

TRANSLATIONS
SOFTWARE SUBSYSTEM DESCRIPTION
NO. 3 ELECTRONIC SWITCHING SYSTEM

	PAGE		PAGE
1. GENERAL	6	B. Foreign Area Table Translation Processing	16
2. BASIC TRANSLATION TABLES	7	C. Code Index Expansion Processing	16
TRANSLATORS	7	D. Screening	18
A. Master Table Index	7	E. Route Index Expansion Processing	18
B. Head Table	7	F. Charging	19
C. Subtranslators	7	5. INTRAOFFICE TRANSLATIONS	19
D. Accessing Translators	8	A. General	19
E. Expansion Tables	8	B. 4-Digit Translations	19
3. LINE ORIGINATIONS	9	6. INTEROFFICE	22
A. General	9	A. General	22
B. Scan Point Number Translations	9	B. Trunk Selection Translations	22
C. CDPR Selection Translations	11	7. TRUNK ORIGINATIONS	22
D. Line Subtranslator	12	A. General	22
E. Line Subtranslator Expansions	13	B. Trunk Origination Scan Point Translations	22
F. PBX/MLHG Expansions	14	C. Trunk Origination Digit Translations	23
G. Two-Party Expansions	14	8. DISCONNECT	23
H. Billing/Directory Number Expansions	14	A. General	23
4. 3-DIGIT TRANSLATIONS	15	B. Line-to-Line Translations (Intraoffice Calls)	23
A. General	15		

NOTICE

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

SECTION 233-151-150

CONTENTS	PAGE
C. Interoffice Calls (Incoming or Outgoing)	24
9. SPEED CALLING	24
10. CALL FORWARDING	25
11. MESSAGE REGISTER LIST	25
12. TRIPLET INDEX INFORMATION	25
13. CHARGE INDEX EXPANSION	25
14. ROUTE INDEX EXPANSION	26
15. ALTERNATE ROUTE INDEXING	28
16. ONE-DIGIT TRANSLATION	28
17. GROUP TRANSLATION	28
A. PBX/MLHG Group Table	29
B. Service Circuit Group Table	29
C. Trunk Group Table	29
18. MEMBER LISTS	31
19. SELECTION STATUS BLOCKS	31
20. SCAN LIST	32
21. TERMINAL OFFICE TEST	32
22. PRECUT	32
23. AUTOCONNECT	32
24. GROWTH PARAMETERS	33
25. NOC TO OFFICE CODE CONVERSION	34
26. MISCELLANEOUS OFFICE PARAMETERS	34
27. LINE CLASS CODE	34
28. SCAN POINT NUMBER TRANSLATION (XSLSPN) PROGRAM	35
A. Call Forwarding Expansion (CF_EXP) Subroutine	35

CONTENTS	PAGE
B. Class A Line (CLASSA) Subroutine	35
C. Call Forwarding (CLFRD) Subroutine	35
D. Coin Triplet Index Expansion (CNI_EXP) Subroutine	36
E. Coin Line Operated and Release (COINOPR, COINRL, COINEN) Subroutines	36
F. Data Processing (DATAPR) Subroutine	37
G. Directed Scan Point Number (DIR_SPN) Subroutine	38
H. Get Directed Scan Point Number (GET_DSPN) Subroutine	38
I. Get Supervisory Scan Point Number (GET_SSPN) Subroutine	38
J. Get Telephone Number (GET_TN) Subroutine	39
K. Get Billing Number (GETBIL) Subroutine	41
L. Get Group and Member Numbers (GETGRP) Subroutine	42
M. Line Data (LNDDATA) and Terminal Equipment Number Data (TENDATA) Subroutines	42
N. Local Numbering Plan Area (LOCALNPA) Subroutine	43
O. Message Register Charge (MR_CH) Subroutine	43
P. Normalized Office Code to Binary Coded Decimal (NOC_BCD) Subroutine	43
Q. Originating Line Translation (ORGXLA) Subroutine	43
R. PBX Data (PBXDAT) Subroutine	43
S. Speed Call Expansion (SC_EXP) Subroutine	43

CONTENTS	PAGE	CONTENTS	PAGE
T. Speed Call (SPDCL) Subroutine	44	J. Precut Map (PCUTMAP) Subroutine	49
U. Scan Point Number Translation (SPNXLA) Subroutine	44	K. ROTL Security Callback Status (RSCBS) Subroutine	49
V. Service Circuit Number Conversion to Terminal Equipment Number (SV_TEN) Subroutine	44	L. Telephone Number to Unpack Telephone Number (TN_UPTN) Subroutine	49
W. Terminal Equipment Number Translation (TENXLA) Subroutine	44	M. Callback Search (CBSERCH0, CBSERCH1) Subroutine	50
X. Trunk or Service Circuit Scan Point Number Conversion to Terminal Equipment Number (TK_TEN) Subroutine	44	N. Multiline Hunt Group (MLHG_SUB) Subroutine	50
Y. Tone Present Scan Point Number Conversion to First Tone Scan Point Number (TN_SPN) Subroutine	44	O. Terminal Equipment Number to Terminating Translation (TEN_TERM) Subroutine	51
Z. Unpack Directory Number (UPK_DN) Subroutine	44	30. EQUIPMENT SELECT (EQPSEL) PROGRAM	51
29. COMMON TRANSLATION SUBROUTINES (XLSUB) PROGRAM	44	A. All Busy Check (ABCK) Subroutine	51
A. Get Entry (GNTRY, GNTRY3) Subroutines	44	B. Busy Circuit (BUSYCKT) and Busy Service Circuit (BUSYSVC) Subroutines	51
B. Expand (EXPAND_C) Subroutine	46	C. Circuit List (CKTLST) Subroutine	52
C. Frame Check (FRMCHK) Subroutine	46	D. Data Circuit (DATCKT) and Data Circuit Group (DATCKTG) Subroutines	54
D. Clear 16 Words (CLR_16A) Subroutine	47	E. Get Distributor Triplet Address (GET_DTA) Subroutine	54
E. Line Status Bits Test and Manipulation (LINESTATUS) Subroutine	47	F. Group Address (GRPADR) Subroutine	55
F. Announcement Scan Point Number (ANN_SPN) Subroutine	48	G. Group Translation (GRPX) Subroutine	55
G. Autoconnect Callback Block (AUTO_CB) Subroutine	48	H. Idle Circuit (IDLECKT, IDLESVC) Subroutine	55
H. Convert Office Code to Normalized Office Code (NXX_NOC) Subroutine	48	I. Member List (MBR_LIST) Subroutine	57
I. Precut Status (PRECUTST) Subroutine	48	J. Member Data (MBRDATA) Subroutine	57
		K. Next Service Circuit (NEXT_SVC) and Next Trunk Circuit (NEXT_TRK) Subroutine	58

SECTION 233-151-150

CONTENTS	PAGE	CONTENTS	PAGE
L. Number of Working Members (NUWKMEMS) Subroutine	58	D. Get the Alternate Route Index (GARI) Subroutine	71
M. PBX Busy (PBX_BUSY) and PBX Idle (PBX_IDLE) Subroutines	58	E. Charge Expansion (CH_EXP) Subroutine	72
N. PBX Key Status (PKEYSTAT) Subroutine	59	F. Country Code Digit (CCDIGIT) Subroutine	72
O. Port Busy (PORTBUSY) Subroutine	59	G. Country Code Expansion (CCEXP, CCEXP) Subroutine	72
P. Port Idle (PORTIDLE) Subroutine	60	H. Country Code List Translation (CCLST) Subroutine	73
Q. Port Status (PORTSTAT) Subroutine	60	I. One-Digit Table Translation (ODIGLST) Subroutine	74
R. Port Test (PORTTEST) Subroutine	60	J. 1-Digit Translation (1DIGIT) Subroutine	74
S. Select an Idle Circuit (SELECTS) Subroutine	61	K. Terminating Office Test Access Number List (TOTANLST) Subroutine	74
T. Selection Status Block Address (STATUS_A) Subroutine	62	32. 4-DIGIT TRANSLATION (XSL4DG) PROGRAM	74
U. Get Selection Status Block With Group Number (STATUS_G) Subroutine	62	A. 4-Digit Subroutine Using Series Completion Directory Number (4DSCX)	75
V. Get Selection Status Block With Scan Point Number (STATUS_S) Subroutine	62	B. 4-Digit (4DIGIT) Subroutine	75
W. Transient Call Record Selection (TCRSEL) Subroutine	63	C. Termination Translation No Precut (TXLNPC) Subroutine	76
X. Test Circuit (TESTCKT) and Test Service Circuit (TESTSVC) Subroutines	63	D. Termination Translation (TERMXL) Subroutine	76
Y. Member Select (MBRSEL) Subroutine	64	33. GLOSSARY	78
Z. Trunk and Service Circuit Select (TRKSEL) Subroutine	64	34. ABBREVIATIONS	80
31. 3-DIGIT TRANSLATION (XSL3DG) PROGRAM	67		
A. Code Index (CODE_NDX), 7-Digit (7DIGIT), and 10-Digit (10DIGIT) Subroutines	67	Figures	
B. 3-Digit (3DIGIT) Subroutine	68	1. Master Table Index Entry Formats for Generic SO-2	85
C. Route Index Expansion (RI_EXP) Subroutine	70	2. Master Table Index Entry Formats for Generic 3E3	86
		3. Translator Accessing for Generic SO-2	87

CONTENTS	PAGE	CONTENTS	PAGE
4. Translator Accessing for Generic 3E3	88	27. Termination of 4-Digit Translation	131
5. Line Origination Translations	89	28. Route Index Expansion Words	131
6. SPN, TEN, and OE Formats	91	29. Interoffice	133
7. SPN Processing	91	30. Trunk and Service Circuit Group Data for Generic SO-2	135
8. Line Translation Processing for Generic SO-2	93	31. Trunk and Service Circuit Group Data for Generic 3E3	137
9. Line Translation Processing for Generic 3E3	95	32. Trunk Origination	139
10. Line Data Processing	97	33. Trunk Origination Translations	141
11. Trunk Group Number Processing	99	34. 1-Digit Translations	143
12. Selection Status	101	35. SCRATCH Data Layout Containing the Line Data for Generic SO-2	145
13. MEMLIST Access	103	36. SCRATCH Data Layout Containing the Line Data for Generic 3E3	147
14. SPN to TEN Conversion	105	37. Base Level SCRATCH Area Upon Exit From Subroutine 4DIGIT or 4DSCX (Return Code of 5)	149
15. Determination of Trunk or Service Circuit	107	38. Base Level SCRATCH Area Upon Exit From Subroutine 4DIGIT or 4DSCX (Return Code of 6)	149
16. 3-Digit Translations	109	39. Base Level SCRATCH Area Upon Exit From Subroutine TERMXL (Return Code of 2) for Generic SO-2	151
17. Code Index Processing	111	40. Base level SCRATCH Area Upon Exit From Subroutine TERMXL (Return Code of 2) for Generic 3E3	153
18. Type 1 Code Index Expansion Processing	113	41. Base Level SCRATCH Area Upon Exit From Subroutine TERMXL (Return Code of 3) for Generic SO-2	155
19. Type 2 Code Index Expansion Processing	115	42. Base Level SCRATCH Area Upon Exit From Subroutine TERMXL (Return Code of 3) for Generic 3E3	157
20. Type 3 Code Index Expansion Processing	117		
21. Type 5 Code Index Expansion Processing	119		
22. Type 6 Code Index Expansion Processing	121		
23. Route Index Expansion	123		
24. Charge Index Expansion	125		
25. 4-Digit Processing (Intraoffice)	127		
26. 4-Digit Translation	129		
		Tables	
		A. Sequential Memory Layout of Master Table Index Entries	159

SECTION 233-151-150

CONTENTS	PAGE
B. Originating and Terminating Major Classes (SO-2, Issues 4 and 4A and 3E3)	164
C. Circuit Codes for Trunk and Service Circuit Groups	165
D. Entry Identification Codes	166

1. GENERAL

1.01 This section provides a basic description of No. 3 Electronic Switching System (ESS) office call processing translations. Translation data defines a particular physical office to the generic programs. The use and form of the translation data are described in this section through the several steps of any telephone call processed through any No. 3 ESS office. Functionally, the steps are divided into line originations, 3-digit processing, 4-digit processing, terminations, trunk originations, and disconnect. Also included in this section is a description of the programs involved in translations and a description of the subroutines within each program.

