A-party (calling party) is a line.

Line B—Line bit set when B-party (called party) is a line.

LKO—Locked-out.

LKSR—Indicates a shared link.

MB_AMA—Bit set when the automatic message accounting link is maintenance busy.

MBC—Maintenance busy counter.

MBUSY—The maintenance busy status bit in the test vertical status register.

MISPWR—Maintenance power alarm counter.

MSCAN—Miscellaneous scan point.

MSF—Multiscan function.

NETMAP—The data area in store containing the active network information.

NIS—Not-in-service bit.

OSLST—Out-of-service list.

PD—Peripheral decoder.

PERM—Bit in TCR that indicates the call is receiving permanent signal treatment.

PERTN—Peripheral return bit in the TCR.

PPD—Peripheral pulse distributor.

P24VPWRS—Plus 24-volt power state.

REVRS—Bit set if path is from line B to line A.

SPLTONE—Bit in TCR set for when special tone is received.

SPN—Scan point number.

STAB—Bit in TMR set if call is stable.

SVCOOSC—Service circuit out-of-service counter.

TCR—Transient call record.

TEN—Terminal equipment number (office equipment number).

TM—Test multiple.

TMR—Terminal memory record.

TMRPM—Bit in TMR that indicates if the CCPM is active.

TRKOOSC—Trunk out-of-service counter.

TSVLIM—Trunk and service circuits out-of-service limit counters.

TV—Test vertical.

2PTSVC—Bit set in TMR when the B-party SPN is actually the second port terminal equipment number.

AUDCOUNT				UNUSED								AUDPM			
AUDPM															
AUDFORCE (COUNT OF BASE LEVEL SCANS WITHOUT AUDITS)															
AUDIDENT (2 CHARACTER IDENTIFICATION OF AUDIT IN PROGRESS)															
SPACE AVAILABLE TO AUDIT PROGRAMS															
SPACE AVAILABLE TO AUDIT PROGRAMS															
SPACE AVAILABLE TO AUDIT PROGRAMS															
SPACE AVAILABLE TO AUDIT PROGRAMS															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Fig. 1—Audit Control Block (AUDCNTL), SO-2 Issue 4A Generic

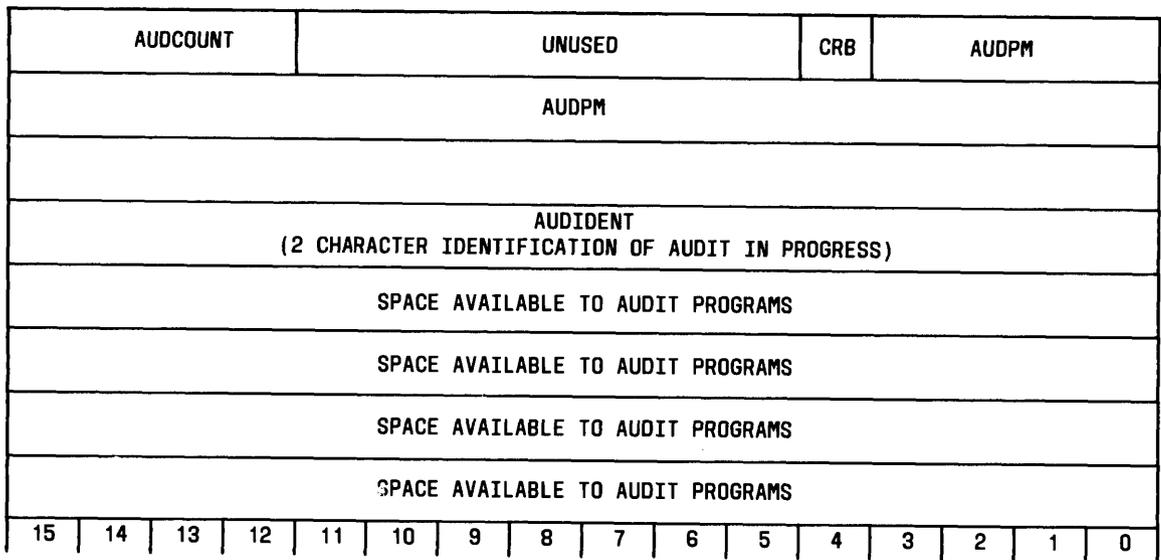


Fig. 2—Audit Control Block (AUDCNTL), 3E3 Generic

AUDIT TABLE (AUDTABLE)

AUDIT NUMBER AND TITLE	NAME	ID
0 UNUSED	AUDDONE	
1 STABLE TMR AUDIT	STABLEAUD	ST
2 TRANSIENT CALL AUDIT	TRANSAUD	TR
3 BUSY LINE AUDIT	BLAUDIT	BL
4 BUSY SERVICE CIRCUIT AUDIT	BSCAUD	BS
5 BUSY TRUNK AUDIT	BTRKAUD	BT
6 TEST VERTICAL AUDIT	TVAUD	TV
7 ROUTINE NETWORK AUDIT	RUTAUD	NW
* 8 MAIN STORE AUDIT CONTROL	MASAUD	MS
9 UNUSED	AUDDONE	

