INPUT PROCESSING AND SCANNING SOFTWARE SUBSYSTEM DESCRIPTION NO. 3 ELECTRONIC SWITCHING SYSTEM

CONTENTS PAGE

CONTENTS PAGE

CONTENTS PAGE

Figures

Tables PAGE

1. GENERAL

1.01 This section describes the software required for the input processing and scanning functions of the No. 3 Electronic Switching System (ESS). The intent of this section is to describe the functions, correlating the software necessary to perform the functions. In addition, it will serve as an aid in accessing the program listings. Detailed program functions and coded software instructions can be found in the program listings.

1.02 This section is reissued to incorporate any changes necessary to make it compatible with both Issue 4 of the SO2 generic and the 3E3 generic. Since this is a general revision, change arrows have not been used.

1.03 Part 5 contains a glossary of terms, abbreviations, and definitions necessary for comprehension of the information in this document.

1.04 The following documents provide background and more detailed information about some of the operations that are briefly described in this document.

SECTION	TITLE
233-121-100	Scanner Description and Theory of Operation, No. 3 ESS
233-151-105	Call Processing Software Subsystem Description, No. 3 ESS
233-151-115	Operator Functions Software Subsystem Description, No. 3 ESS
233-151-130	Basic Call Processing Software Subsystem Description, No. 3 ESS
233-151-135	Custom Calling Software Subsystem Description, No. 3 ESS
233-151-145	Digit Processing Software Subsystem Description, No. 3 ESS

233-151-150 Translations Software Subsystem Description, No. 3 ESS

233-151-155 Peripheral Input/Output Control Software Subsystem Description, No. 3 ESS.

1.05 The software listings which contain the 3A code and comments for the input processing and scanning functions described in this section are given in Table A.

1.06 Scanning is performed for lines, trunks junctors, and service circuits to find service requests and supervisory state changes. It is performed both at base level and at interrupt level.

1.07 Scanning functions consist of monitoring the scan points and determining state changes. Scanning functions are described in Part 3 of this section.

1.08 Input processing consists of the following functions:

- Distribution of service requests and supervisory state changes to the appropriate processing routines for service of the inputs
- Saving of denied reports in a holding hopper
- Timing functions for hits, flashes, and disconnects
- Base level scan control
- Processor overload control
- Dynamic service protection

Input processing functions are described in Part 4 of this section.

2. SOFTWARE STRUCTURE

HOPPERS

A. General

2.01 Hoppers are dedicated areas of writable memory into which entries with a fixed format are made. Scanning routines make an entry into the appropriate hopper for detected scan point state changes. These entries are used by subsequent software routines. There are four hoppers used by the scanning and input processing software.

B. Line Origination Hopper

2.02 Line scanning routine LINESCAN in the base level scanning program (SCANS) reports detected line originations by making an entry for that scan point number in the line origination hopper. See Fig. 1 for the format of a scan point number. The line origination hopper is searched for service requests by the input monitor program (INPUT) in which line origination processing is

TABLE A

PROGRAM IDENTIFICATION

PROGRAM NAMES	PROGRAM TITLES	PROGRAM TITLES
INPUT	Input Monitor Program	3H160
TKPROC	Trunk/Junctor/Service Circuit Input Processing Program	3H177
LNORIG	Line Service Request Processing Routine	3H162
SCANS	Base Level Scanning Programs	3H173
FASTTK	Fast Trunk Scanning Program	3H159
CBLM	Common Base Level Monitor	1C950

initiated. An entry consists of two 16-bit words with entry format as shown in Fig. 2.

C. Input Hopper

2.03 An input hopper entry (Fig. 3) is two 16-bit words. Entries are made in the hopper for scan point state changes from trunks, junctors, miscellaneous scan points, and service circuits scanned at base level as well as for operator trunk originations found at interrupt level. The input hopper is then processed by INPUT.

D. Interrupt Hopper

2.04 The interrupt hopper is used by the fast trunk scanning program (FASTTK) to store and update information pertaining to immediate start (bylink) and operator trunk scan point state changes. The hopper also contains trunk entries supplied for stop dial and start signal detection by the outgoing call handling program (OUTCAL). Each entry is two words long. The entry format is shown in Fig. 4. The interrupt hopper directly follows the input hopper in core and is read as a part of the input hopper by INPUT when searching for interrupt entries to be processed.

E. Holding Hopper

2.05 This hopper is used to save entries from either the input or interrupt hoppers when the call processing routine denied the report. The holding hopper directly precedes the input hopper in memory.

SCANNING SOFTWARE

A. Interrupt Level Scanning Software

2.06 The categories of scan points that are interrogated during interrupt level are:

(1) Scan points of trunks placed in the interrupt hopper by OUTCAL for directed scanning for stop dial and start signals

(2) Immediate start trunks and operator trunks

(3) Supervisory scan points of customer dial pulse receivers (CDPRs)

(4) Supervisory and tone present scan points of multifrequency (MF) and TOUCH-TONE® (TT) tone receivers

(5) Tone scan points of MF and TT receivers

(6) Supervisory scan points of incoming trunks when receiving dial pulses (DPs)

(7) Polarity scan points in transmitters (DP and MF) and outgoing E and M trunks.

2.07 The program FASTTK performs directed scanning for stop dial and start signals from trunks placed in the interrupt hopper by OUTCAL. It also performs the scanning functions for immediate start and operator trunks. FASTTK scans one-fourth of the immediate start and operator trunks every 10 msec. This provides an effective rescan period of 40 msec for any given immediate start or operator trunk. Reports of detected scan point state changes are made in the proper hopper for processing. The other scan points listed in paragraph 2.06(3) through paragraph 2.06(7) are scanned in DIGPRO, the 10-msec interrupt program for digit receiving and sending. These scanning functions are described in Section 233-151-145 on digit processing functions.

B. Base Level Scanning Software

2.08 The categories of scan points that are interrogated during base level for service requests and supervisory state changes by the base level scanning program (SCANS) are:

(1) All line scan points

(2) All junctor scan points

(3) The slow scan list

(a) All trunks except immediate start and operator trunks

(b) Some service circuits

(c) Miscellaneous scan points.

2.09 Scan control is provided by the input monitor program, INPUT. If the system is not overloaded and there is enough time remaining in that base level for the scanning of lines, INPUT calls LINESCAN (a routine in SCANS) to scan for

line originations. The originations are placed in the line origination hopper for processing by INPUT.

2.10 The program INPUT calls SLOWSCAN (a routine in SCANS) to monitor scan point changes in state from trunks, junctors, and service circuits within its jurisdiction. This is done during every base level approximately every 100 msec. SLOWSCAN makes an entry for changes in state in the input hopper for processing by INPUT.

2.11 There are directed scanning functions performed elsewhere in No. 3 ESS software on special service circuits and other scan points. These will not be covered in this section but will be covered in other sections which describe the software functions for which they are performed.

INPUT PROCESSING SOFTWARE

2.12 The program FASTTK performs some input processing functions for immediate start and operator trunks at interrupt level. However, disconnects and operator trunk originations are processed by INPUT during base level.

2.13 The program INPUT performs input processing functions for the other trunks, junctors, and service circuits and consists of three segments or modules:

(1) Trunks, junctor, and service circuit input processing

(2) Line origination processing

(3) Scan control.

The first segment, trunk/junctor/service circuit input processing, consists of the following three sections:

(1) Interrupt level scan input processing

(2) On-hook hit scan

(3) Valid input processing for the trunk, junctor, and service circuit input hopper.

The program TKPROC is a program given control by the first segment to handle the inputs. The line origination hopper is serviced in the second segment of INPUT by giving control to LNORIG, the program which processes line originations. In addition, INPUT performs some timing functions as well as dynamic service protection (DSP) and processor overload control. The third segment performs base level scan control by calling LINESCAN (if time permits) and SLOWSCAN, the two scanning routines in SCANS.

2.14 Figure 5 shows a general diagram of the major functions and software required for input processing and scanning.

3. SCANNING

GENERAL

3.01 Scanning functions can best be described by dividing them into those which are performed during interrupt level (paragraphs 3.08 through 3.32) and those which are performed during base level (paragraphs 3.33 through 3.49). Scanning consists of the monitoring of scan point changes in state and the reporting of detected state changes by making entries in the appropriate hoppers for further processing.