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

1.03 *This section contains information applicable to both the 3E3 generic and Issue 4A of the SO-2 generic. Where information applies to a particular generic, it will be so noted.*

1.04 Part 33 contains a glossary of terms and definitions necessary for comprehension of information contained in this document.

1.05 Part 34 contains a list of abbreviations and acronyms used in this document.

1.06 The following Bell System Practices provide background information related to No. 3 ESS translations:

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

233-151-145 Digit Processing, Software Subsystem Description, No. 3 Electronic Switching System

233-152-105 Recent Change Processing, No. 3 Electronic Switching System

254-340-102 Basic and Extended 3A Processor Instruction Set, 3A Processor Common Systems

254-340-104 Program Listing Organization and Usage, Common System Software Description, 3A Processor.

1.07 The following software program listings contain the 3A Central Control (CC) code and comments for the translation programs and subroutines:

(a) Trunk Originations Program (TKORIG), PR-3H176

(b) Line Origination Program (LNORIG), PR-3H162

(c) Digit Interpretation Progress Marks (DNTRP), PR-3H155

(d) Disconnect Progress Marks (DISCON), PR-3H154

(e) Translation Data (TDATA), PR-3H018

(f) Scan Point Number Translation Program (XSLSPN), PR-3H179

(g) Basic Translation Subroutines (XSLSUB), PR-3H180

(h) Equipment Selection Subroutines (EQPSEL), PR-3H157

(i) Three-Digit Translation Program (XSL3DG), PR-3H181

(j) Four-Digit Translation Program (XSL4DG), PR-3H182.

1.08 The following program application (PA) specifications contain the office data layouts:

- (a) Office Data Tables Layout Specification For SO-2, PA-3H302
- (b) Office Data Tables Layout Specification For 3E3, PA-3H303.

1.09 Translations make it possible for general purpose telephone control programs to function in any No. 3 ESS office. Translation routines are part of the program; therefore, they are the same in any No. 3 ESS office using the same issue. The translation routines operate to locate specific office data and provide the data to requesting programs. This information consists of directory numbers, office codes, line and trunk identifications, routing and charging information, etc, and is the translation data for a specific central office.

1.10 Translation data is furnished by local plant, traffic service, service order bureaus, and line engineering organizations and defines particular lines, trunk groups, or service circuits to the using programs.

1.11 Translation data is stored in the translation area of memory and is write-protected. Certain translations such as call forwarding, message register, coin status, and group status change frequently, so temporary store is used for this type of data.

1.12 Translation data is added, changed, or removed through use of the recent change (RC) programs. RC programs are a series of nonresident (tape storage) programs which are used to modify the translation data resident and nonresident in a particular No. 3 ESS office. For a description of the recent change programs refer to Section 233-152-105, Recent Change Processing.

2. BASIC TRANSLATION TABLES

TRANSLATORS

2.01 A translator consists of a group of tables in translation store that contain data pertinent to a specific translation process. Access to a translator is gained by locating its address in the Master Table Index (MTI). A typical translator consists of an entry in the MTI, head tables, and

subtranslators. Details of the No. 3 ESS MTI and translation tables are covered in PA-3H302 for Issue 4 of the SO-2 generic and PA-3H303 for the 3E3 generic.

A. Master Table Index

2.02 The MTI is a fixed area of translation store which contains data and provides linkage to various parameters and translation data required by the generic programs.

2.03 The MTI is an index of symbolic names that identify tables and their address locations in translation store. The symbolic names represent the functions of the translator with which they are associated. Table A shows the symbolic names and word sizes of the No. 3 ESS MTI.

2.04 The MTI is a quick-reference table used by the system to locate the address in translation store where data pertinent to a specific type of translation process is stored. Two different word formats are used and identify whether further addressing is required. Figure 1 for generic SO-2 and Fig. 2 for generic 3E3 shows both the 1-word and 3-word format. The 1-word format is used when the required data is in the MTI and no additional addressing is required. The 3-word format furnishes details required for additional addressing and indexing. Details of the MTI word formats and functions are covered in PA-3H302 for Issue 4A of the SO-2 generic and PA-3H303 for the 3E3 generic.

B. Head Table

2.05 A head table contains information about a subtranslator and an increment. A translator may or may not have a head table.

C. Subtranslators

2.06 A subtranslator contains either detailed information required by the call processing programs for specific translations or index information which points into another translator or an expansion which holds data. A subtranslator is found by using addresses from one of the tables of the MTI.

D. Accessing Translators

2.07 Translators are accessed by using one of the symbolic names (Table A) from the MTI which points to a location of translation store. This location will contain either a 1-word or 3-word entry. The 3-word entry is more common and use of this entry to find other entries and/or tables is illustrated in Fig. 3 for generic SO-2 and Fig. 4 for generic 3E3. The first word contains information used for indexing and identification of the type of translator to be expected. The second word contains the number of entries for the table. The low four bits of the second word and all 16 bits of the third word form a 20-bit store address for a storage location that contains the beginning of the translation tables. Each table can have multiple words per entry; thus, an index must be added to the store address in order to locate the proper entry. The index may be formed in different ways determined by external parameters, entry size indicator (ESI) of the MTI, number of words per entry, etc. For example, the index needed to locate the proper head table for a scan point is formed by adding the value of the high seven bits of the scan point number (SPN) to the store address. The head table entry may contain data, additional addresses, or index information that will be used to form new addresses giving the location of additional subtranslators, expansion tables, etc. Once the beginning of a subtranslator table is located, another index may be required to access the proper entry. Again, using the scan point as an example, the subtranslator required is a 2-word entry; therefore, an index is formed by multiplying the low six bits of the SPN by 2 and adding the result to the store address which points to the subtranslator table. The resulting address gives the location of the entry pertaining to a particular SPN. Addresses and indexes of other MTI entries are formed in a similar way.

2.08 If translation data requires more storage than is available in translators, expansion tables are used to provide the additional area. Expansion tables are accessed by information in a subtranslator (eg, table number and entry number). A fixed address in the MTI gives the address of the expansion translator. In the 3E3 generic expansion translators speed call and call forward have their own expansion translators. This address points to the beginning of the head table entries which are indexed by the table number from the subtranslator. This provides an increment to the

subtranslator which is indexed by the entry number. These lead to an expansion entry which may contain another table number and entry number.

E. Expansion Tables

2.09 Expansion tables are used when more data is required than can be stored in available subtranslator tables. General purpose expansion data can only be accessed through the MTI entry EXPTBL. Generic 3E3 also has a code list expansion table accessed through the MTI entry CODLST. Both of these translators use table number and entry number fields from a subtranslator entry or other expansion entries.

2.10 A 3-word program store entry, EXPTBL, includes the translation store address of the expansion head table. Expansion head table entries are 2-word entries. The first word contains the number of entries minus one and an ESI. The second word contains a store increment value to point to a specific expansion table.

2.11 The table number is used to index into the head table to select the appropriate 2-word entry and subsequent expansion table. The entry number is then used as an expansion table offset to select the desired 1-, 2-, or 4-word entry (generic SO-2) or 1-, 2-, 4-, or 8-word entry (generic 3E3) within that expansion table.

2.12 The expansion head table contains a maximum of 32 two-word entries. The expansion tables are sized as follows:

One-word expansion table—128 words

Two-word expansion table—256 words

Four-word expansion table—512 words

Eight-word expansion table (3E3 generic only)—1024 words.

The precise number of entries in these tables depends on the excess data from the line subtranslator, 4-digit translation, and/or route index expansion.

2.13 In the 3E3 generic the MTI entry CODLST is expansion tables capable of consisting of 2- and 4-word entries. With the 3E3 generic, only the 2-word entries are used. The MTI entry CODLST follows the same format as EXPTBL.

These expansion tables are used as a digit storage area for speed call and call forwarding when the number of digits are greater than 8.

2.14 Translation information defines the physical details of an office to the originating and call processing programs. In order to describe various types of calls in terms of the translation routines and data required, the origination and processing of calls are divided into steps as discussed in the following paragraphs.

3. LINE ORIGINATIONS

A. General

3.01 Since both intraoffice and interoffice calls are initiated by line originations, translation information is required to define the line originating the call and the way it should be processed through the office. These initial translations are necessary to translate a scanner identification into line characteristics. Translation programs and subroutines use the SPN to locate data in the translation tables and to retrieve it for use by the call processing programs. Figure 5 is a basic block diagram of the line origination process in terms of translations.

3.02 As shown in Fig. 5, a terminal equipment number (TEN) is equal to a SPN for networks 1 through 15 and is loaded when a customer off-hook indication is detected. The SPN is passed to translation program XSLSPN. The subroutine ORGXLA contained in XSLSPN accesses the proper MTI entries, retrieves the data from memory, and stores the results in the transient call record (TCR). This translation data identifies the originating line and consists of class-of-service information, any assigned special options, and the type of customer dial pulse receiver (CDPR) required by the line. The CDPR information is passed to the subroutine TRKSEL of EQPSEL. This subroutine accesses the proper translation tables via the MTI entries and returns translation data which call processing puts into the TCR. The translation data is the location of a CDPR and is given as a terminal equipment number, a service circuit number, and a distributor triplet address (DTA). The call processing programs use this data to locate and connect the proper CDPR to the line and to turn on the dial tone. The system is then ready to accept customer-dialed digits.

B. Scan Point Number Translations

3.03 Translation routines are entered from the originating programs via the SPN. The SPN is a 13-bit word with a format as shown in Fig. 6. The division of the high seven bits and low six bits is used for indexing the translation tables pertaining to scan points.

3.04 Scan point number translation provides conversion from a scan point number to a terminal equipment number and/or group and member numbers for nonlinear terminals (trunks, PBX, etc). Scan point number translation also provides originating class of service, equipment features, and telephone or directory/billing number for line terminals. The class-of-service information includes such things as the major class, the screening class used in routing and charging, the type of dialing, and service priority.

3.05 The translation store 3-word MTI entry for scan point table is SPTBL. The third word of the entry contains the translation store address of the SPN head table. The SPN head table contains a maximum of 128 words (ie, eight words/logical scanner). The number of entries that actually exist in the SPN head table is determined by the number of scan points in the office.

3.06 The SPN is a 13-bit number. The first (leftmost) seven bits of the SPN are used as an index into the SPN head table to select one of the possible 128 entries. These entries consist of a subtype and a store increment value. The store increment value is used to increment to the desired subtranslator (ie, the subtranslator address equals translation store address plus high seven bits of SPN plus store increment value). The low six bits of the SPN are used to select specific entries within the subtranslator.

3.07 The 3-word translation store MTI entry for service circuit table is SVCTBL. The third word in the entry points to the master scanner subtranslator (SPN equals zero).

3.08 The 8-bit service circuit number is used as an index into the subtranslator. SVCTBL provides a faster means of translation than SPTBL since no head table translation is involved. SVCTBL is used in situations where time is critical as in the case of an interrupt.

3.09 For those scan points that do not have corresponding service circuit numbers, SPTBL is required. Service circuit numbers equal scan point numbers through the first eight bits. However, there are more scan point numbers than service circuit numbers.

3.10 The following types of subtranslators are associated with scan point number translations.

(a) **Unassigned Subtranslator:** This type of subtranslator is not used during translation. It provides an area for table enlargement for more SPNs prior to those SPNs being ready for use. Also, SPNs in scanner 1 through 15, rows 24 through 27 are not translated (reserved for junctors). These are always unassigned entries in the head table.

(b) **Miscellaneous Subtranslator:** This subtranslator consists of a 64-word table of 1-word entries and is used for key SPNs (generic SO-2) test circuits, alarm circuits, announcement circuits, ringing and tone plant circuits, power cross circuits, and other miscellaneous circuits.

(c) **Universal Subtranslator:** This subtranslator consists of a 128-word table of 2-word entries. It is used for trunks, key SPNs (generic 3E3), tone present SPNs, and directed SPNs.

(d) **Line Subtranslator:** This type of subtranslator is a table of 128 words consisting of 2-word entries. The subtranslator is used for lines, tone and announcement circuits, and nonlines.

3.11 The sequence of subroutines which process the SPN is shown in Fig. 7. The SPN is used as the entry from LNORIG to XSLSPN. Within XSLSPN, subroutine ORGXLA handles the initial processing. For a normal line origination, the TEN and SPN are the same, so ORGXLA saves the SPN as a TEN, clears the translation data bits, and calls LNDATA using TEN/SPN as an entry.