* ONLY INCLUDED IN SO-2 ISSUE 4A GENERIC

Fig. 3—Audit Table (AUDTABLE)

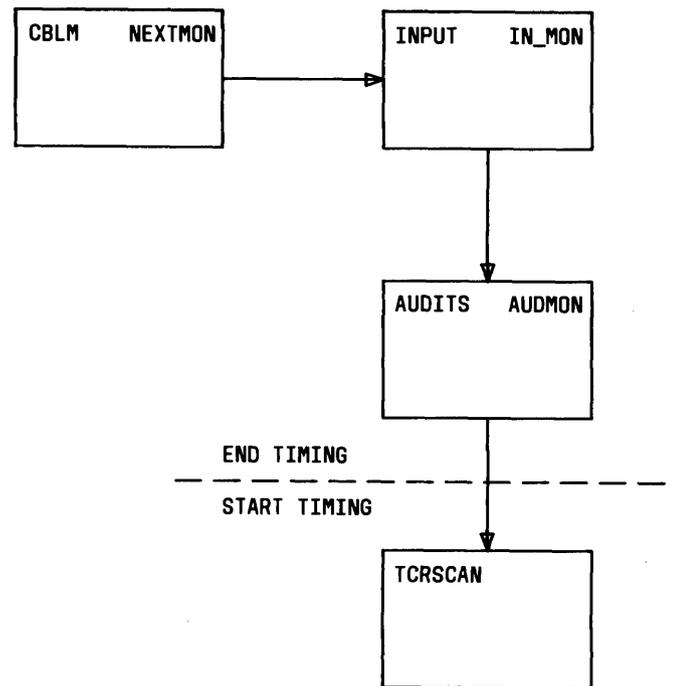


Fig. 4—Relationship of Audits to Base Level Loop, SO-2 Issue 4A Generic

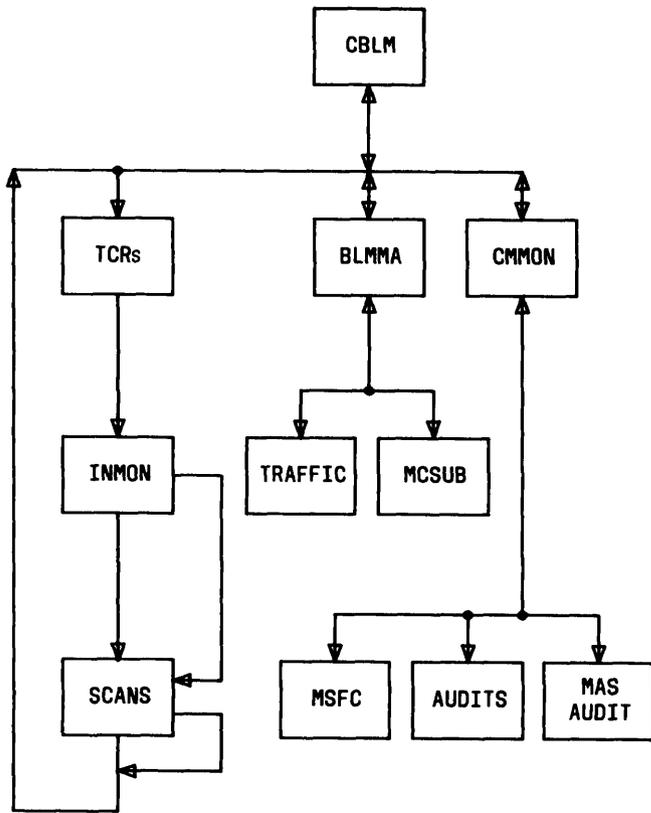
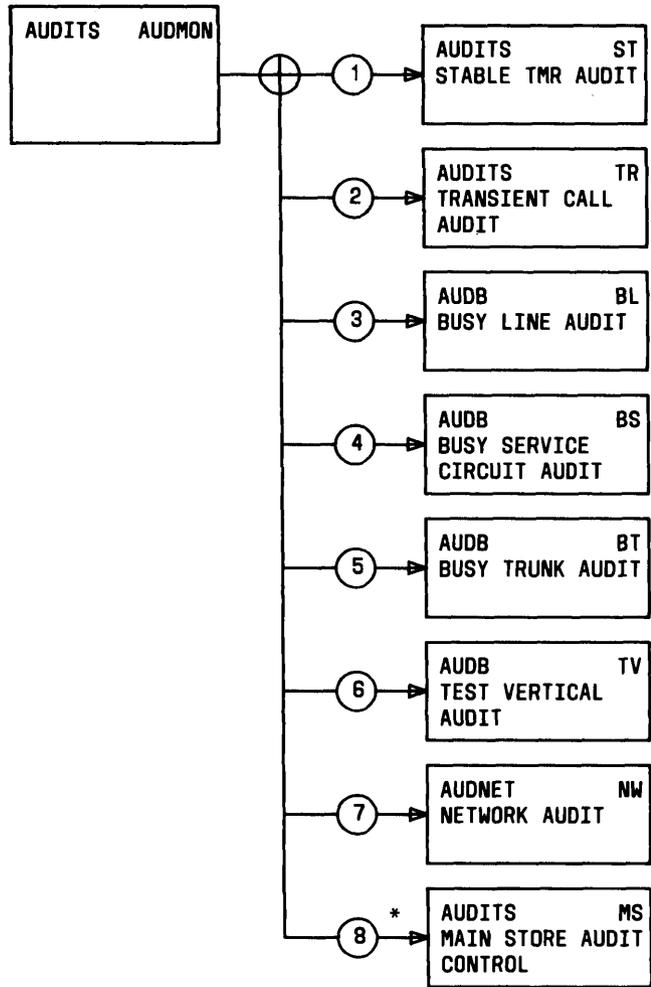