3.02 Scanners are organized into rows with 16 points per row. The scanner referred to in this document is the "logical" scanner. Lines can appear on scanners 1 through 15. The last four rows (28 through 31) on each scanner are allocated for assignment to trunks and miscellaneous service circuit scan points.

3.03 In addition to trunks, regular ring and full-feature ring circuits must be scanned periodically. To limit the slowscan list entries required for this purpose, it is necessary that in a given row (rows 6 through 15 of logical scanner 0), all regular ring and full-feature ring supervisory scan points must be grouped together. Trunk scan points other than those for immediate start and operator trunks can be grouped with the ring circuits if necessary.

3.04 A trunk scanning table is used to determine which trunks are to be scanned at interrupt level and which trunks at base level. The trunk scanning table contains a group of three words of data for each of the last four rows of every logical scanner (rows 28 through 31; see Fig. 6). The three words of data are used to mask out scan points which are not of interest to a particular routine. The masks are constructed in a manner in which a 1 indicates that a particular trunk type

utilizes the scan point held by that bit position. (See Fig. 6.)

3.05 The first word of the group contains the bylink and operator trunk mask which may be applied to obtain all of the scan points which contain bylink or operator trunks. This is the principle mask used by FASTTK. The second word of the group contains the operator trunk mask. This mask is applied to obtain scan points which belong to operator trunks which use multiple wink or inband signaling. The third word of the group contains the base level trunk scanning mask. This mask is not used by any of the interrupt level programs.

3.06 The masks are an integral part of the trunk scanning task since they determine which rows and scan points will be serviced by interrupt. It is important to note that both interrupt and base level masks are never set for scanning the same scan point.

3.07 The circuit status table in writable memory consists of 2-word entries. Each entry corresponds to one circuit scan table entry. The first word of the entry consists of the last look bits which indicate the states of the circuits in a scanner row the previous time they were scanned. The second word consists of the ignore bits which indicate those circuits in a scanner row to be ignored at that time.

INTERRUPT LEVEL SCANNING

A. General

3.08 Scanning functions for immediate start and operator trunks and for trunk entries made by OUTCAL in the interrupt hopper are performed during interrupt level by FASTTK. These tasks are performed at interrupt level because they must be performed more frequently and accurately than is possible at base level.

3.09 The program FASTTK is entered each interrupt after DIGPRO. The low end of the on-hook hit timing for operator trunks is performed by this routine (paragraphs 4.05 through 4.07) and control is then passed to the task distributor.

3.10 The task distributor in FASTTK uses the two low order bits of the system timer to access one of the following groups of software functions performed each interrupt period.

(1) Disconnect check
Immediate start and operator trunk scan (row 28)

(2) Ringing and tone plant functions
Stop dial and start signal check
Immediate start and operator trunk scan (row 29)

(3) Reset operator and immediate start overload protect bits
Immediate start and operator trunk scan (row 30)

(4) Ringing and tone plant functions
Stop dial and start signal check
Immediate start and operator trunk scan (row 31).

Each of the preceding groups is processed every fourth interrupt.

B. Start and Stop Signal Detection

3.11 Start dial and stop dial signal scanning is performed every 20 msec. It is a directed scan, meaning that only the trunks provided by the outgoing call handling program (OUTCAL) are scanned. During base level, OUTCAL makes entries in the interrupt hopper giving the scan point numbers of the trunks to be scanned. FASTTK then searches the interrupt hopper for these entries and sends a directed scan order, using the scan point number (SPN) to determine the scanner and row to be interrogated.

3.12 A stop dial signal, an on-hook to off-hook state change in the trunk, is detected when the status of the scan point is off-hook and the stop signal bit (STOPSIG) in the interrupt hopper entry is zero. FASTTK then sets STOPSIG to one to indicate that a stop dial signal has been detected.

3.13 A start dial signal, an off-hook to on-hook, is detected when the status of the scan point is on-hook, the stop signal bit is one with the start signal bit (STRTSIG) being zero, and the previous off-hook period was the required length to be a stop dial signal. FASTTK then sets the STRTSIG

bit to one to indicate that a start dial signal has been detected.

3.14 An off-hook period shorter than the required length for a stop dial signal is interpreted as a hit; therefore, the STOPSIG bit is reset to zero to indicate that a valid off-hook transition has not been detected.

3.15 The program FASTTK performs timing functions for off-hook periods and for entries to maintain the integrity of the hopper in case OUTCAL loses track of an entry. The entry is removed from the hopper if for some reason it remains there too long.

C. Overload Protect During Interrupt Scanning

3.16 Overload protect prevents FASTTK from recognizing more than one operator trunk origination during a 40-msec period. In addition, the recognition of only one immediate start trunk origination or one operator trunk wink is permitted during the same 40-msec period. This prevents FASTTK from overloading the system. The word containing the overload protect bits is accessed and FASTTK resets the bits to zero during the third interrupt of each 40-msec period. Then the scanning software in FASTTK sets the proper bits when that type of input is detected and uses them to control the number of inputs to be recognized.

D. Immediate Start And Operator Trunk Scanning

3.17 Scanning for immediate start and operator trunks is performed every interrupt. One-fourth of the immediate start and operator trunks is scanned by FASTTK each 10-msec interrupt so that all are scanned in a 40-msec period. The low two bits of the system timer are used to determine which row to scan during the present interrupt and to access the proper entries in the trunk scan table and in the circuit status table for the scanner row. (For instance, row 28 of scanners 0 through 15 is scanned during interrupt 00, row 29 during interrupt 01, row 30 during interrupt 10, and row 31 during interrupt 11.)

3.18 The number of the highest logical scanner which contains immediate start or operator trunks is retrieved from the translation data and is used to determine the rows with which FASTTK is concerned. FASTTK determines the scanner and row to be scanned, supplies the register information for the scan order, and then sends the scan order. FASTTK then uses the row information (masks) from the trunk scanning table (Fig. 6) to determine if there are any immediate start or operator trunks in the row. If there are not, that row is ignored.

3.19 If there are immediate start or operator trunks in the row, FASTTK uses the last look bits from the circuit status table and the results of the present scan (0 = on-hook, 1 = off-hook) to detect any changes in state in the circuits of that row. If there are no changes in state, the row is ignored.

3.20 When changes in state are detected, the ignore bits are used to eliminate from consideration any circuits which are to be ignored at that time. Figure 7 shows an example of a scanner row status and its last look and ignore bits. Assuming the row masks indicate slowscan trunks start in bit 11, only the changes in bits 4 and 0 are detected as input from immediate start and operator trunks. After the ignore bits have been applied, FASTTK determines the SPN of each circuit with a detected state change and determines whether the circuit is an operator trunk or an immediate start trunk. Control is then given to the appropriate software dedicated to handling either operator trunk changes or immediate start trunk changes. The software reports the detected state changes in the appropriate hopper.

3.21 When an ***immediate start trunk*** changes states, it is analyzed to determine the type of change (on-hook to off-hook or off-hook to on-hook). If the change was from off-hook to on-hook, FASTTK makes an entry in the interrupt hopper if an entry is available and the last look bit is set to show an on-hook state (on-hook = 0). The hit timer in the entry is also initialized at this time. The on-hook is either an on-hook hit (less than 240 msec) or a disconnect (240 msec or longer) which will be detected and disposed of by FASTTK (see paragraphs 4.13 through 4.15).

3.22 If the hopper is full and an entry cannot be found, the last look bit remains unchanged (bit = 1) to show an off-hook state so that the change can be detected during the next scan 40 msec later. FASTTK then continues to look for additional changes in state.

3.23 When an immediate start trunk changes state from on-hook to off-hook (possible origination), the interrupt hopper is checked to determine if the trunk was being timed for disconnect (ie, has a matching on-hook entry for that SPN). When the off-hook is from a trunk which was on-hook for less than 240 msec, the off-hook is assumed to be the end of an on-hook hit. FASTTK then removes the on-hook entry from the interrupt hopper and updates the last look bit.