3.12 Subroutine LNDATA saves the tip party indication and then calls TENXLA, which immediately calls SPNXLA. Subroutine SPNXLA separates the high seven bits and the low six bits of the TEN/SPN and stores them in different

positions. It then loads the address of the SPTBL MTI entry and the table index (high seven bits) and calls subroutine GNTRY3.

3.13 The GNTRY/GNTRY3 subroutine is part of the XLSUB program and is the subroutine that reads the translation store. This is a general subroutine and is used to access all MTI entries. Subroutine GNTRY differs from GNTRY3 only by entry point. For GNTRY, it is necessary to multiply the table index by three before obtaining the MTI entry. This covers the cases where tables contain more than one 3-word MTI entry. The processing performed by GNTRY and GNTRY3 is identical once the different entry point is passed.

3.14 Subroutine GNTRY3 (Fig. 8) accesses the MTI entry and verifies that a translator exists, the type it is, and that the range is correct. An index is formed based on the ESI. This index is then added to the store address contained in words 2 and 3 of the MTI entry. This address and the SPTBL head table word [Fig. 8(A) for generic SO-2 and Fig. 9(A) for generic 3E3] are returned to SPNXLA.

3.15 Subroutine SPNXLA (Fig. 8 for generic SO-2 and Fig. 9 for generic 3E3) checks the SPTBL head table word for subtype, etc. It then adds the store increment to the address of the SPTBL head table which is composed of the store address from the MTI entry plus the high seven bits of the SPN. The resultant address gives the location of a line subtranslator. It then multiplies the low 6-bit port of the SPN by 2 and adds it to the subtranslator address to locate the 2-word entry. The formats of the entry are shown in Fig. 8 for generic SO-2 and Fig. 9 for generic 3E3 for a normal line or for expansion words if required. The number of expansion words depends on the amount of special features assigned to a particular line. The first word and its address are then returned to TENXLA. Subroutine TENXLA checks for line subtranslator and passes the same word and address of only line subtranslator entries to LNDATA.

3.16 Subroutine LNDATA identifies the scratch area and clears it. The scratch area is a 16-word location in temporary store which is used to hold translation data during processing steps before the data is loaded in the TCR. Subroutine LNDATA checks certain items in the subtranslator such as the billing/directory word, type of line (eg,

PBX and multiparty), whether a line is assigned, and whether any expansion data is required. It then stores all these items in the proper locations of the scratch area. Subroutine ORGXLA is notified, via a return code, that the proper data is in the scratch area and ready for processing.

3.17 Subroutine ORGXLA processing (Fig. 10) checks for line features and either loads data into the TCR or requests additional translation data. Subroutine ORGXLA performs an item-by-item check on the type of line (ie, PBX, manual, coin, etc) and line information. It also checks for special features such as call forwarding, speed calling, sleeve lead, noisy line circuit, hot line, motel, etc. If any of the options are required, ORGXLA branches to other subroutines to secure the necessary translation data. Subroutine ORGXLA then loads the office data word [Fig. 10(A)] and, using this word and TCR indicators, determines the type of CDPR required. This information is returned to the call processing program (LNORIG) as a return code. The code for dial is equal to two; for TOUCH-TONE®, the code equals one.

C. CDPR Selection Translations

3.18 Program LNORIG (Fig. 11) forms a trunk group number using the return code and office data. The trunk group number (TGN) is an 8-bit number which is used to form the index required to locate trunk or service circuit group data.

3.19 Program LNORIG calls GETCKT and furnishes the TGN. Subroutine GETCKT calls TRKSEL which in turn calls GRPX. The TGN is passed as entry data in each case. Subroutine GRPX loads MTI entry GRPTBL and forms the table index using the low six bits of the TGN. Subroutine GNTRY is called, and the table index and MTI entry address are furnished to it. Subroutine GNTRY receives the MTI entry, checks the words and indexes using the table index, and returns the first word and address of a service circuit (CDPR) group data translator [Fig. 11(B)] to GRPX. Subroutine GRPX checks the first word, loads a return code indicating the type of circuit, and returns the first word and address of the group data to TRKSEL.

3.20 Subroutine TRKSEL (Fig. 12) checks the group data to ensure that the group exists and has a working member and saves the largest

member number. Subroutine TRKSEL then calls STATUS_A and furnishes the group data address.

3.21 Subroutine STATUS_A (Fig. 12) determines whether a valid selection status block exists. If one exists, it calls GNTRY and furnishes the MTI entry address, GRPSTAT, and the trunk group number. Subroutine GNTRY accesses GRPSTAT and returns the first word and address of the selection status block for service circuits [Fig. 12(B)]. Subroutine STATUS_A returns the first word and address of the selection status block to TRKSEL.

3.22 Subroutine TRKSEL (Fig. 13) uses the selection status block information by calling SELECTS subroutine to locate an idle circuit within the group and to generate a member number for the circuit. Subroutine TRKSEL then calls CKTLST (Fig. 13), furnishing the member number and group data address. Subroutine CKTLST retains the member number and list size and calls MBR_LST, furnishing the group data address. Subroutine MBR_LST forms an index from group data information. It then loads the MTI entry address, MEMLST, and calls GNTRY. Subroutine GNTRY accesses the MTI word and returns the first word and address of the service circuit member list to MBR_LST [Fig. 13(A)], which returns the same information to CKTLST.

3.23 Subroutine CKTLST (Fig. 14) tests the member list word [Fig. 13(A)] for the type of circuit. After forming indexes, it locates the appropriate SVCNBR (service circuit SPN) and DTA. Subroutine TK_TEN is called to translate the given SPN into a TEN. The SPN is translated into a TEN through the use of subroutines SPNXLA and GNTRY3, and the return is a universal subtranslator as depicted in Fig. 14(B). Subroutine CKTLST stores the TEN from the subtranslator and the DTA from the member list into the scratch area and returns a success return code and the SPN to TRKSEL.

3.24 Subroutine TRKSEL checks the return code and sets the ignore bits for the selected circuit (Fig. 15). If the selected circuit is a service circuit, TRKSEL also increments the group peg count on the first and last try. Subroutine TRKSEL then returns a success return code to LNORIG through GETCKT.

3.25 Program LNORIG now has the data necessary to identify a CDPR. The TEN and DTA of the CDPR are moved from the scratch area to the TCR. Program LNORIG calls the path hunting routines to select a path between the calling line and the CDPR. When this is completed, LNORIG sets the DIALTON base level progress mark and turns control over to INPUT. The dial tone is turned on and the system is ready to receive dialed digits.

D. Line Subtranslator

3.26 All entries in the line subtranslator are 2-word entries. A description of each possible entry type is as follows:

- (a) **Unassigned:** This entry consists of all zeros in both words.
- (b) **Normal Line:** This entry supplies the line originating major class, screening class, and billing and directory number. The major class (see Table B) includes any special originating and terminating actions required for a line or a directory number. The screening class is used to indicate the type of treatment given a line for various types of calls, including routing and charging information.

The normal line entry also indicates if the line is an essential line or a ground start line or if it has TOUCH-TONE service.

The second word of this entry either contains the billing and directory number or has a table number and entry number that point to expanded billing information. It also contains QZ billing, service observing and special studies information. The expanded billing information may be a 1- or 2-word entry. If it is one word, then it contains the directory number, which is the billing number also. If it is two words, then the first word contains the directory number and the second word contains the billing number.

- (c) **Normal PBX/MLHG:** This entry provides the PBX/MLHG group number and member number for the line and the directory number. If expanded billing/directory information is required, a table number and entry number are provided instead of the directory number in the second word. The table and entry numbers are used to access the expansion tables. Two types

of expanded billing/directory number entries are provided for normal PBX/MLHG lines. The first is a 1-word entry containing a number used as both a billing and directory number or as a directory number only and the billing number is used from the group data. This 1-word expansion is also necessary when a PBX/MLHG line has associated with it QZ billing, service observing, and special studies information. The second type is a 2-word entry used when the billing number is different from the directory number; also, when the billing number is different from the billing number in the group data. One word provides the billing number, and the other word provides the directory number. The second type may also contain QZ billing, service observing, and special studies information.

- (d) **Tone and Announcement Circuit:** This entry provides a group and a member number for tone and announcement circuits.
- (e) **Three-Port Circuit:** This is a 2-word entry that contains the port number, member number, and group number.
- (f) **Nonline:** This entry provides either the circuit supervisory scan point number or its directed scan point number. The entry also includes a supervisory scan point flag bit to indicate which type of scan point number is given (supervisory scan point or directed scan point).
- (g) **Normal Line With Expansion:** This entry is used to access expansion tables. A table number and entry number are provided for accessing line feature data. Either a directory number is provided or a second table number and entry number are provided for billing and directory number expansion. Also included in this entry are indications as to whether QZ billing, service observing, or special studies are provided for the line.
- (h) **Two-Party Line:** This entry includes table and entry numbers used for accessing 2-party expansion tables. The entry indicates if the line has either TOUCH-TONE service for ring party or TOUCH-TONE service for tip party. A 2-bit exit code is provided to indicate the type of customer dial pulse receiver to connect to the 2-party line. The following exit codes exist.

- 00—Error.
- 01—Attach a TT receiver.
- 10—Attach a DP receiver.
- 11—Do not attach any receiver. Both parties are denied, or one is unassigned and the other denied.
- (i) **Four-Party Without Expansion:** This entry supplies the line originating major class and screening class. It also indicates if the line is an essential line or ground start line or if it has TOUCH-TONE service. The second word contains a table and entry number which points to a 4-word expansion containing directory numbers. Service observing and special studies are also provided for the line.
- (j) **Eight-Party Without Expansion:** This entry supplies the line originating major class and screening class. It also indicates if the line is an essential line or ground start line or if it has TOUCH-TONE service. The second word contains a table and an entry number which points to a 2-word expansion containing table and entry numbers. Each table and entry number points to a 4-word expansion containing directory numbers. Service observing and special studies are also provided for the line.
- (k) **Four-Party With Expansion:** This entry contains table and entry numbers used for accessing 2- or 4-word expansion tables. A table and entry number in the first word is provided for accessing line feature data. The second word contains a table and entry number which points to a 4-word expansion containing directory numbers. Service observing and special studies are also provided for the line.
- (l) **Eight-Party With Expansion:** This entry contains table and entry numbers used for accessing 2- or 4-word expansion tables. A table and entry number in the first word is provided for accessing line feature data. The second word contains a table and entry number which points to a 2-word expansion containing table and entry numbers. Each table and entry number points to a 4-word expansion containing directory numbers. Service observing and special studies are also provided for the line.
- (m) **PBX/MLHG Member With Different Data from Group Data:** This entry contains table and entry numbers used for accessing PBX/MLHG expansion tables and either a directory number or table and entry numbers for billing/directory number expansion and service observing, special studies, and QZ billing information.
- (n) **Line Expansion Autoconnect:** This uses a 2-word expansion entry from the first word. This entry contains a table number and entry number in the first word and a table number and entry number for the trigger numbers. The first word contains a major (MAJ) of 29 and a screening class number (SCR) of 31, and essential line, ground start, and TOUCH-TONE bits; the second word contains a sleeve lead, DTA.
- (o) **Normal Line Autoconnect (3E3 generic only):** This entry contains the MAJ of 29, an SCR of 31 and essential line, ground start, and TOUCH-TONE bits, and the second word contains the table and entry numbers for the trigger numbers. There is no sleeve lead in this type of autoconnect, so no expansion entry from the first word is needed.

E. Line Subtranslator Expansions

3.27 The line expansion entries consist of a word supplying originating major class, screening class, and information as to whether the line has 3-way service, TOUCH-TONE service, or an essential line or ground start line. In addition to this word, the entry may contain from zero to three line data words for Issue 4 of the SO-2 generic and from zero to seven line data words for the 3E3 generic.

3.28 If several expansion entries are provided, they are linked by means of table and entry numbers. A table and entry number word, when provided, must be the last word of the entry.

3.29 There are 12 possible formats in Issue 4 of the SO-2 generic and 13 possible formats in the 3E3 generic of line data words as follows.

- Type 0000 is a vacant word consisting of all zeros.
- Type 0001 contains table and entry numbers for linking several expansion entries.