Fig. 5—Relationship of Audits to Base Level Loop, 3E3 Generic



* ONLY INCLUDED IN SO-2 ISSUE 4A GENERIC

Fig. 6—Audits Functional Sequence

00	ACT	MB AMA	LINE_A	ASPN CALLING PARTY SCAN POINT NUMBER												
01	REVRS	STAB 1	LINE_B	BSPN CALLED PARTY SCAN POINT NUMBER												
02	TIM	TIPTY	CHARG_I CHARGE INDEX				2PTSVC	TCRNUM TCR NUMBER ASSIGNED								
03	TMRPM	CALL- FOR	CWA	ASPLIT	CWB	BSPLIT	ANSW *	AMAINIT *	CONHOOK *	CCPM			APOINT		BPOINT	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

* 3E3 GENERIC ONLY

Fig. 7—Stable Terminal Memory Record (TMR)



AUDIT CODES (ACODE)

CODE	PATHS	USERS
0	NONE	DP TRUNK DURING RECEIVING
1	A	LINE OR TRUNK TO RECEIVER
2	B	NOT USED
3	A, B	NOT USED
4	T	NOT USED
5	A, T	NOT USED
6	B, T	RINGING AND OUTPUTSING
7	A, B, T	OVERLAP OUTPUTSING PRIOR TO RECEIVING

00	ACT	PER-TN	PPN PERIPHERAL WORK PROGRESS MARK						INTPM INTERRUPT PROGRESS MARK							
01	SPL-TONE	2ND TRY	ACODE AUDIT CODE			BAC-TION	PERM	BASE PM BASE LEVEL PROGRESS MARK								
02	TIMER															
03	USED FOR PERIPHERAL ORDERS															
04	USED FOR PERIPHERAL ORDERS															
05	LKSR A	ALINE 1=LN	A-PARTY A-PARTY TERMINAL EQUIPMENT NUMBER													
06	REVRS A	AJCTR A-PARTY JUNCTOR SWITCH NUMBER					ASVC RECEIVER NUMBER (CDPR OR MFREC)									
07	LKSR B	BLINE 1=LN	B-PARTY B-PARTY TERMINAL EQUIPMENT NUMBER													
08	REVRS B	BJCTR B-PARTY JUNCTOR SWITCH NUMBER					BSVC RINGER OR TRANSMITTER NUMBER									
09	REVRS T	TJCTR TALK PATH JUNCTOR SWITCH NUMBER					TIP PTY	TO PTY	CN	MSG	HOMO	FLAGS				
* 10	REVRS P2	P2JCTR 2ND PATH JUNCTOR SWITCH NUMBER (3-WAY)														
11	NOT USED BY AUDITS															
12	NOT USED BY AUDITS															
13	NOT USED BY AUDITS															
14	NOT USED BY AUDITS															
15	NOT USED BY AUDITS															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

* WORD 10 NOT USED BY AUDITS IN SO-2 ISSUE 4A GENERIC

Fig. 8—Transient Call Record (TCR) Layout for Audits



00	ACT	ANI OR MB	LINE_A	ASP CALLING PARTY SCAN POINT NUMBER												
01	REVRS	STAB 0	LINE_B	BSP CALLED PARTY SCAN POINT NUMBER.												
02				CHARG_I CHARGE INDEX.				2PTSVC	TCRNUM TCR NUMBER ASSIGNED							
03	TMRPM	CALLFOR	CWA	ASPLIT	CWB	BSPLIT	ANSW *	AMAINIT *	CONHOOK *	CCPM			APORT		BPORT	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

* 3E3 GENERIC ONLY

Fig. 9—Transient Terminal Memory Record (TMR)

TABLE A
MAINTENANCE AUDITS

AUDIT	DEFINITION
TSVLIM	TRKLIM—Total trunk groups that have reached their out-of-service limits SVCLIM—Total service circuit groups that have reached their out-of-service limits
TRKOOSC	Total trunk circuits out of service
SVCOOSC	Total service circuits out of service
MBC	Number of circuits in a group out of service (Maintenance busy counter)
NIS	Group not in service (maintenance busy) bit
P24VPWRS	Plus 24-volt power state
MISPWR	Word 0—Major power alarm count Word 1—Minor power alarm count Word 2—Fuse alarm count
MSCAN	Miscellaneous scan point ignore bits
EQPT_PD	Equipped peripheral decoder (PD) map
PLGP_TBL	Plug-up list table
LTIM_TBL	Line timing table