3.24 Immediate start trunk off-hooks require special treatment if the trunk has reoriginated before base level has completely disconnected a previous call. Thus, when a matching on-hook entry is found in the interrupt hopper and the INT bit in the entry equals 1, the on-hook entry is a disconnect entry waiting for service by INPUT at base level. When the DONE bit in the entry is zero, base level has not hardware—idled the trunk; therefore, FASTTK must send the distribute order to idle the trunk before processing the new origination. Dial pulses cannot be collected from a trunk scan point when the trunk is still connected to a junctor. After the trunk is idled, the new origination is processed as other immediate start trunk originations.

3.25 If no matching on-hook entry is found in the interrupt hopper for the off-hook, the off-hook transition is considered to be a valid immediate start origination. No off-hook hit timing is performed for immediate start trunks. The overload protect bit for immediate start originations is checked to see if overload protect is in effect. If it is in effect, the off-hook is ignored for 40 msec because only one immediate start origination or operator wink is permitted to be processed during one 40 msec period to prevent overloading the system.

3.26 Otherwise, if overload protect is not in effect, FASTTK obtains a transient call record (TCR) for the origination. It places the BIOFF base level progress mark in the TCR and sets the TCR timer so that the BIOFF routine in TKORIG, the trunk origination program, will get control of processing the call at base level. It also acquires an entry in the dial pulse receiving trunk hopper so that digit reception can proceed at interrupt level. FASTTK sets the ignore bit so that dialing will not be interpreted as changes in state. It also sets the overload protect bit to one to prevent more than one bylink origination or operator trunk wink from being recognized in one 40-msec period. When either a TCR or a dial pulse receiving trunk hopper entry is not available, the origination is ignored until the next scan.

3.27 When an ***operator trunk*** changes states, it is analyzed to determine the type of change (on-hook to off-hook or off-hook to on-hook) so that a proper hopper entry can be made. This is done by examining the last look bit. An on-hook transition from an operator trunk is either a disconnect, an on-hook hit, or an operator wink. FASTTK makes an entry in the interrupt hopper for the trunk SPN if an entry is available. It initializes the hit timer in the entry. Hit timing and disconnect checks are performed by FASTTK (see paragraphs 4.05 through 4.10).

3.28 If an off-hook transition is found in an operator trunk, FASTTK searches the interrupt hopper for an on-hook entry with a matching SPN. If a matching entry is not found, the off-hook is a possible origination; therefore, the overload protect bit is checked to determine if overload protect is in effect. If it is in effect, the off-hook is ignored until the next scan 40 msec later.

3.29 If overload protect is not in effect, FASTTK searches for an available entry in the trunk/junctor/service circuit input hopper (paragraph 2.03). If an available entry is found, the SPN of the trunk and the time that the entry is made are placed in the hopper in order that it can be passed to the input monitor (INPUT) for input processing including off-hook hit timing and origination processing. The ignore bit is set to one so that future off-hooks are not considered originations and the overload protect bit is set to prevent the recognition of more than one operator trunk origination during a 40-msec period. If the input hopper is full and no entry is available, the off-hook is ignored and the last look bit is not changed so that the off-hook transition can be recognized during the next scan 40 msec later.

3.30 If a matching on-hook entry is found in the interrupt hopper for the trunk with the off-hook transition, the off-hook is either the end of an on-hook hit or the end of a wink. If the entry has not passed hit timing, the off-hook is interpreted as the end of an on-hook hit. The on-hook entry is removed from the interrupt hopper and the last look bit is updated.

3.31 The off-hook is interpreted as the end of a wink when the entry has passed hit timing. This means the on-hook period is greater than that interpreted as a hit. An operator wink is an uninterrupted on-hook in the range of 50 to 160 msec. Also, an on-hook in the range of 10 to 50 msec may be interpreted as a wink if FASTTK sees the change to on-hook twice. An operator wink is a form of signaling when something must be done. Only one operator wink or one bylink origination is recognized in a 40-msec period so a check is made to determine if overload protect is in effect. If it is not in effect, a TCR is obtained and the SPN is supplied. FASTTK then stores the base level progress mark in the TCR to indicate that control is to be given to the operator calls program (OPER). In addition, FASTTK obtains an entry in the dial pulse receiving trunk hopper and prepares it for multiple wink reception by the digit receiving and sending program DIGPRO. FASTTK then indicates that one wink has been received. It sets the ignore bit and the overload protect bit for operator trunk winks and immediate start originations.

3.32 When there is no TCR or hopper entry available or when an operator trunk sends a wink during a period when overload protect is in effect, the operator trunk is ignored for a specified period of time by setting the timed ignore bit in the interrupt hopper entry to one. The operator must then regenerate the signal until a TCR is available or until overload protect is not in effect. The ignore bit is set to one and the last look bit is set to indicate the off-hook state. When the timer reaches zero, the ignore bit is reset to zero and the entry is removed from the interrupt hopper.

BASE LEVEL SCANNING

3.33 The program SCANS is the base level scanning program. It contains two major scanning routines, LINESCAN and SLOWSCAN. LINESCAN performs the scanning functions for lines, while SLOWSCAN scans trunks (except immediate start and operator), junctors, and service circuits for scan point state changes.

A. Line Scanning

3.34 The routine LINESCAN is called by INPUT during a base level loop if time permits (paragraphs 4.03 and 4.04) to scan for new requests for service. It scans all line scan points and reports requests for service by placing the SPN of the line in the line origination hopper for processing by INPUT. If the hopper becomes full, scanning stops during that base level loop and resumes the next base level loop at the point where scanning was stopped. Otherwise, the highest equipped scanner number is obtained from the office data, and scanning begins at that point and continues from the highest scanner to the lowest. In addition, before scanning each scanner, LINESCAN performs an "all ones" and "all zeros" test to ensure that the scanner is working properly.

3.35 Lines can appear on logical scanners 1 through 15 in rows 0 through 23. There are two status bits per line in memory which are organized into two consecutive words corresponding to a scanner row. The states represented by these bits are shown in Table B. The location of the status bits for the row to be scanned is calculated and the status bits are accessed. LINESCAN then generates the information necessary for the scan order. The scan order is sent and the results are returned (1 = off-hook and 0 = on-hook).

TABLE B

LINE STATUS BITS

BIT 1	BIT 2	MEANING
0	0	Line is idle.
0	1	Line is maintenance busy.
1	0	Line is normal busy.
1	1	Line is in the high and wet state.

3.36 By using the status bits and the scan results LINESCAN looks for a valid line origination on the row. When LINESCAN finds an origination, it searches for an idle line origination hopper entry. It determines and places the scan point number in the idle entry if one is found and sets the line status bit 1 to one to show that the line is in the normal busy state. The time of the origination is approximated and stored in the hopper entry to be used for dial tone speed measurement. LINESCAN continues to look for state changes.

3.37 Lines that remain off-hook after a sequence of tones/announcements/operator challenges are placed in the high and wet state. In this state, current is flowing in the loop. An entry is made for the line in the high and wet list and the line status bits are set to indicate high and wet. Lines on the high and wet list are scanned by SCANS every scan cycle for on-hooks. When the line is on-hook, the line status bits are changed to idle and the entry for the line is removed from the high and wet list. For a multiline hunt group, the selection bits are idled. Otherwise, when the line is still off-hook, it is ignored until the next scan.

3.38 There are 10 possible entries in the high and wet list. The PUSH_HW routine in SCANS is entered from the time monitor in the application base level maintenance monitor, BLMMA, every 10 minutes. To prevent overheating of the ferrods, the entries in the high and wet list are moved down one position in the stack so that lines cannot remain high and wet more than 100 minutes. When "popped out" of the stack, the lines are placed high and dry—that is, the line ferrod is disconnected from the network by opening the cutoff contacts. Any new entries added to the high and wet list causes the length of time lines remaining on the list to be shortened.