- Type 0010 contains a call forwarding index used to access call forwarding expansion entries in temporary store and the availability list in translator store in Issue 4 of the SO-2 generic, and to access the digit storage list in translator store in the 3E3 generic.
- Type 0011 contains a message index used for indexing into the software message register list table in temporary store.
- Type 0100 contains a distributor triplet address and uses a hardware message register for unit counting.
- Type 0101 has information as to whether the line has speed call 8 service and whether it is permitted to change the speed call 8-code list. In generic 3E3 there is not a special indicator that the line has speed call, but the assignment of the entry indicates assignment. If the line has speed call 8 service, a speed call 8 index is provided for accessing entries in the speed call 8 head table.
- Type 0110 is the same as type 0101 above except that it provides speed call 30 information.
- Type 0111 provides terminating translation data in the form of a sleeve lead distributor triplet address.
- Type 1000 provides terminating translation data in the form of a noise immunity line circuit distributor triplet address.
- Type 1001 provides a triplet index and triplet point used in coin line circuit triplet translation.
- Type 1010 provides a table and entry number for accessing an expansion table of telephone numbers stored in BCD for hot line service. Digits 1 through 8 require a 2-word expansion, and digits 9 through 11 require a 4-word expansion.
- Type 1011 contains an originating major class and a screening class for a PBX/MLHG member with different data from the group data. This entry word also indicates whether TOUCH-TONE service is applicable.

- Type 1100 (3E3 generic only) contains 2 bits. The low bit (AL bit) indicates whether or not automatic line insulation testing is inhibited for the line. The second bit (ROH bit) indicates whether or not the receiver off-hook tone is inhibited on this line.

F. PBX/MLHG Expansions

3.30 PBX/MLHG expansion entries consist of one word containing PBX/MLHG member and PBX/MLHG numbers. This word also supplies information as to whether the line is an essential or ground start line and if it has 3-way service. In addition to the word described above, the entry may also have from zero to three line data words for Issue 4 of the SO-2 generic and from zero to seven line data words for the 3E3 generic (3.27).

G. Two-Party Expansions

3.31 Two types of entry formats exist for either tip or ring party. Either party may also be unassigned. One entry format describes whether the line is an essential line or a ground start line. It also tells whether the line has TOUCH-TONE service and provides an originating major class and a screening class. The second entry format has table and entry numbers for further expansion. Both formats also contain either a billing and directory number or table and entry numbers used for obtaining expanded billing and directory information.

H. Billing/Directory Number Expansions

3.32 Billing and directory number expansion is used for lines which require more information than can be contained in a normal line subtranslator entry.

3.33 Four possible entry formats exist for billing/directory number expansions. A 1-word entry may be used which contains a number defined as both a billing and directory number or as a directory number only. In the event that a line has a directory number different from its billing number, a 2-word entry is available. One word of this entry contains the originating line directory number, and the other contains its billing number. Four-word entries exist for use in 4- and 8-party service since each party has its own directory number. In the case of an 8-party line, a 2-word entry containing table and entry numbers

in each word is used to access two of the 4-word entries containing the directory numbers for each party. For autoconnect there are two 4-word entries. The first entry contains the billing number, two directory numbers, and a table and entry word which points to the second entry. The second entry contains four trigger numbers.

3.34 The billing/directory number entry used in the line subtranslator and expansions is in coded form. The coded number includes a 5-bit office index and the last three digits of the billing/directory number in binary representation. This coded form is used so that a complete billing/directory number can be represented in one 16-bit word. The 5-bit office index replaces the normalized office code (NOC) and the thousands digit in the billing/directory number.

3.35 This office index is used to index the 16-word MTI entry OIEXP (office index expansion) to obtain its associated NOC and thousands digit. OIEXP is constrained to contain a maximum of 32 entries (two per word). The high four bits of the office index are used to select a particular data word in OIEXP. The low bit of the office index is then used to select which half of the data word contains the associated NOC and thousands digit. An office index of zero is not used and an office index of 31 indicates special billing (presently used by WATS).

4. 3-DIGIT TRANSLATIONS

A. General

4.01 The 3-digit translation is used to check the validity of an office code, to determine whether a call is to be permitted, to provide routing information for completing the call, and to provide charging information applicable to the call.

4.02 Three-digit translation data is accessed from translation store via the MTI entry FATTBL (foreign area translator table). FATTBL is indexed by a 2-bit code called FAT. FAT points to one of four translators, one local area translator (FAT equals zero) and three foreign area translators (FAT equals one through three).

4.03 A local No. 3 ESS office can use the following types of local and foreign area translators:

- (a) Index table

- (b) Search table.

The index table requires 150 or more words than the search table.

4.04 The three digits dialed by the customer (200 through 999) are converted into a 10-bit number series. If the office uses an index table, the high nine bits are used to select one of the words (400 words/table) in the local or foreign area translator table. The low bit selects which half of the word contains the associated "code index."

4.05 If the office uses a search table, the maximum size of the table is 256 words (128 two-word entries). A match between the 10-bit number (minus 200) associated with the 3-digit number dialed and the 10-bit number in the table entry yields the appropriate code index.

4.06 Since a number of 3-digit office codes may use identical routing and charging data, an abbreviated number called a code index is assigned to replace the three digits dialed. Therefore, a number of 3-digit office codes may have the same code index if they have similar associated routing and charging data. The code index in the last entry of a search table is used as a catchall in the event that a 3-digit (ten binary bits) match cannot be found. This code index can be set to any desired value as default, depending on desired routing.

4.07 After line origination is completed and a CDPR is connected to the calling line, the system is ready for digit processing. The digit interpretation programs interpret the customer-dialed digits to define the call as either terminating or outgoing. This is accomplished by evaluating the first three or six digits dialed (excluding prefixes). The digit processing subroutine DNTRP performs most of the interpretation required and provides the main interface between the digit processing and translation routines. Details of digit processing software are covered in Section 233-151-145, Digit Processing, Software Subsystem Description.

4.08 Subroutine DNTRP interprets the dialed digits in the TCR and determines how they are to be translated by evaluating various prefixes and status bits. It then furnishes the digits to be translated to translation program XSL3DG. Program XSL3DG (Fig. 16) uses the subroutine 3DIGIT and MTI entries FATTBL, CDIEXP, and SCTBL to

SECTION 233-151-150

translate the dialed digits into a charge index and a route index. The charge index is used by subroutine CH_EXP to develop charge information. The route index is used by subroutine RI_EXP to determine routing and to identify whether the call is intraoffice or interoffice. A more detailed description of 3-digit translation processing is given in the following paragraphs.

B. Foreign Area Table Translation Processing

4.09 Subroutine DNTRP calls 3DIGIT (Fig. 17) when three digits are in the TCR. If there are more than three digits, indicators in the TCR denote the type of prefix. At this time, the TCR also contains data pertaining to the calling line such as a screening class, type of service to be given, etc. Subroutine 3DIGIT receives the three digits from the TCR and has them converted into a 10-bit binary number. It then gets the MTI address, FATTBL, and the foreign area translating (FAT) table entry. On the first 3-digit translation, FAT should equal 0 denoting a local area translator. Subroutine 3DIGIT then forms an index using the FAT number and the 10-bit binary number formed from the dialed digits. Subroutine GNTRY3 is called and the first word and address of the FATTBL entry specified by the index [Fig. 17(A)] are returned. This entry is one word, containing two code indexes, and is returned to 3DIGIT. If the office uses the search table type of FAT, a match is found between the 10-bit number associated with the dialed 3-digit number and the 10-bit number in the table; the appropriate code index is then returned as above. If there is no match, the last code index is used.

4.10 Subroutine 3DIGIT gets the MTI address (CDIEXP) and calls GNTRY3, furnishing the code index to use as a table index and CDIEXP. Subroutine GNTRY3 returns the address and the first word of a 2-word code index expansion entry [Fig. 17(B)]. The code index expansion can be any one of the following types:

- Type 0—Vacant
- Type 1—Ignore all prefixes

- Type 2—Prefix of 1 normally received
- Type 3—No prefix normally received
- Type 4—Ignore a prefix of 1
- Type 5—FAT required on next three digits
- Type 6—Conflict between area code and office code
- Type 7—Preroute peg count desired.

Subroutine 3DIGIT determines the type and then branches according to type. In terms of telephone calls, type 1 represents the general type which can be used on any type of call. Type 2 generally represents a 7-digit interoffice call. Type 3 is used for intraoffice calls, and type 5 covers the 10-digit call (different area code).

C. Code Index Expansion Processing

4.11 The code index entries from the local or foreign area translators are used in association with the MTI entry CDIEXP (code index expansion) which provides the translation store address for the code index expansion table. The code index is used to select the desired 2-word entry within this 512-word (maximum) translation table. The first entry (code index equals zero) in the table must handle vacant code treatment. The second entry (code index equals one) in the table is for "dial 0" operator treatment.

4.12 There are eight types of code index expansion table entries. Bits 0 through 3 of the first word of each entry are used to designate the type as described below.

- (a) **Type = 0:** This type is used to designate a vacant entry.
- (b) **Type = 1:** This type of entry ignores any prefixing. It contains a direct route index

to the office represented by the office code. It also contains a screening table number. The direct route index is used if the screening process does not result in a route index.

(c) **Type = 2:** This type of entry contains, in addition to a screening table number and direct route index, two code indexes that allow or deny a "0" or "no" prefix. A "1" prefix is normally received. When a "1" prefix is not received, the appropriate code index is used to select another entry (containing no code indexes) within the code index expansion table. Entries pointed to by a "0" prefix code index must be defined in the first 128 entries of the table for Issue 4 of the SO-2 generic and in the first 64 entries of the table for the 3E3 generic. The "0" code indexes need not be denied if "0+" dialing is not allowed in the office.

(d) **Type = 3:** Normally, no prefix is received for this type of entry. Code indexes are provided to allow or deny a "0" prefix or a "1" prefix. This type of entry also has the screening table number and direct route index. Entries pointed to by a "0" prefix code index must be defined in the first 128 entries of the table for Issue 4 of the SO-2 generic and in the first 64 entries of the table for the 3E3 generic.

(e) **Type = 4:** This type of entry ignores a "1" prefix but provides a code index to allow or deny a "0" prefix. The usual screening table number and direct route index parameters are also provided. Entries pointed to by a "0" prefix code index must be defined in the first 128 entries of the table for Issue 4 of the SO-2 generic and in the first 64 entries of the table for the 3E3 generic.

(f) **Type = 5:** This entry is used when a call is being made to a foreign numbering plan area. A 2-digit foreign area translator (FAT) code is provided to index FATTBL in the MTI. The FAT code selects an entry word pointing to a foreign area translator. The foreign area translator contains a code index for each 3-digit combination in the foreign area.

(g) **Type = 6:** This expansion type is required when the same three digits are used for both office and area codes. Both an area code index and an office code index are provided.

The index that is used depends upon whether seven or ten digits are dialed.

(h) **Type = 7:** This type is used when there is a need to determine the amount of traffic for a specific office code. A peg count number entry is provided as well as a code index back into the code index expansion table for proper routing.

4.13 Processing a type 1 expansion (ignore all prefixes) is basic to several other types which are returned at times to be processed as a type 1. (Refer to Fig. 18.) The screening class number (SCR) is checked. If screening is not required, the route index (RI) contained in the code index expansion is loaded into the TCR and subroutine RI_EXP is called to expand the RI.

4.14 If screening is required, the SCRTBL MTI entry is furnished to GENTRY3 which returns the first word and address of the screening table. This table is then indexed using the SCRTBL field from a code index expansion entry and an SCR from the TCR to obtain a screening table entry of the type shown in Fig. 18(B), which contains a toll diversion bit (TDV), a charge index, and possibly a route index.

4.15 The RI is checked to see whether it does exist (nonzero). If it exists, it is stored in the TCR; if not, the direct route index (DRI), which is the route index (RI) from the code index expansion entry, is stored. A check is then made in the TCR to see if the call is to be forwarded; if so, the forwarded number route index is obtained. The charge index, TDV, and proper RI are stored in the TCR, and RI_EXP is called to perform route index expansion.

4.16 The type 2 code index expansion (prefix of 1 normally received) contains a screening table number and a direct route index which are used when a prefix of 1 is received. [Refer to Fig. 19(A).] This indicates a 7-digit interoffice call within the same area code. This entry can also allow or deny 0 prefixing or no prefixing.

4.17 If a prefix of 1 is received as expected, the type 2 entry returns to type 1 for processing and the entry is processed as previously described for a type 1, with the data from the type 2 word entered in the TCR.

SECTION 233-151-150

4.18 If no 1 prefix is received, a check is made for a 0 prefix; if 1 is received, the 0 prefix code index is obtained from the expansion word and a return is made to code index processing (paragraph 4.09 and Fig. 17). Processing then proceeds to obtain another code index expansion, which is again processed according to its type. The presence of 0 code index in an entry depends on whether a particular office allows 0 prefixing.