3.39 High and dry lines are scanned every 2 minutes or 1200-scan cycles by the HDRY progress mark routine in SCANS. A timer (HWTIMER) is maintained to count the scan cycles between high and dry scans. When it is time for scanning high and dry lines, a TCR is selected and initialized with the HDRY progress mark. The BACTION bit is set to 1 so that during the next execution of the base level TCR scanning program (TCRSCN) control is given to the HDRY progress mark routine in program SCANS. HDRY searches the line status bits for all lines in the permanent signal state. Any lines in the permanent signal state and not on the high and wet list are high and dry. HDRY restores the line cutoffs and scans those lines. Lines which are still off-hook are restored to the high and dry state (the line cutoffs are opened). When the line has gone on-hook, the status bits are restored to idle. When all high and dry lines have been scanned, the TCR is cleared.

3.40 This scanning process for lines is continued until the line origination hopper is filled or until all lines have been scanned. A word in memory is set to zero when all lines are scanned and control is returned to INPUT, the calling program. However, if the line origination hopper is full, the hopper overflow counter, which is used by the dynamic service protection function in INPUT, is incremented by 1, short CDPR time-outs are initiated, and scanning is stopped. The point at which scanning is halted is saved for the next base level line scan and control is returned to INPUT.

B. Junctor Scanning

3.41 The routine SLOWSCAN in SCANS is called by INPUT once every base level loop for the scanning of junctors, most trunks (except immediate start and operator), and service circuits. The junctor scanning portion of SLOWSCAN, which is processed first, is used to scan for supervisory changes in junctor scan points. It looks at all junctor scan points and reports changes in supervision (on-hook to off-hook or off-hook to on-hook) by placing the junctor scan point number in the trunk/junctor/service circuit input hopper. If the input hopper is full, scanning stops and the remaining points are not seen until the next base level scan when scanning is started from the beginning.

3.42 Junctor scan points occupy four rows in each scanner (rows 24 through 27). These 64 scan points correspond to both sides of 32 junctors. There are two junctor status bits associated with each junctor SPN. They are the last look bit (0 = on-hook, 1 = off-hook), which indicates the state of the junctor SPN the previous scan, and the ignore bit, which indicates those junctor SPNs to be ignored. The program scans the junctor rows in reverse order and starts from the highest numbered scanner equipped to logical scanner 1.

3.43 The routine SLOWSCAN retrieves the junctor status bits for the scanner row. It issues the scan order which returns the status of the junctor scan point (0 = on-hook, 1 = off-hook).

3.44 By using the last look bits for the row and the results of the present scan, SLOWSCAN looks for changes in state. It also uses the ignore bits to mask out those junctors which are to be ignored. When it detects changes in state, it generates the SPN for the junctor and places it in an idle entry of the input hopper if one is available. It records in the hopper entry the time of the

entry and the state of the junctor scan point (ONOFF bit). The routine also updates the last look bit and continues to look for additional changes in state and to scan all the rows of junctor scan points. INPUT performs the input processing functions for junctors.

3.45 If the input hopper is full and no idle entry can be found, control is returned to INPUT, the calling program, and scanning of trunks, junctors, and service circuits is halted until the next base level loop.

C. Trunk and Service Circuit Scanning

3.46 The routine SLOWSCAN is used to scan for supervisory changes in circuits referenced by appropriately marked entries in the trunk scan table (paragraph 3.04). The table lists all circuits which must be regularly scanned, with the exception of lines, junctors, and CDPRs. It directs the scanning of operator and immediate start trunks (scanned at interrupt level) and all trunks and service circuits that are scanned at base level. Those circuits scanned at base level are the concern of SLOWSCAN.

3.47 The circuit status table (paragraph 3.07) contains the last look and ignore bits for the circuits scanned under the control of the circuit scan table. The routine SLOWSCAN scans the circuits and reports changes in the supervisory state of the circuit by placing the circuit scan point number in the trunk/junctor/service circuit input hopper (paragraph 2.03). If the input hopper is full, scanning stops and the remaining circuits are not seen until the next base level loop when the entire slow scan list is scanned from the beginning.

3.48 The circuit scanning software searches the trunk scan table for entries of interest. The software computes the row number of the trunk scan table entry of interest and determines the logical scanner. It issues the scan order. After testing the validity of the scan result, the circuit scanning software uses the last look bits and the present scan results to find changes in state. It also applies the ignore bits to mask out those circuits to be ignored.

3.49 When a change in state is detected, an idle entry if available is obtained in the trunk/junctor/service circuit input hopper to report the change for processing by INPUT. The SPN, entry time, and circuit status are stored in the entry, and the circuit's last look bit is updated. Scanning and detection of changes in state are continued. If the input hopper has no idle entries, scanning is stopped and control is returned to INPUT, the calling program. Scanning is resumed at the beginning of the list during the next base level loop as previousy stated.

4. INPUT PROCESSING

GENERAL

4.01 Input processing consists of the distribution of service requests and supervisory state changes to the appropriate processing routines to service the inputs and the control of scanning for these requests and state changes. It includes timing functions, such as timing for hits and disconnects. In addition, it includes the dynamic service protection and processor overload protect functions. FASTTK performs some input processing functions for immediate start and operator trunks at interrupt level; however, most input processing functions are performed during base level by programs INPUT, TKPROC, and LNORIG.

PROCESSOR OVERLOAD FUNCTIONS

4.02 Several measures are used to prevent the processor from being overloaded during peak periods. An overload protect function in FASTTK prevents the recognition of more than one operator trunk origination during a 40-msec period. In addition, it permits the recognition of only one operator trunk wink or one immediate start trunk origination during the same 40-msec period. This prevents overloading of the system by FASTTK. More details are given in paragraph 3.16.

4.03 Another overload protection function is performed by INPUT. INPUT stops line scanning and input processing as the time spent in the call processing portion of the base level loop grows longer. All processing of interrupt entries and on-hook hit timing for trunks, junctors, and service circuits are performed without a check of the time. However, section 3 of the first segment in INPUT, before passing each trunk/junctor/service circuit input hopper entry to TKPROC for processing, checks the time already spent in the base level loop. The time of the start of the base level loop is subtracted from the present system time. If

the result is greater than a specified interval, the processing of the entry and all other remaining entries in the input hopper, the processing of all of the line origination hopper entries, and the scanning of the lines for new service requests are bypassed and the SLOWSCAN subroutine is invoked to scan trunks, junctors, and service circuits.

4.04 Similarly, segment 2 of INPUT which processes the line origination hopper entries checks the elapsed time and abandons processing of the remaining entries if the elapsed time exceeds a specified time. Line scanning is also bypassed and SLOWSCAN is invoked. When any input hopper entries or line origination hopper entries are not processed, a routine compresses the entries still in the hoppers into the beginning part of the hoppers. This helps provide first-in first-out service.

OPERATOR TRUNK INPUT PROCESSING

A. On-Hook Hit Timing

4.05 The program FASTTK makes an entry in the interrupt hopper and increments the hit timing counter when an operator trunk on-hook transition is detected. The OPWTEST routine of FASTTK is entered each interrupt period after DIGPRO completes digit receiving. Its function is to perform the low end on-hook hit timing for operator trunk entries. The counter in memory for operator trunks needing low end on-hook hit timing is checked to see if hit timing functions are required. If the counter is zero, indicating that no timing is necessary, control is passed to the task distributor in FASTTK. Otherwise, the interrupt hopper is searched for the entries requiring hit timing. FASTTK derives the information necessary to send an order to scan the trunk scan point and then issues the scan order to determine the present state of the trunk.

4.06 If the trunk has gone off-hook, the on-hook is interpreted as a hit. The DONE bit is set in the interrupt hopper entry to show that hit timing has been performed and the hit timing counter is decremented. During the next regular 40-msec scan of the trunk, FASTTK scanning software, seeing the off-hook transition, removes the entry from the interrupt hopper (paragraph 3.30).

4.07 However, if the trunk is found to be still on-hook after the hit timing software, the on-hook is either an operator wink or a disconnect. OPWTEST then sets the bit in the interrupt hopper entry to indicate that the entry has passed hit timing. Further timing of the entry for winks and disconnects is performed by the operator trunk scanning software and the disconnect checking software in FASTTK.

B. Operator Winks

4.08 When an off-hook to on-hook transition is seen by FASTTK scanning software from an operator trunk, an entry is made into the interrupt hopper. After it has passed low end on-hook hit timing, an on-hook to off-hook transition is interpreted by the scanning software as the end of an operator wink or an operator trunk disconnect. See paragraphs 3.31 and 3.32 for operator wink treatment.

C. Operator Trunk Disconnects

4.09 Checking for disconnects from operator trunks and immediate start trunks is performed by FASTTK every fourth interrupt (the first interrupt of a 40-msec period). Also performed at the same time is the decrementing of the timer which indicates how long an entry must remain in the hopper before being eligible for disconnect (240 msec). The interrupt hopper is searched for immediate start and operator type entries. If the delay timer is zero and the entry is not a timed ignore entry, it is converted into a form acceptable to the input monitor program, INPUT, which services the interrupt hopper for disconnects. FASTTK sets the interrupt bit (INT) in the entry to show that the entry needs service and that it is an interrupt entry.

4.10 In section 1 of the first segment of INPUT, the entire input hopper including the interrupt hopper is searched for interrupt entries which need processing. Processing of interrupt entries is performed first by INPUT because fast processing of those entries is required. When an interrupt entry is found, it is passed to TKPROC, the trunk/junctor/service circuit input processing program. TKPROC, after determining that the report is an on-hook from a stable call, sends the necessary data to DISCON which performs the disconnect functions for the call and then returns to TKPROC.

TKPROC removes the hopper entry and returns to the proper section of INPUT.

D. Operator Trunk Originations

4.11 Operator trunk originations are recognized by the scanning software in FASTTK, which makes an entry in the input hopper for the origination. These inputs are handled by section 3 of the first segment of INPUT, which processes valid inputs from trunks, junctors, and service circuits. These procedures are described in paragraph 4.23.

IMMEDIATE START TRUNK INPUT PROCESSING

A. Immediate Start Trunk Originations

4.12 Immediate start trunk originations are recognized by the scanning software in FASTTK. The processing of the origination is described in paragraphs 3.23 through 3.26 since the scanning routine obtains a TCR and readies it for digit reception and call processing. No off-hook hit timing is performed. All off-hooks are considered to be originations.

B. Immediate Start On-Hook Hits

4.13 When an on-hook transition from an immediate start trunk is detected by the scanning software in FASTTK, it is placed in the interrupt hopper as described in paragraph 3.21. Then, if an off-hook transition is found for that trunk in less than 240 msec (the time required for a disconnect), the on-hook is recognized as a hit and the entry is removed from the interrupt hopper (paragraph 3.23).

C. Immediate Start Disconnects

4.14 Checking for disconnects from immediate start and operator trunks is performed every fourth interrupt (the first interrupt of a 40-msec period). Also performed at the same time is the decrementing of the timer which indicates how long an entry must remain in the hopper before becoming eligible for disconnect (240 msec). The interrupt hopper is searched for immediate start and operator type entries. If the delay timer is zero and the entry is not a timed ignore entry, it is converted to a form acceptable to the input monitor program, INPUT, which services the interrupt hopper for disconnects. FASTTK sets the INT bit in the entry to show that the entry needs service and that it is an interrupt entry.

4.15 In section 1 of the first segment of INPUT, the whole input hopper including the interrupt hopper is searched for interrupt entries which need processing. Processing of interrupt entries is performed first by INPUT because fast processing of those entries is required. When an interrupt entry is found, it is passed to TKPROC, the trunk/junctor/service circuit input processing program. TKPROC examines the DONE bit in the hopper entry to determine whether the trunk has been hardware idled. If not, TKPROC sends the distribute order to idle the trunk and sets the DONE bit. TKPROC, after determining the report is an on-hook from a stable call, sends the necessary data to DISCON which performs the disconnect functions for the call and then returns control to TKPROC. TKPROC removes the entry from the hopper and returns control to INPUT.

TRUNK/JUNCTOR/SERVICE CIRCUIT INPUT PROCESSING

4.16 The first segment of INPUT consists of three sections to perform trunk, junctor, and service circuit input processing. Scanning software reports detected changes of state by making an entry in the appropriate hopper. As already stated, the first section processes interrupt inputs. Hit timing is performed during the second section, and flashes and originations are marked reportable to be handled by the third section. The third section passes valid inputs to TKPROC, which distributes the input to call processing routines.

A. On-Hook Hit Timing

4.17 The second section of segment 1 of INPUT searches the trunk/junctor/service circuit input hopper for busy entries not already marked reportable. The entry is rescanned if it has not already been rescanned to check for short duration hits. If the states disagree, the entry is interpreted as a short duration hit and is removed from the hopper. The status bits are also updated.

4.18 When the states agree, there is no short duration hit. A check is made to see if the entry is off-hook. When the entry is an off-hook entry, the input hopper is searched for a matching on-hook entry to determine if a possible hit or flash has occurred. No matching on-hook entry means the off-hook is an origination which is

marked reportable to be handled by the third section.

4.19 When a matching on-hook entry is found, the time difference between the two entries is computed. If the difference is less than the hit time interval, the two entries comprise a hit pair and are both removed from the input hopper.

B. Flashes

4.20 If the time difference exceeds the hit time interval, the time difference is placed in the time portion of the on-hook entry, and the entry is marked reportable as a flash to be handled by the third section. The off-hook matching entry is thus removed from the hopper.

4.21 Section 3 passes the flash entry to TKPROC which determines that it is a stable call and an on-hook entry. All on-hooks from stable calls are passed by TKPROC to the disconnect program DISCON. DISCON then processes the flash and returns to TKPROC. TKPROC zeroes the hopper entry and returns to INPUT for further hopper processing.

C. Disconnects

4.22 Section 2 ignores an on-hook entry and does not mark it reportable. Section 3, when it processes the input hopper, assumes that entries not marked reportable are on-hook entries which must be checked for completion of on-hook hit timing. The software subtracts the time that the entry was placed in the hopper from the present system time to determine if the on-hook is eligible for disconnect. If it is not, the entry is ignored. However, an on-hook entry eligible for disconnect is marked reportable and is passed to TKPROC. TKPROC, after determining the on-hook is from a stable call, passes the necessary data to DISCON. DISCON does the disconnecting functions for the call and returns control to TKPROC, which clears out the hopper entry and returns control to INPUT. If DISCON could not handle the report at this time, the entry is transferred to the holding hopper and control is returned to INPUT for further input hopper processing.

D. Trunk Originations

4.23 Off-hook entries in the input hopper are marked reportable by section 2 of segment 1 of INPUT. Section 3 then searches the hopper for entries marked reportable and passes each entry to TKPROC if time permits. TKPROC searches for a terminal memory record (TMR) associated with the hopper entry. In the case of a trunk origination, there is no TMR. TKPROC then calls the trunk origination program, TKORIG, which obtains a TCR for the call and the trunk translation data. In addition, TKORIG starts the peripheral sequence to connect the trunk to a receiver if necessary and to send the wink signal. It also supplies the base level progress mark, identifying the call processing routine to be executed next, and the signal digit for the TCR. TKORIG then returns control to TKPROC. If TKORIG completed its function successfully, TKPROC clears the input hopper entry and returns to INPUT for further hopper processing. On the other hand, when TKORIG could not handle the origination, TKPROC transfers the entry to the holding hopper and returns control to INPUT.

E. Trunk Returning From Permanent Signal

4.24 An on-hook from a trunk not associated with a stable or transient call is checked by TKPROC to see if the trunk is returning from permanent signal. (The entry was passed to TKPROC by section 3 of segment 1 of INPUT.) If the on-hook is from a trunk returning from permanent signal, TKPROC removes the hopper entry and returns control to INPUT.

F. Inputs Associated With Transient Calls

4.25 The program TKPROC finds a TMR associated with the scan point in the input hopper entry. The TMR indicates that the call is a transient call; therefore, TKPROC looks for a TCR associated with the call. It then causes the base level progress mark routine to be invoked and indicates which party (APARTY or BPARTY) caused the report. Upon successful completion of the progress mark routine, TKPROC clears the input hopper entry and returns to INPUT. If the base level progress mark routine is unable to accept supervision, the entry is transferred to the holding hopper to be processed later.

G. PBX Key Reports

4.26 The program TKPROC calls a translation routine to determine whether a report is a PBX key report because it must be handled

separately. The PBX status block is prepared for further processing and the hopper entry is removed.