4.19 If neither a 1 nor a 0 prefix is received, then the no prefix code index is obtained from the type 2 expansion word and is returned to the code index processing routines in exactly the same manner as described above.

4.20 The type 3 (normally no prefix is received) code index expansion contains a screening table number and a direct route index [refer to Fig. 20(A)]. This entry is used for intraoffice calls. Code indexes may be provided to allow or deny a 0 or 1 prefix.

4.21 If there is a 1 prefix, the 1 prefix code index is obtained and a return is made to code index processing (paragraph 4.09 and Fig. 17) for a new expansion entry which is then processed in accordance with the new type. If there is a 0 prefix rather than a 1 prefix, the 0 prefix code index is obtained and another entry is located as described above.

4.22 If there is no prefix, a return is made to type 1 processing (paragraph 4.13 and Fig. 18) and the data contained in the entry [Fig. 20(A)] is processed as a type 1.

4.23 The type 5 (foreign area translation required on next three digits) entry contains a FAT number ranging from 1 through 3 [Fig. 21(A)]. This entry implies that the three digits dialed are for a different area code and that a new code index from a foreign area translator is needed to route this type of call. The FAT number is therefore stored in the TCR, and a return code of 8 is returned, implying to the calling program to call 3DIGIT (Fig. 17) to return through FATTBL translations using the new FAT in the TCR.

4.24 The type 6 (conflict between area code and office code) entry contains an area code index and an office code index (Fig. 22). This entry implies that either 7 or 10 digits may be dialed. Presently there are only 3 digits dialed

(area code index or office code index). The code index number entered is stored in the TCR and a return code of 9 is returned. This implies to the calling program to call subroutine 7DIGIT or 10DIGIT subroutines depending upon whether 7 or 10 digits were dialed. The 7DIGIT and 10DIGIT subroutines are also entry points into the 3DIGIT subroutine.

4.25 The type 7 (preroute peg count desired) entry contains a peg counter number and a code index (Fig. 17). The peg counter is incremented and the new code index obtained from the entry is used to index the CDIEXP table for new entry and type.

D. Screening

4.26 A 3-word translation store MTI entry for screening table is SCTBLS. The third word of the entry contains the store address of the screening tables. A 5-bit screening table number (SCRTBL) from code index expansion is used, along with the entry size indicator in the first word of the entry, to select a particular screening table.

4.27 There is a maximum of 32 (for the SO-2 generic) or 64 (for the 3E3 generic) screening tables. All the screening tables are the same size with a maximum of 31 (for SO-2) or 63 (for 3E3) words each. A screening class number (SCRCLASS) from the line expansion of the SPN translator is used to select a particular entry within the selected screening table. A screening class of 31 (or 63) indicates that no screening is to be performed.

4.28 The screening table entry is a 1-word entry containing:

- A toll diversion (TDV) bit
- A 5-bit charge index
- An 8-bit route index.

E. Route Index Expansion Processing

4.29 Subroutine RI_EXP is called after the screening table processing (Fig. 23). Subroutine RI_EXP is furnished a route index, either as a direct route index from a code index expansion entry or from a screening table entry if screening was required. Subroutine RI_EXP gets the MTI address RIEXP, forms an index from the route

index, and calls GENTRY3. The first word and address of a route index expansion are returned.

4.30 The expansion entries are two words and can be one of the following types [Fig. 23 (A)]:

- Type 0—Unassigned
- Type 1—Intraoffice
- Types 2 through 5—7- or 10-digit interoffice with and without overlap outpulsing
- Type 6—Destination determined by DEST code in RI
- Type 7—Interoffice (outpulse received digits).

The particular type is loaded as a return code by RI_EXP and returned to 3DIGIT which in turn returns to DNTRP.

4.31 A type 1 is returned for intraoffice calls.

It contains a route index which is the normalized office code (NOC). The NOC is the data required for entering 4-digit translation (discussed in Part 5). Route index expansion types 2 through 5 represent various types of interoffice calls and provide the trunk group number of the primary trunk to be used and information (ARP bit) as to whether an alternate route is available. Also provided is the prefixing and deleting information. Interoffice calls are discussed in Part 6. Type 6 represents routings to destinations such as vacant code operator, announcements, etc. Type 7 is used for tandem calls, N11 calls, IDDD calls (generic 3E3), or office code split calls and contains the outgoing trunk group number, in addition to the same information found in types 2 through 5. See Part 14 for a more detailed explanation of route index.

F. Charging

4.32 The CH_EXP subroutine (Fig. 24) is called to get charging data. The charge index is obtained as a result of the screening table processing of a type 1 code index (Fig. 17). The charge index is contained in the screening table entry [Fig. 17(B)]. Subroutine CH_EXP forms an index from the charge index and accesses MTI address CHARGETB. The charge table [Fig. 24(A)] is returned, and CH_EXP obtains the coin or message indicator and then returns to 3DIGIT. Any charge

indicator is loaded in the TCR by 3DIGIT. A charge index of 1 represents a free call. See Part 13 for a more detailed explanation of charge index expansion.

5. INTRAOFFICE TRANSLATIONS

A. General

5.01 Upon completion of 3-digit processing, a route index is loaded into the TCR and a return code is generated for return to the calling program (DNTRP). Based on the return code, DNTRP will set a base level progress mark. For a local call, the route index is the NOC. The TERMX progress mark is set, indicating a local call. Base level processing recognizes the TERMX progress mark and activates the program TERM. Program TERM performs some validity checking and then branches to subroutine 4DIGIT in order to translate the last four customer-dialed digits.

5.02 Subroutine 4DIGIT (Fig. 25) translates the normalized office code (NOC) and the last four digits dialed by the customer into the equipment location of the called customer line. The translation uses the NOCD MTI entries and stores a TEN and line information pertaining to the called line into the TCR. This information is then used by call processing programs to locate the correct equipment terminal and a path through the network and to reserve a talking path. The ringing circuit information is also present in the TCR. The EQPSEL program translates this data into the TEN, SPN, and DTA for the ringing circuit. This data is then used by the call processing programs to locate and connect the selected ringing circuit to the called line.

B. 4-Digit Translations

5.03 Four-digit translation is used to obtain the TEN of the called line and to provide terminating major class and possibly other specific information necessary for completion of a call.

5.04 The input parameter for the 4-digit translation is the NOC, representing the 3-digit office code dialed and the last four digits of the called line. The four digits used are binary coded decimal (BCD) equivalents of the digits dialed by the calling customer or obtained from a translation involving series completion or call forwarding.

SECTION 233-151-150

5.05 The NOC is obtained from the route index in the 3-digit translation for intraoffice calls.

5.06 For incoming calls, the NOC is obtained from the trunk group translation. If the trunk group translation cannot supply the NOC, it refers to either the 3-digit or 1-digit translator.

5.07 A 24-word (eight 3-word entries) program store MTI entry used for 4-digit translations is NOCD. The NOC is used to index NOCD to select the desired 3-word entry. This entry contains the translation store address for the number group table associated with the dialed office code.

5.08 Each NOC can have up to 10 number group tables associated with it. However, there is a maximum of 30 total number group tables because OIEXP only allows for 30 combinations of NOCs and thousands digits. These 30 combinations allow for 300 hundreds group tables. Each number group table consists of ten words. The number group table uses the thousands digit dialed by the calling customer to locate a particular entry within the table.

5.09 The number group table can have the following types of entries:

- (a) A vacant entry denoting a vacant thousands group
- (b) An entry having an 8-bit code index, indicating routing to another office
- (c) An entry containing a store increment used to access a thousands group table.

5.10 The thousands group tables are accessed by adding the store increment from the number group table entry to the address of that number group table entry. There are ten thousands group tables per number group table. Each thousands group table consists of ten words or entries.

5.11 The hundreds digit dialed by the calling customer is used to index the thousands group table to select one of the entries. The thousands group tables may have the following types of entries:

- (a) A vacant entry denoting a vacant hundreds group

(b) An entry containing a store increment value used to access one of the hundreds group tables and a bit indicating whether the hundreds group is in the precut state.

(c) A third option in the 3E3 generic is an entry containing an 8-bit route index used to route calls to another office per hundreds group.

5.12 There are 10 hundreds group tables per thousands group consisting of 200 words per table for Issue 4 of the SO-2 generic and 300 words per table for the 3E3 generic. Once the hundreds group table for a particular called line is located, the tens and units digits dialed by the calling customer are used to locate the called line directory entry in the table. The actual algorithm for the value used as an index is $2(10 \times \text{tens} + \text{units})$ for Issue 4 of the SO-2 generic and $3(10 \times \text{tens} + \text{units})$ for the 3E3 generic.

5.13 There are five basic types of entries that may be found in the hundreds group tables. In the SO-2 generic, the entries are 2-word entries. In the 3E3 generic, the entries are 3-word entries. The unassigned entry third word is blank. All other entries contain a 2-bit call trace alarm and an 8-bit line class code index. The five basic types of entries are described as follows:

(a) **Unassigned:** This entry in the hundreds group table is used for directory numbers not assigned in the office. This entry consists of all zeros. If the directory number has been unassigned and remarks were inputted at that time, the high 3 bits of the first word would be zero and the low 13 bits would contain a year index, a month number, and a remark index.

(b) **PBX/MLHG:** This entry supplies the PBX/MLHG group number, the terminating major class, and the first and last hunt members. The first hunt member specifies the first member in a group of terminal equipment numbers (TENS) listed in the PBX terminal list over which busy-idle hunting is required, which is the member number of this terminal. The last hunt member specifies the last member to be hunted in the group. If the member is denied termination, then instead of the last hunt member, there is a route index for further handling of the call. This entry also specifies whether the line has a call trace feature and whether the originating

translator supplies terminating translation data. An originating translator can provide the following termination translator data options:

- (1) Ground start
- (2) Call forwarding index (not supplied for PBX/MLHG)
- (3) Sleeve lead distributor triplet address
- (4) Noisy line circuit distributor triplet address.

(c) **Normal Line:** This entry supplies the terminating major class and terminal equipment number of the line. Call waiting service and call tracing information are supplied whether or not the originating translator has terminating translation data as indicated. This entry also supplies a route index to the route index expansion for possible further special direction of call handling.

(d) **Intercept or Special Route Line with Remarks:** This entry supplies the terminating major class, year index, month number, index for remarks, route index, and call tracing information.

(e) **Line Requiring Expansion Data:** The first word of this entry contains table and entry numbers to index the expansion tables. The second word has a format identical to the second word of either a normal line entry or a PBX/MLHG entry.

5.14 The expansion entries for lines requiring expansion data are described as follows:

(a) **Lines With Series Completion Directory Number:** This expansion entry provides a terminal equipment number and a 4-digit station number in BCD. The NOC used is the same as the entry NOC for Issue 4 of the SO-2 generic. In the 3E3 generic, the series completion directory number is a packed number and therefore can be any TN in the office.

(b) **Line Having a Key Scan Point Number:** This entry contains the terminal equipment number, the key scan point number, and tone information. The tone can be either a reorder tone or a busy tone.

(c) **Line With Both Key SPN and Series Completion Number:** This entry supplies the key SPN, the series completion directory number, tone information, and terminal equipment number.

5.15 Subroutine 4DIGIT (Fig. 26) is called by TERM and initially obtains the four dialed digits and NOC from the TCR. It then calls TERMXL with the NOC as entry data. Subroutine TERMXL prepares the scratch area and procures the MTI address (NOCD) and then calls subroutine GENTRY. Subroutine TERMXL furnishes GENTRY with the NOCD and the thousands digit. Subroutine GENTRY returns the first word and address of the number group table entry [Fig. 26(A)], which is a store increment. Subroutine TERMXL uses the store increment and the hundreds digit from the four dialed digits, forms an index, and then accesses the thousands group table which contains another store increment [Fig. 26(B)]. The tens and units digits of the dialed number are converted into a binary number, multiplied by two or three, and added to this store increment. This forms the address of the hundreds group table 2-word entry for Issue 4 of the SO-2 generic and 3-word entry for the 3E3 generic [Fig. 26(C)]. This entry is stored in the scratch area and a return is made to 4DIGIT (Fig. 27).

5.16 Subroutine 4DIGIT moves the translation data from the scratch area to the TCR and returns to TERM via the proper return code, which is formed by adding one to the hundreds group table identification (ID) number. In the case of a normal line, the entry ID is 6; adding one results in a return code of seven. Subroutine TERMXL returns this code to 4DIGIT, which in turn returns it to TERM.