H. Custom Calling Entries

4.27 A report associated with a stable or transient call may be from a call in a three-way or call waiting connection. If there is an active TMR progress mark, the call is involved in a three-way or call waiting connection and control is transferred to routine CCTASK in the threeway calling program (TREWAY) for processing. When control is returned, TKPROC clears the hopper entry and returns to INPUT.

I. Miscellaneous Scan Point Reports

3H 177-02

4.28 The program TKPROC checks for miscellaneous scan point input. The miscellaneous scan points are in rows 16 through 27 of logical scanner 0. These inputs are sent to the routine MSCAN in the power and alarm scanning program, PWSC, for processing. After MSCAN has processed the input, control is returned to TKPROC which removes the hopper entry and returns to INPUT.

J. Errors Associated With Trunk/Junctor/Service Circuit Reports

4.29 The program TKPROC provides a teletypewriter message for most errors it encounters. In addition, it clears the hopper entry causing the error. Errors detected by TKPROC are:

- Wink from a trunk which should not be seen by TKPROC
- TMR for junctor report is idle
- TCR pointed to by a TMR is not active in the case of a report associated with a transient call
- An off-hook from a stable junctor.
- A report has been purged from the holding hopper because it was in there for over 1 minute or because it was the oldest entry and a new entry forced it to be purged.

4.30 If a flash report is seen from a trunk which is not associated with a stable or transient call, the report is assumed to represent the situation where a trunk had made an origination request which could not be serviced before the call was abandoned. TKPROC then clears the hopper entry. In addition, all off-hooks from trunks or service circuits associated with a stable call are ignored by TKPROC and are removed from the hopper.

LINE INPUT PROCESSING

A. Dynamic Service Protection

4.31 Segment 2 of INPUT processes the line origination hopper if enough time remains in the base level loop and passes valid service requests to the line origination program, LNORIG. In addition, it determines whether DSP should be put into effect. Dynamic service protection is an automatic way of protecting the service of class A (as opposed to class B) lines during a traffic overload. Classes of lines are assigned by an office in the translation data. When DSP is in effect, the treatment of a class A line is not affected. A class B line, however, is serviced only if it is in the first hopper entry while all other class B entries in the hopper are cleared. This results in a delay in the receipt of dial tone by class B customers. The method by which software invokes DSP is described in the following paragraphs.

4.32 The line scanning routine, LINESCAN in SCANS, increments the hopper overflow counter HOPOVCNT by one each time it finds the line origination hopper full. Segment 3 of INPUT, which does scanning control, decrements HOPOVCNT by one if it is nonzero and if segment 2 found the line origination hopper empty (LNSERV bit = 0) with enough time to invoke the line scanning routine, LINESCAN. Segment 2 of INPUT, therefore, checks the counter before processing the line origination hopper. When the value of the counter becomes greater than a specified number, DSP is invoked by setting the DSP flag bit to one (if the office has not overridden the automatic DSP function.)

4.33 Segment 2, in processing the line origination hopper under DSP, processes the first valid busy entry in the hopper regardless of the line class if enough time remains in the base level loop. It sets the LNSRV bit to one to indicate that an entry has been serviced. After at least one entry has been serviced, only class A lines are serviced under DSP. Entries not associated with

class A lines are removed from the line origination hopper by INPUT and the status bits are set to idle so that the origination will be detected again during the next scan.

B. Line Origination Processing

4.34 Segment 2 of INPUT searches the line origination hopper for valid requests for service placed in the hopper during the scanning of lines. It passes the valid busy entries to LNORIG for distribution to the call processing routines if enough time remains in the base level loop. As has already been described, DSP controls which entries are serviced when it is in effect.

4.35 There are two entry points in program LNORIG. Entry point LNORIG is for new originations, while entry point 3WAYORIG is for second dial tone on an add-on request. Both entry conditions are processed in the same manner with a few exceptions. Hit timing is not performed for add-on requests. Add-on requests already have a TCR associated with the call; therefore, no TCR selection is necessary. The peripheral action (PACT) macro used to order the connection of the CDPR to the line supplies the CDRCON base level progress mark in the TCR for add-on calls so that control is given to the CDRCON routine in program TREWAY. However, for new originations base level progress mark DIALTON is provided so that control is given to the DIALTON routine in the digit interpretation program DNTRP. Processing of new originations is described in the following paragraphs.

4.36 The program LNORIG orders the rescanning of the scan point to look for short duration hits. If the line has gone back on-hook, the off-hook entry is interpreted as a hit and is removed from the hopper; control is returned to INPUT for further hopper processing.

4.37 When the scan point state is still off-hook, LNORIG initiates the selection of a TCR for the call. When no TCR is available, the entry is retained in the hopper for the next base level loop. Otherwise, LNORIG places in the TCR the hopper entry time for the dial tone speed measurement and the scan point number, which is also the terminal equipment number of the line. It then calls a translation routine in XSLSPN to obtain translation data for the line. If the translation shows the scan point is for a trunk or service circuit, LNORIG opens the cut-off contacts for that scan point. For an unassigned line, LNORIG opens the cut-off contacts, prints a TTY message, and sets the status bits to the maintenance busy state. The TCR is cleared and control is returned to INPUT.

4.38 However, if the translation is successful, it returns the translation information in the TCR and a code which indicates the type of customer dial pulse receiver needed. The bit in the TCR which flags a two-party line is checked by LNORIG because the tests on the bits in the TCR which indicate denied originations and manual lines are not performed on two-party lines. The status bits of a line which is denied origination are set to the high and wet state and the TCR and the line origination hopper entry are cleared.

4.39 The manual line bit is one in the TCR for a manual line. If there is a line circuit, LNORIG sends the order to operate the relays. Those manual lines without hot-line service are handled by calling the route index expansion routine, ROUTE, in DNTRP. ROUTE, using the route index provided by LNORIG, calls for a route index expansion which returns a destination code in this case. The route index expansion entry, the TCR location, and the route index expansion location are passed via the destination code branch table to a routine in OPER, the operator calls program, for call processing. The hopper entry is cleared.

4.40 Lines with hot-line service (HOT bit = 1 in TCR) are handled by SPEEDIN in DNTRP. SPEEDIN begins the call processing tasks for the call and returns to LNORIG after it has completed its tasks.

4.41 Lines other than manual lines or lines denied origination must be connected to the proper type of CDPR. LNORIG calculates the trunk group number of the appropriate receiver type and then initiates the selection of an idle CDPR. If an error is detected, LNORIG clears both the hopper entry and the TCR. If no CDPR is available, LNORIG clears the TCR but retains the hopper entry for future tries. After a CDPR is selected, a path through the network from the line to the CDPR is hunted by the network path hunting program, PATHNT. If a path is not available, another CDPR is selected and a second try is made to find a path through the network. After two tries with no success, LNORIG idles the CDPR

and removes the line origination hopper entry. It also calls a failure routine to report the blockage and resets the line's status bits to show an idle state. This eliminates hot spots in the network from tying up the line origination hopper.

4.42 When PATHNT finds a path, it returns the address of the TMR associated with the selected junctor. The TMR contains the necessary information from the TCR. LNORIG then orders the peripheral action to connect the calling customer to the CDPR. The peripheral action macro loads the address of the base level progress mark routine used to give dial tone into the TCR. LNORIG removes the entry from the hopper and returns control to INPUT for further line origination hopper processing.

SCAN CONTROL

4.43 The program INPUT performs base level scan control in segment 3. It checks to see if there is enough time remaining in the base level loop to scan lines for new service requests. If not enough time remains, line scanning is skipped and trunk, junctor, and service circuit scanning is invoked. If enough time remains, the routine LINESCAN in program SCANS is called to perform the line scanning. After line scanning is completed, INPUT calls SLOWSCAN in program SCANS to perform the base level trunk, junctor, and service circuit scanning. For Issue 4 of the S02 generic, when scanning is completed, INPUT passes control to the audit monitor (AUDITS). For the 3E3 generic, when scanning is completed, INPUT passes control to the common base level monitor (CBLM).

5. GLOSSARY

5.01 The following terms and definitions are used in this section.

Base level — major software loop including all functions not done during interrupt level

BACTION bit — bit in TCR which is set to indicate that base level action is needed

Base level progress mark — indicates in the TCR the next software routine to be executed during base level

Bit (contracted from binary digit) — the binary unit of information which is represented by one of two possible conditions; such as, the digits 0 and 1, high potential or low potential, on or off

Bylink — any incoming immediate start trunk (assumed from a step-by-step office)

CDPR — customer dial pulse receiver

Clear, wipe out, or remove — to restore a storage device to the "zero" state

DSP (dynamic service protection) — an automatic way of protecting the service of class A lines during a traffic overload

High and Dry — state in which the line does not have electrical or logical supervision except once every 2 minutes

High and Wet — state in which the trunk or line is monitored for an on-hook only

Hopper — dedicated areas of writable memory into which entries with a fixed format are made

Hot line — a line with direct access to a party for which no dialing must be done

Immediate start trunks — a trunk which does not wait for a signal before beginning to send dial pulses

Interrupt (Timed) — a hardware-initiated interrupt which interrupts the base level loop every 10 msec for a period of time necessary to perform frequently required functions such as sending and receiving tasks, immediate start and operator trunk scanning, and peripheral functions

Macro — an abbreviated notation for a sequence of operations

MF — multifrequency

Operator trunk — one of five types of trunks (TSPS, TSP, toll switching, recording completing, operator office trunk)

Operator wink — form of signaling

Permanent signal treatment — a permanent signal condition arises when a line remains in the off-hook state without dialing for a long period of time. It can be caused by a line failure or the

customer leaving his receiver off-hook. Such lines are given a sequence of tones to alert the customer and eventually are put into the permanent signal state by using the status bits for the line

Scan point — ferrod sensor used in scanners for supervisory purposes

Set — to place the storage device in the "one" state

Signal digit — is used in the TCR to indicate the location of the digit to be received before base level is alerted by setting the BACTION bit that more base level action is needed

TCR (transient call record) — a 16-word block of writable main storage assigned to a call in the transient state and containing control information, terminal and path information, and receiving and sending data applicable to the call

TMR (terminal memory record) — a 4-word block of writable main storage assigned to each junctor. For stable calls, the junctor's TMR specifies the SPN of the talking parties and provides timing control. For transient calls, the TMR also specifies the TCR assigned to the call as well as the SPNs of the connected circuits

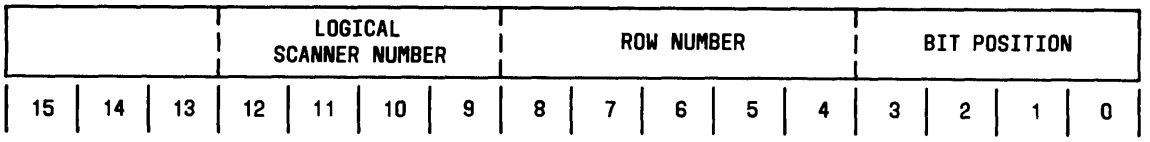
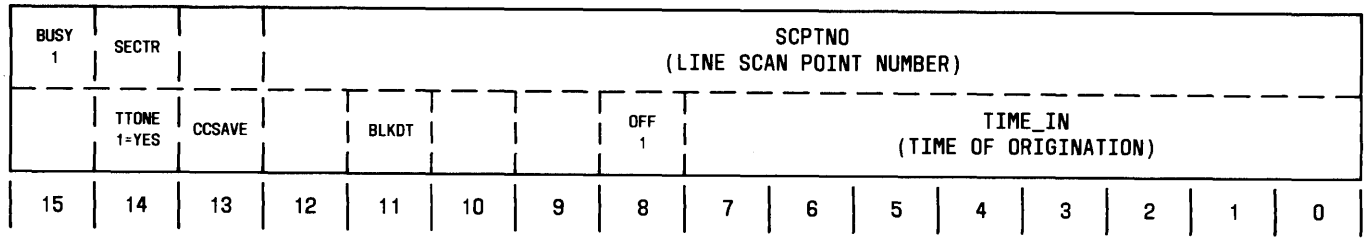
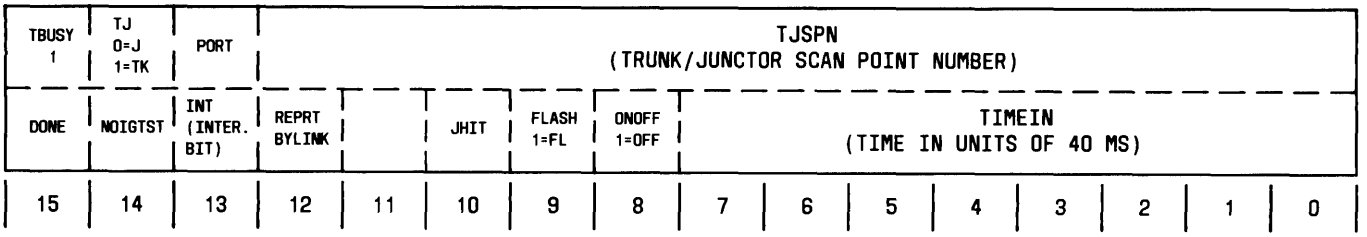


Fig. 1—Scan Point Number Format



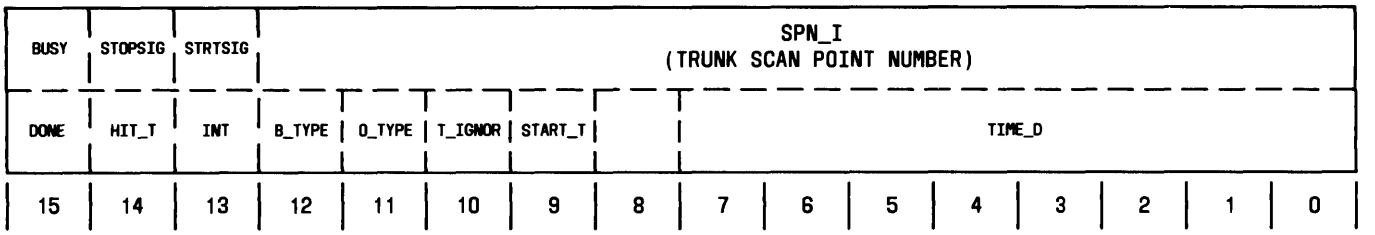
SCPTNO	Line Scan Point Number.
SECTR	Second time the SPN has been rescanned.
BUSY	=1. Set when this slot of hopper is in use.
TIME_IN	Time of origination (for Dial Tone Speed measurement). 8 bits of SYSTIME to 80 ms accuracy on entry to hopper.
OFF	=1 for off-hook.
BLKDT	Set when a dial tone blockage has occurred.
CCSAVE	Indicates that this is a custom calling origination.
TTONE	1=YES. A TOUCH-TONE receiver is attached to this line.

Fig. 2—Line Origination Hopper Entry



TJSPN	Circuit Scan Point Number.
PORT	Indicates when port of a conference circuit is in use.
TJ	1=TK.
TBUSY	1 Set when this slot of hopper is in use.
TIMEIN	Time in units of 40 ms.
ONOFF	0=On.
FLASH	1 Implies a flash was detected.
JHIT	Set when hit timing is completed.
REPRT OR BYLINK	Set when input is no longer timing entry. Also indicates a bylink trunk disconnect in INTHOP.
INT	Entry is from the interrupt hopper.
NOIGTST	Do not test for ignored SPN – used by FASTTK & TKPROC.
DONE	Used to indicate if a bylink trunk was idled.

Fig. 3—Input Hopper Entry



SPN_I	Scan Point Number of Trunk.
STRTSIG	Set when start signal is detected.
STOPSIG	Set when stop signal is detected.
BUSY	1 means this 2-word entry is in use.
TIME_D	Timer.
START_T	Set to 1 for start signal entry.
T_IGNORE	Set to 1 for timed ignore entry.
O_TYPE	Set to 1 for operator trunk on hook entry.
B_TYPE	Set to 1 for bylink trunk on hook entry.
INT	Set to 0 since interrupt is using entry.
HIT_T	Set to 1 when an operator trunk passes wink hit timing.
DONE	Set to 1 after operator trunk hit timing has been done.