5.17 The returned data indicates to TERM that the called number is valid. Program TERM checks for special features and whether the line is busy. Once these checks have been made, TERM initiates the path and ringer selection. The path is selected using the macro PATHUNT, which returns the junctor switch number of the path selected to the TCR. Program TERM next determines whether a special or regular ringing circuit is required. The ringing circuit is a service circuit and is selected by the TRKSEL subroutine of EQPSEL in an identical manner as described for the CDPR in paragraphs 3.16 through 3.22. Once the ringer is selected, the path is selected to connect

it to the called party and various checks are made by TERM. Details of these activities are covered in Section 233-151-130, Basic Call Processing. Once the necessary checks are completed, the called party is rung and the call is completed.

6. INTEROFFICE

A. General

6.01 When 3-digit translation returns a route index expansion of types 2 through 5 or 7 (Fig. 28), some type of interoffice call is indicated. Program DNTRP uses the type to determine how much to add to the SIG DIG bit in the TCR to indicate how many more digits should be expected (four for a 7-digit number and seven for a 10-digit number). If overlap outpulsing is allowed, a transfer will be made to the overlap routine, OGTOV, in OUTCAL. If overlap outpulsing is not allowed, the base level progress mark will be set for a nonoverlap routine and the TCR set to wait for the end of dialing.

6.02 The route index expansion entry (Fig. 28) contains a trunk group number. The trunk group number is furnished to TRKSEL (Fig. 29) which utilizes MTI entry GRPTBL to determine the location of a trunk within the specified group. This translation is returned as a TEN, SPN, and DTA in the TCR; the type of transmitter required for the trunk is also included. The call processing programs use the TCR information to locate and connect the trunk to the calling line. If a transmitter is needed, TRKSEL then performs another translation and locates the proper type of transmitter, returning a TEN, SPN, and DTA for a service circuit. Call processing programs connect the transmitter to the proper trunk so that dialed digits may be outpulsed.

B. Trunk Selection Translations

6.03 Subroutine TRKSEL has the function of locating the translations which define an available outgoing trunk within the given trunk group. Processing for outgoing trunk translation is identical to CDPR selection, which is described in paragraphs 3.16 through 3.22, except that the group peg count is not incremented on trunks.

6.04 The call processing programs use the returned translation data (Fig. 30 for generic SO-2 and Fig. 31 for generic 3E3) to establish the

network path and to define the features of the trunk. The trunk data also indicates the type of service circuits required (eg, multifrequency, dial pulse, etc) for outpulsing. The translation processing required for locating the proper service circuits is the same as previously described in paragraphs 3.16 through 3.22.

7. TRUNK ORIGINATIONS.

A. General

7.01 Trunk originations are detected and validated by the scanning and input monitor programs. The routine TKORIG then takes over the processing, calls the appropriate translation routine, and processes the returned data. Details of the scanning and origination processing are covered in Section 233-151-125, Input Processing and Scanning Software Subsystem Description.

7.02 Trunk origination translations (Fig. 32) are initiated by a SPN. The SPN is translated by program XSLSPN into equipment location addresses (TEN, SPN, and DTA). Subroutine BUSYCKT is then used to reserve the incoming trunk. An MF receiver (if MF is to be used) is then identified and located using subroutine TRKSEL in the EQPSEL program. The digit processing programs call various translation routines based on the number of digits expected. The digits may be handled directly by 4DIGIT, 1DIGIT and then 4DIGIT when the exact number of digits is not specified, or by 3DIGIT and then 4DIGIT. The final translations are provided by 4DIGIT which returns the terminating translations (TEN, TMAJ, etc) to the TCR. Subroutine TRKSEL locates a ring circuit and returns the TEN, SPN, and DTA. The call processing programs use the translation information to complete the call.

B. Trunk Origination Scan Point Translations

7.03 Translation data for the trunks is obtained by calling subroutine DATCKT in the EQPSEL program and by furnishing the SPN (Fig. 33). The GETGRP subroutine, part of XSLSPN, passes the SPN to the SPNXLA subroutine, which processes the SPN and ultimately returns the first word and address of a universal subtranslator to GETGRP. Subroutine GETGRP then returns the trunk group and member numbers to DATCKT.

7.04 Subroutine DATCKT next calls GRPX and furnishes the trunk group number. Subroutine GRPX returns the group data on the specified trunk. The group data completely defines the features associated with the trunk (Fig. 30 for generic SO-2 and Fig. 31 for generic 3E3).

7.05 Subroutine DATCKT identifies the particular type of circuit and furnishes the member number to CKTLST (Fig. 33), which branches to MBR_LST and GENTRY3 and then to TK_TEN. On completion, the TEN, SPN, and DTA are loaded into the TCR by DATCKT. A return is then made to TKORIG. Subroutine BUSYCKT is then called, and the data is used to reserve the selected trunk by setting the busy and ignore bits.

7.06 After the trunk translation data is stored in the TCR, trunk originations are ready for digit interpretation. Some trunks may require the selection and connection of a receiver; if so, translation routines and data required for these activities are carried on via the subroutine TRKSEL, as described previously in paragraphs 3.16 through 3.22.

C. Trunk Origination Digit Translations

7.07 The incoming digit translation code in the trunk group data (Fig. 30 for generic SO-2 and Fig. 31 for generic 3E3) indicates the number of digits expected and specifies the pretranslation required. Depending on the code, the signal digit in the TCR is set to indicate the number of digits to receive before signaling base level and passing control to one of the following routines:

- (a) TERMX in TERM, which initiates 4-digit translation as described in Part 5.
- (b) 3DTRE in DNTRP, which initiates 3-digit translation as described in Part 4.
- (c) The 1-digit translation routine which calls 1DIGIT in XSL3DG. Subroutine 1DIGIT is used when a variable number of digits is expected. It will return the number of digits expected and auxiliary information if the type of call can be determined at that time. The routine may be called more than once until enough digits have been translated to determine whether the call is locally terminated or tandem. When local termination is indicated, the program TERM is

called and the necessary translation data is secured from MTI address 1DGTBL (Fig. 34).

7.08 On completion of digit translation, the translation processing to locate the proper line and service circuits is the same as described in previous parts.

8. DISCONNECT

A. General

8.01 Program DISCON is responsible for acting on supervision received from a stable call. A stable call is one that has reached the talking state and needs no supervision until some action is performed by one of the parties involved in the call.

8.02 Program DISCON receives control by a supervisory report detected by the input monitor or, for an abandoned call, via another call processing program. Program DISCON reduces all calls to a "talking" path by initializing the TCR from data in the terminal memory record (TMR), just as if the TMR represents a talking path. During initialization of the TCR, line and trunk translations are accomplished through use of the LNDATA or DATCKT subroutines in exactly the same way described in Parts 3 and 7.

8.03 Program DISCON then uses the TCR information to classify the call and to determine what further translations are required, if any. Additional data needed depends on the type of call being disconnected. All calls can be classified as one of the following types:

- (a) Line-to-line, intraoffice, or revertive
- (b) Line-to-circuit, outgoing, "A-party" line service trunk hold, joint hold, or circuit hold
- (c) Circuit-to-line, incoming, party line, or service trunk hold
- (d) Circuit-to-circuit, tandem, or trunk-to-service circuit.

B. Line-to-Line Translations (Intraoffice Calls)

8.04 The proper type of supervision report causes custom call screening to be invoked. This requires complete line information on both "A-party"

and "B-party" lines. Subroutine LNDATA is used to secure this translation data.

8.05 The processing required to secure the proper line data is identical to that previously described in paragraphs 3.03 through 3.17 and Fig. 7 through Fig. 10. Once the line data is secured, it is placed in the reinitiated TCR and used by DISCON to idle the proper lines.

C. Interoffice Calls (Incoming or Outgoing)

8.06 On either incoming or outgoing calls, certain supervision requires custom calling screening on the line, thus requiring translation processing by LNDATA as previously described. An outgoing call is terminated by hardware idling the outgoing trunk and hardware and software idling the path and "A-party" if it is a trunk. This requires that the complete translation pertaining to the trunk be accomplished. Program DISCON acquires the proper data through DATCKT.

8.07 Subroutine DATCKT does translation processing as described in paragraphs 7.03 through 7.06 and Fig. 33 and 34.

8.08 A tandem call (circuit-to-circuit) is handled in the same way as the line-to-circuit, except that the line translations required for custom calling screening are not accomplished.

8.09 In addition, DISCON uses various other translation routines such as IDLECKT, LINEIDLE, BUSYCKT, etc. The usage depends entirely on the call variables. Generally, the primary translations required by DISCON are the same as required during call processing (eg, line or trunk information).

9. SPEED CALLING

9.01 Speed call service is a custom calling service that allows a customer to specify a frequently called number by dialing only one or two digits instead of the normal seven or more digits. Each speed call customer has a list of unique numbers assigned to each abbreviated code. These lists can be modified by dialed instructions from the customer or by service order.

9.02 Two types of speed call service exist. A line with speed call 8 service has a list containing up to eight directory numbers accessed

by dialing one digit (2 through 9). For speed call 30 lines, two digits (20 through 49) are dialed to access one of 30 directory numbers. A line may have either or both types of service.

9.03 A 3-word MTI entry for speed call 8 is SC1LST, which contains the translation store address of the speed call 8 head table. The speed call 8 index obtained from the originating line subtranslator or PBX/MLHG subtranslator indexes the head table to select an entry word. This word contains a store increment value and a bit indicating whether the speed call 8 list pointed to by the entry is vacant or assigned. In generic 3E3 this word does not contain the bit indicating speed call. The entry ID is used for this purpose in the line subtranslator. In the PBX/MLHG translator, which does not have the entry ID, an index of zero is not allowed. The store increment value is used to obtain the correct speed call 8 list for the line. This speed call 8 list contains 24 words (eight 3-word entries) for Issue 4 of the SO-2 generic and contains 16 words (eight 2-word entries) for the 3E3 generic. The speed call 8 list entries contain the directory numbers represented by the code digits dialed by the customer. The index used to select a particular entry in the speed call 8 list is derived from the code digit dialed (index equals three times code index minus six for the SO-2 generic) (index equals two times code index minus four for the 3E3 generic).

9.04 The MTI entry for speed call 30 is SC2LST. The description of SC2LST is the same as that for SC1LST except for the following deviations.

- (a) The speed call 30 index obtained from the line subtranslator line data words or PBX/MLHG subtranslator is used to index the speed call 30 head table.
- (b) The speed call 30 list contains 90 words (30 three-word entries) for Issue 4 of the SO-2 generic and contains 60 words (30 2-word entries) for the 3E3 generic.
- (c) The list index used for selecting the appropriate entry (containing the directory number represented by the 2-digit code dialed) is derived as follows:

Index equals 3 times code minus 60 for the SO-2 generic. The index equals 2 times code minus 40 for the 3E3 generic.

10. CALL FORWARDING

10.01 Call forwarding is an arrangement that allows a local office customer to redirect calls intended for his telephone to another telephone.

10.02 The call forwarding MTI entry, CFTBL (call forwarding table), contains the call forwarding table address in temporary store. The call forwarding table is a one-level table (no head table) consisting of 4-word entries indexed by the call forwarding index for Issue 4 of the SO-2 generic and 1-word entries for the 3E3 generic. This index is obtained from the line data words in the line subtranslator. The call forwarding table for Issue 4A of the SO-2 generic may have a variable number of entries (4 words per customer and 4 words per spare), each containing the number to which the call is to be forwarded, an active entry bit, and information about the call when activating the service. The call forwarding table for generic 3E3 may also have a variable number of entries (1 word per customer and 1 word per spare). Each assigned entry is used when a customer activates the call forwarding of the customer service and contains information pertaining to that activation and also an active bit.

10.03 In Issue 4 of the SO-2 generic, CFALST is an MTI entry containing the translation store address of the call forwarding availability list. This list is used when assigning new call forwarding customers. One bit in the list is used per entry in the call forwarding table (including spares). The bit indicates whether the entry in the call forwarding table is assigned or unassigned.

10.04 In the 3E3 generic, CFLST is an MTI entry that contains the translation store address of the call forwarding digit list. This list contains the call forwarded digits when the service has been activated and also indicates unassigned indexes.

11. MESSAGE REGISTER LIST

11.01 A 3-word MTI entry containing the temporary store address of the message register table is MRTBL. The message index obtained from the line data words of the line subtranslator is used to select a particular word (message register) in the table. The table contains one word per customer with the service plus spares.

11.02 A 3-word MTI entry containing the translation store address of the message register availability list is MRALST. There is 1 word in the list for every 16 words in the message register table. Each word in the list contains 16 bits. There is one bit per message register to indicate whether the message register is assigned or unassigned. This list is used when assigning new customers. The MRALST is deleted in the 3E3 generic.

12. TRIPLET INDEX INFORMATION

12.01 The MTI entry COINTRIP (coin triplet) contains the translation store address of the coin triplet address table. This table has a maximum of 1024 words or one word per three coin lines plus spares. The coin triplet address table is indexed by the 10-bit triplet index obtained from the line data words of the line subtranslator. Except for vacant entries, all entries in the table contain a coin triplet address and the individual assigned points. This table determines which triplet is being used.

12.02 The MTI entry COINSTAT contains the temporary store address of the coin triplet status table. This table is used in determining which point in the triplet to access. The coin triplet status table has a maximum of 256 words or one word per 12 coin lines plus spares. The table is indexed by bits 2 through 9 of the triplet index to obtain the desired word. Bits 0 and 1 of the triplet index are used to select one of four triplets in the word. The 2-bit triplet point number selects one of three status bits corresponding to the three distributor points in the triplet.

13. CHARGE INDEX EXPANSION

13.01 The charge index obtained from the screening tables consists of a type bit (high order bit) and a charge field. The type bit is used to denote AMA (equals zero) or local charge (equals one). A combination of a type bit equal to zero and the charge field signifies call per the following:

CHANGE FIELD	CALL
0	Illegal
1	Free
2	WATS band 0 special

CHANGE FIELD	CALL
3	WATS band 1 interoffice
4	WATS band 3 interoffice
5	WATS band 3 interface
6	WATS band 4 interoffice
7	WATS band 5 interoffice
8	WATS band 6 special
9	WATS band 7 interoffice
10	WATS band 8 intraoffice
11	WATS band 9 interoffice
12	USRP
13	Reserved
14	Unused
15	Toll call

When the type equals one, the charge field is used as an index into the charge table, CHARGETB. Also, the charge indexes 16 through 31 are the message billing index (MBI) for an office with AMA. The particular charge index is used for charging a coin call.

13.02 CHARGETB is a 16-word table of 1-word data entries. There are three types of entries in CHARGETB designated by a 2-bit field as follows.

- (a) **Type = 00:** This type denotes a vacant entry.
- (b) **Type = 01:** This type denotes a coin entry with initial time and charge fields and overtime time and charge fields. For coin calls, there are fixed initial and overtime charges determined by the coin phone; hence, both these fields are set to zero. When the initial time interval has expired, a recorded announcement is given requesting the overtime charge. If the coin is deposited, the call continues, etc. If a coin is not deposited within a specified time,

the connection is broken. The time field has the following options:

- (1) Time equals 0—local untimed coin call
- (2) Time equals 1 through 7 minutes—local overtime call.
- (c) **Type = 10:** This type denotes a message rate (MR) entry. This entry also has initial time and charge fields and overtime time and charge fields. The time field is defined as follows:
 - (1) Time equals 0—local untimed MR
 - (2) Time equals 1 through 7 minutes—timed MR.

The charge fields can have a value from 1 to 15, indicating the number of message units to charge for timed or untimed MR. The initial charge is made at the beginning of the call and, when the initial time has expired, the overtime parameters are used. The overtime charge is made at the beginning of the overtime period, and time is clocked using the overtime time, etc, until the call is completed.

14. ROUTE INDEX EXPANSION

14.01 Final information needed for local call completion and distant office call routing is found in the route index expansion.

14.02 A 3-word translation store MTI entry, RIEXP (route index expansion), contains the address of the route index expansion tables. The route index expansion tables consist of eight possible types of 2-word entries. The route index obtained from 1-, 3-, or 4-digit translation is used to select a particular 2-word entry from a maximum of 256 entries (512 words) in the table.

14.03 A description of each possible type of entry in the route index expansion table is as follows.

- (a) **Type = 0:** This type means unassigned.
- (b) **Type = 1:** This entry type is for intraoffice calls. The first entries in the route index expansion table (route indexes 0 through 7) are of this type. Route indexes 0 through 7 are

called normalized office codes (NOCs). The first word of entry type 1 contains the type and a cutover route index and cutover route index bit which are used for routing calls intended for directory numbers not yet cut over. The second word of the entry contains the office-to-office test route index. The second word in the 3E3 generic also contains the route index to be used when a line is assigned to intercept by recent change.

(c) **Types = 2 through 5:** All these entry types have the same entry format. The first word of the entry contains the type and 3-digit prefixing fields used to modify the dialed information by prefixing the call number before it is outpulsed to a distant office. The second word of these entries contains the following parameters.

- (1) A 3-bit number (DLT) indicating the number of digits to delete (does not include prefixes which are also deleted).
- (2) A bit (ARP) to show whether an alternate route index is provided.
- (3) A bit (FREE) used to determine whether there is a charge associated with the route.
- (4) An 8-bit trunk group number specifying the trunk group that should be used for completing the call. Alternate route indexing (Part 15) is used if the specified trunk group is busy. Entry types 2 through 5 are defined as follows:
 - Type = 2- through 10-digit interoffice (no overlap outpulsing)
 - Type = 3- through 10-digit interoffice (overlap outpulsing permitted)
 - Type = 4- through 7-digit interoffice (no overlap outpulsing)
 - Type = 5- through 7-digit interoffice (overlap outpulsing permitted).

(d) **Type = 6:** Two different formats exist for this type of entry. The first format uses a 2-word expansion table to convert dialed digits to a 4-, 5-, 6-, or 7-digit number. This

entry format has table and entry numbers for accessing the expansion table. The second word contains a destination code of 8, indicating code conversion, and 3-bit field (DIGCN), indicating the number of digits stored in the expansion table (3 through 7).

(e) The second format for type = 6 is used when a 3-digit service code or 7 digits are dialed for line tests and for customer dialing errors. The first word of this entry format contains only the type. The second word is used to specify the trunk group number to be used (no alternate route indexing needed), whether the route has a charge associated with it, and the class-of-service tone to the completing operator (high or low tone), if needed. This word also contains a destination code. The destination codes are defined as follows:

- (1) 0—vacant circuit group
- (2) 1—vacant code operator
- (3) 2—no outpulsing trunk group
- (4) 3—tones which time out
- (5) 4—tones with no time-out
- (6) 5—announcement local
- (7) 6—route to reorder
- (8) 7—station ringer test
- (9) 8—conversion of dialed digits to a 4-, 5-, 6-, or 7-digit number
- (10) 9—balance line test
- (11) 10—interrupted milliwatt test line
- (12) 11—synchronous test line
- (13) 12—loop-around test line 0
- (14) 13—loop-around test line 1
- (15) 14—short circuit test line
- (16) 15—open circuit test line
- (17) 16—charge test line

SECTION 233-151-150

- (18) 17—continuous milliwatt test line
- (19) 18—AC/DC open circuit test line
- (20) 19—spare 2
- (21) 20—autoconnect
- (22) 21—ROTL input terminal (3E3 only)

(f) **Type = 7**—Interoffice (for outpulsing received digits) which contains the same data field as types 2 through 5.

14.04 Route indexes 8 through 20 and 22 are for Issue 4 of the SO-2 generic. Route indexes 8 through 20 are predefined. Route indexes 21 and 23 through 31 are for Issue 4 of the SO-2 generic and 21 through 31 for the 3E3 generic. These route indexes are reserved and cannot be used.

15. ALTERNATE ROUTE INDEXING

15.01 A translation store 3-word entry, ARIEXP, contains the address of the alternate route index list. Route indexes 8 through 255 (interoffice) are converted for ARI; hence, there is a maximum of 124 words in the list with two route indexes per word.

15.02 Route index 8 [route indexes 0 through 7 (intraoffice) are not used for ARI] is the first route index used to index the alternate route index list. The low bit of the route index selects which half (which alternate route index in the entry word) to use. The selected alternate route index is then used to reaccess the route index expansion table as a new route index.

15.03 Two alternate routes are possible before the third alternate route directs the call to reorder.

16. ONE-DIGIT TRANSLATION

16.01 When a variable number of digits is expected on an incoming trunk, one-digit translation is used to individually examine each digit received until the destination of the incoming call and the number of digits expected can be determined.

16.02 The one-digit translation entry, 1DGTBL, is a 3-word MTI containing the number of

one-digit translation tables, the initial translation store address of the tables, and bit XSL, indicating whether the translator exists or is undefined.

16.03 The one-digit translator tables are indexed by a table selector and the dialed digit. The table selector is a 6-bit number obtained from the trunk group data table or from another one-digit translation table entry.

16.04 There is a maximum of 64 one-digit translation tables of ten words each for a total of 640 words maximum. The table selector multiplied by 10 (since there are ten words per table) is used to select the correct data table.

16.05 The dialed digit is then used to select the desired entry in the table. Each entry is one word in length (ie, ten entries/table). All entries in the tables have the same format which includes auxiliary information (AUXINFO), the total number of digits to expect (NDE), and an entry type.

16.06 If type equals zero, a dialing error has occurred from an invalid digit being dialed. NDE is zero, and AUXINFO contains a route index. The designated route index is returned to call processing by its translation routine.

16.07 When type equals one, the initial digit cannot be conclusively translated, and AUXINFO contains a table selector to index into another one-digit translation table with the next dialed digit (NDE equals zero).

16.08 When type equals two, the call is a locally terminating call. NDE contains the number of expected digits, and AUXINFO contains the NOC for the local office. NOC is then used in the 4-digit translation.

16.09 When type equals three, the call is a local tandem call. NDE contains the number of expected digits and AUXINFO contains the route index the call is to go on.

17. GROUP TRANSLATION

17.01 Group translation is used to give detailed information concerning a trunk group, service circuit group, or PBX/MLHG group. GRPTBL (group table) is the MTI entry for group translation.

17.02 The input to GRPTBL is the binary equivalent of the group number. The group number is a 3-digit decimal number (000 through 255) which is assigned to a trunk group, service circuit group, or PBX/MLHG group. The group number (eight binary bits) is obtained from the SPN translation, from NOCD translations, or from the route index translator. The two high order bits of the group number (bits 6 and 7) are used to access one of four 3-word entries in GRPTBL.

17.03 The first entry contains the PBX/MLHG table address in translation store. The second entry in GRPTBL contains the service circuit group table address in translation store. The third and fourth entries in GRPTBL contain the translation store addresses of the trunk group tables.

A. PBX/MLHG Group Table

17.04 The low six bits of the PBX/MLHG number are used to access a particular 8-word entry in the PBX/MLHG group table.

17.05 This table can contain up to 64 eight-word entries for a maximum of 512 words. The 8-word entry tells whether the group exists and whether it has any, some, or all of the following characteristics:

- Three-way service
- Service observing
- Stop hunt feature
- TOUCH-TONE dialing
- Night make-busy features
- QZ billing
- An essential line
- A ground start line
- Speed call service
- Ability to change speed call lists
- Traffic schedule C, H, or D.

17.06 The 8-word entry also contains the originating major class, the screening class, speed call indexes, night stop information, stop hunt data, largest member number, largest hunt member number, and billing number. A selection status block index is provided for indexing the group status block in temporary store. A member list index is also supplied for indexing the member lists. In Issue 4 of the SO-2 generic a line class code index is supplied for indexing the nonresident line class code table. In the 3E3 generic a line class code index is supplied for indexing the resident line class code table. Both the nonresident and resident line class code tables contain the line originating and terminating major classes, rate information, screening class, party number, and ASCII characters used by the telephone companies.

17.07 The EXB bit is new in the 3E3 generic. If EXB = 1, the PBX/MLHG has SPNs assigned for remote make busy, stop hunt, and/or night stop. The SPNs are stored sequentially in an expansion entry. The size of the expansion needed is determined by the quantity of scan point keys. When the key SPN data needs more than eight entry words, the EE bit in the last word of the expansion entry is set (EE=1).

B. Service Circuit Group Table

17.08 The low six bits of the group number assigned to a service circuit group are used to index the service circuit group table. This table consists of a maximum of 64 four-word entries.

17.09 The entries supply a selection status block index, a member list index, a circuit code (see Table C for list of codes), the largest member number, traffic schedule bits indicating which schedule (C, H, or D), and information as to whether the group has at least one member or even exists.