Fig. 4—Interrupt Hopper Entry

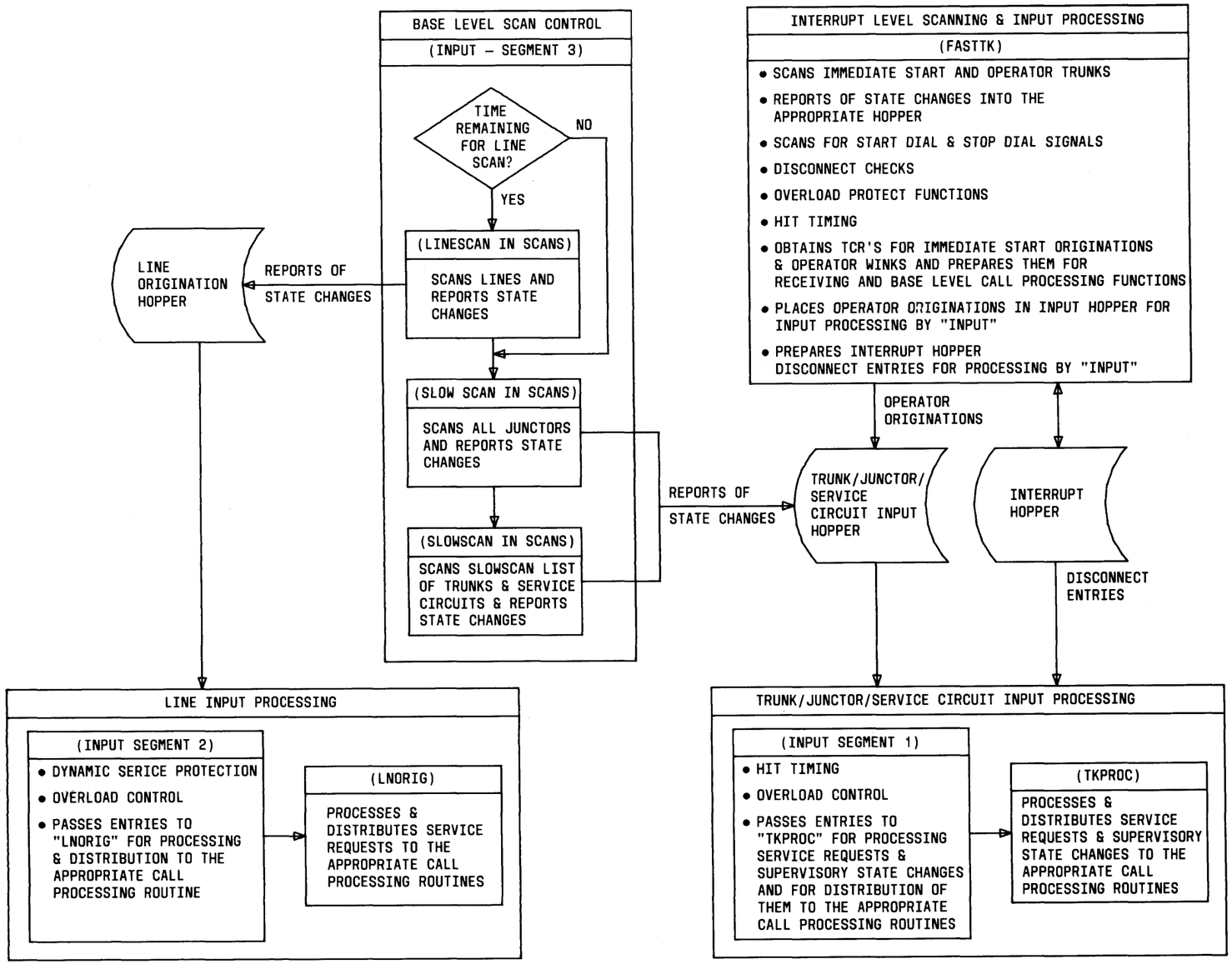


Fig. 5—General Diagram of Input Processing and Scanning Functions and Software Required to Perform the Functions

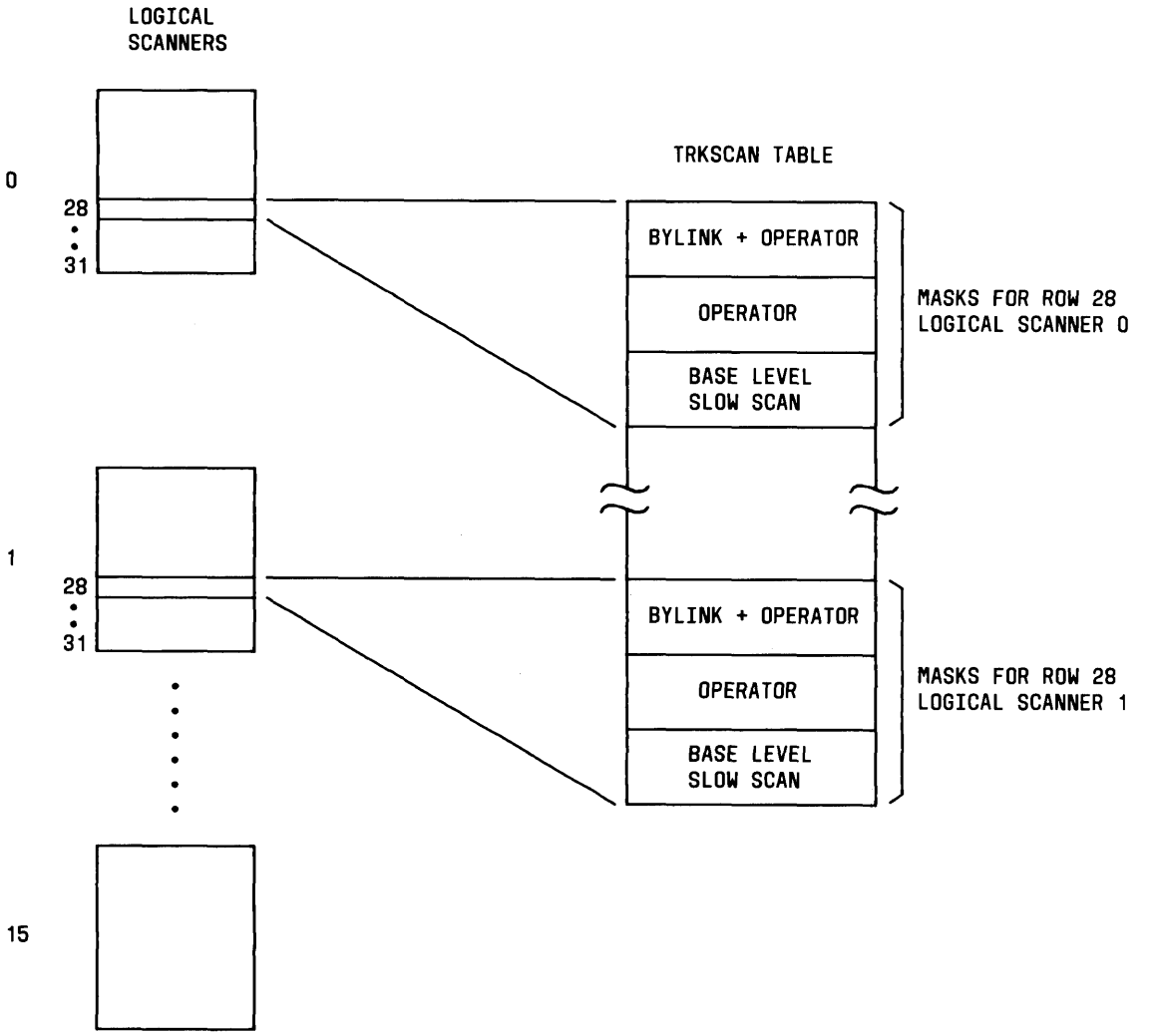


Fig. 6—Circuit Scan Table

SCANNER ROW STATUS

0	0	1	0	0	1	0	0	0	0	1	1	0	0	1	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

LAST LOOK BITS

0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

IGNORE BITS

0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Fig. 7—Example of Detection of Changes in State of a Scan Point