C. Trunk Group Table

17.10 The low six bits of the group number assigned to a trunk group are used to select a particular entry within the trunk group table. There are actually two trunk group tables with a maximum of 512 words in each table for a total of 128 eight-word entries.

17.11 The first four words of the 8-word trunk group table entry are identical to the service circuit group entries in format and kinds of

SECTION 233-151-150

information supplied. The last four words of the 8-word entry contain detailed information about the trunks in the group. This information tells whether the trunks in the group:

- Have multifrequency outpulsing
- Have dial pulse outpulsing
- Permit overlap outpulsing
- Require long loop pulsing
- Are no test trunks
- Are 1-way or 2-way trunks (must be one or the other)
- Are E and M or loop trunks (one or the other)
- Are from or to a local test desk
- Are from a step-by-step office
- Expect MF inpulsing from a far office
- Have a charge for incoming calls
- Have multiwink or MF inband signaling
- Return audible
- Permit interruption of pulsing on outgoing trunks
- Do not expect answer supervision on outgoing trunks
- Have dedicated facilities provided for the local test desk or if nondedicated remote test equipment facilities are provided
- Are regular, CAMA, TSP, or TSPS trunks
- Are 911 trunks
- Have diagnostics run or not
- Type of diagnostics to run.

17.12 The three 3E3 generic changes in the trunk group table are:

- (1) The TDM (tandem) bit is deleted and not replaced.
- (2) The WINK bit is deleted and replaced by the EBS (Emergency Bureau Service) bit.
- (3) The MWS (multiwink signaling TSPS) bit and the INBD (inband signaling) bit are deleted and replaced with the SIG (type of signaling) bit.

17.13 The entry also identifies the trunks as immediate start, delayed dial, or wink start. Two types of wink start trunks exist—one which requires waiting 350 msec for the end of the wink and one that requires waiting 1000 msec for end of wink.

17.14 Auxiliary information for initial digit translation on incoming calls is provided in the field ID_AUX of the entry. Interpretation of this field is based on the incoming digit translation code (ID_XLN) also found in this 8-word entry. ID_XLN directs the translation of the incoming digits by one of the following methods.

- (a) Use 4-digit translation on the first four digits received. ID_AUX contains the NOC in bits 0 through 2.
- (b) Ignore the first digit and use 4-digit translation on the next four digits received. ID_AUX supplies the NOC in bits 0 through 2.
- (c) Use 1-digit translation on the first digit received. ID_AUX contains a table selector number in bits 0 through 5.
- (d) Use 3-digit translation on the first three digits received. ID_AUX contains the screening class in bits 0 through 4.
- (e) Form a 4-digit number from the most significant (thousands) digit and three received digits, and then use 4-digit translation. ID_AUX supplies the thousands digit in bits 3 through 6 and the NOC in bits 0 through 2.

17.15 The type of disconnect supervision required is also supplied in these four words of entry. Possibilities include joint hold (recording

completion operator), service hold (toll switch, TSP, or no test), customer hold (nonoperator intercept trunks), or regular trunk group.

17.16 A 6-bit index, TOTANIDX (terminal office test access number index), is contained in the last word of this entry. This index is used to index the terminal office test access number table used in conjunction with trunk testing, using the synchronous or asynchronous test lines or permanent busy test lines in connecting offices.

18. MEMBER LISTS

18.01 The member lists are used to provide the terminal equipment number or the distributor triplet address and scan point number or service circuit number of each member in a group.

18.02 The MEMLST (member list) is a 12-word MTI entry containing the translation store addresses of the member lists. The MEMLST is indexed by the two high bits of the group number to select one of four 3-word entries containing the member list addresses. The member lists are as follows:

- (a) A PBX member list containing one word per member plus a header word
- (b) Two trunk member lists, each containing two words per member plus one header word
- (c) A service circuit list having 1-1/2 words per regular service circuit member plus one header word and one word per tone and announcement service circuit member plus one header word.

18.03 The header word in each list has the same layout. This word contains the format type (zero through three) of the list, the number of members in the list, and the number of spares in the list.

18.04 The PBX member list has a format of zero. Each word in the list contains the member terminal equipment number and the number of the remote make-busy key associated with the member. Operation of a remote make-busy key at the customer premises makes a preselected set of members associated with the key appear busy to incoming

calls. The member list index obtained from the PBX/MLHG data is used to index the list.

18.05 The service circuit member list is comprised of a regular service circuit member list and a tone and announcement service circuit member list. The tone and announcement service circuit member list has a format type 1. The list is indexed by the member list index obtained from service circuit group data to select the circuit member associated terminal equipment number.

18.06 The regular service circuit member list (format type 2) contains the circuit member associated service circuit number and distributor triplet address.

18.07 The trunk circuit member list (format type 3) contains the member trunk scan point number and distributor triplet address. The member list index to this list is obtained from the trunk group data.

19. SELECTION STATUS BLOCKS

19.01 Status blocks for each circuit type are provided in temporary store. GRPSTAT is a 12-word MTI entry containing the temporary store addresses of the PBX status blocks, the service circuit status blocks, and the two trunk circuit status blocks. GRPSTAT is indexed by the two high bits of the group number to select one of the four 3-word entries that point to the four status blocks.

19.02 The status blocks are indexed by the selection status block index (SSBI) obtained from the group data.

19.03 The blocks provide for recording the peg counts, usage counts, and overflow counts used in traffic measurement. Status of the all circuits busy, night make busy, remote make-busy key, and stop hunt features is recorded in the status blocks for PBX/MLHG groups. An all circuits busy status, a group not in service indicator, a maintenance busy counter, and the last circuit idled is provided for service and trunk circuits. Except for incoming trunks, there is only the maintenance busy counter. In addition, there is a busy/idle bit for each circuit in the group. These bits are checked when an idle circuit is required, and the bits are marked busy or idle as a circuit is seized or released.

20. SCAN LIST

20.01 A 258-word MTI data entry for Issue 4 of the SO-2 generic and the 3E3 generic is CKTSCAN (circuit scan). CKTSCAN consists of a list of all circuits that must be regularly scanned with the exception of lines, junctors, and CDPRs. This scan list is actually three scan lists in one:

- (1) A bylink trunk scan list for bylink trunks
- (2) An operator scan list for operator trunks
- (3) A slow scan list for other trunks and service circuits.

20.02 One-fourth of the bylink and operator trunks is scanned each interrupt (each interrupt is 10 msec, so all are scanned in a 40-msec period). The slow scan list is scanned once every base level loop.

20.03 Rows 6 through 31 of the master scanner (scanner 0) and rows 28 through 31 of scanners 1 through 15 are listed in the circuit scan table. Rows 28 through 31 of scanners 0 through 15 are allocated for assignment to trunks and miscellaneous circuit scan points. Rows 6 through 27 of scanner 0 list regular and superimposed ringing circuit scan points, some slow scan trunk scan points if needed, power and alarm scan points, first supervisory or directed scan point, and other miscellaneous scan points.

20.04 Each entry in the list is a 3-word entry, and all entries have the same format. Entries 22 through 85 of CKTSCAN are referred to as TRKSCAN (trunk scan) for convenience in indexing to the trunk and miscellaneous service circuit area of the circuit scan table.

20.05 The entry consists of three 16-bit words for each row of scan points. The first word is for bylink and operator scan points, the second word for operator-only scan points, and the third for slow scan points.

20.06 The result is 16 vertical columns, three rows high, for each row of scan points. The type of scan point is thus identified by which bit is set in a particular vertical column.

21. TERMINAL OFFICE TEST

21.01 The 3-word MTI entry, TOTANTBL, points to the terminal office test access number table. This table contains a maximum of 256 words. All entries in the table contain four words. Each entry contains two telephone numbers, one for the permanent busy test line and one for the operational test line. Both of these telephone numbers are for use with the trunk and line test panel (TLTP). Also included in the table are bits to indicate whether the permanent busy test or operational test is provided. Another bit describes the operational test as synchronous or nonsynchronous. Table entries are accessed by the 6-bit test access index, TOTANIDX, obtained from the trunk group data. TOTANIDX equals 0 indicates that the automatic program test is not available. The first entry (entry 0) of the translator maintains a header entry which contains a maintenance billing number and three NPAs (numbering plan areas).

21.02 The 4-word MTI data entry, NOTEST, contains the even and odd TENs for both the circuit test multiple and the wire test multiple.

22. PRECUT

22.01 The 3-word MTI entry, CUTSTAT, points to a status map table which indicates the open or closed state of the ferroids for test lines and nontest lines. This map is used in the precutover state and is needed to enable calls to be completed to certain lines before cutover. The status map table contains one bit per TEN in the office.

23. AUTOCONNECT

23.01 The autoconnect translation tables consist of two translators, the MSGCL (message class table) MTI entry, eight 3-word, type 1 data words containing autoconnect numbers, and CALLBACK (call back number translator).

23.02 The MSGCL is accessed, via line translators when the originating major class indicates autoconnect, and is indexed by a message class number (MCNO). This table contains the TTYC number, the miscellaneous controller assignment, dedicated autoconnect assignment, autoconnect trigger number, and the automatic autoconnect indicator.

23.03 CALL BACK is an index table containing:

- Time-out information
- Message class (MSGCLS)
- Designated TTYC
- Designated TTYC port
- No dial back information (NDB)
- Tone information
- Automatic dial out data (ADO)
- Function performed by user
- Priority of autoconnect facilities
- Terminal equipment number (TEN).

The remaining three words contain fields for ten BCD digits. CALLBACK is indexed by an index developed by 4-digit translations.

23.04 The message class number assignments have changed in the 3E3 generic as shown in the following assignments:

- 0 = Maintenance
- 1 = Backup Maintenance
- 2 = Service Orders
- 3 = Traffic
- 4 = Repair Service Bureau
- 5 = Office Record
- 6 = AMA Primary (Issue 4 of SO-2 Generic)
- 6 = Regional Accounting Office (3E3 Generic)
- 7 = AMA Backup (Issue 4 of SO-2 Generic)
- 7 = Spare
- 8 = AMA Primary (3E3 Generic)
- 9 = AMA Backup (3E3 Generic)

10 = ROTL Data (3E3 Generic)

11-15 = Spare.

23.05 The Issue 4 of SO-2 generic functions performed by the user have the following assignments:

0 = TTY

1-2 = Reserved

3 = Spare

4 = Local Test Desk Trunk

5 = Spare.

6-7 = Spare.

23.06 The 3E3 generic functions performed by the user have the following assignments:

0 = TTY

1-2 = Reserved

3 = ROTL data

4-7 = Spare

8 = Local Test Desk Trunk

9 = Remote Record Overload Announcement

10 = ROTL Security

11-15 = Spare.

24. GROWTH PARAMETERS

24.01 The MTI data entry BOSCAN is a 1-word data entry containing the number (BOSCAN) of the last logical scanner that contains scan points assigned to bylink and/or operator trunks.

24.02 The MTI data entry LOG_SCAN is a 1-word data entry containing the last equipped logical scanner number (LOGSCAN).

24.03 The MTI data entry NETWEQPT is a 1-word data entry containing the highest equipped controller number (NETW).

SECTION 233-151-150

24.04 The MTI data entry EQPT_PD is a 32-word data entry. This data entry is a peripheral decoder status map used to indicate the quantity of peripheral decoder equipment in the office.

25. NOC TO OFFICE CODE CONVERSION

25.01 An 8-word MTI data entry, NOCBCD, is used to obtain the office code represented by a NOC. Each word in the table contains three digits associated with an NOC. The NOC is used to search the table to locate its corresponding 3-digit office code.

26. MISCELLANEOUS OFFICE PARAMETERS

26.01 The one-word MTI data entry, SCRSIZE contains the number of screening classes provided.

26.02 A 2-word data entry in the MTI is TLTPL. The first word contains the first terminal equipment number assigned to the trunk and line test panel (TLTP). This number (TEN 1) corresponds to access trunk 1 for the TLTP. The second word contains the second terminal equipment number (TEN 2) assigned to the TLTP corresponding to access trunk 2 of the TLTP.

26.03 A 2-word MTI data entry referred to as an office option word is OFF_DATA. This word tells whether the office allows 0+ dialing, has ac-dc or superimposed ringing, or has dial pulse receivers provided. A bit (MOT) is provided and is defined as follows:

MOT = 0 — +130 volts collect, -130 volts return

MOT = 1 — -130 volts collect, +130 volts return.

Also, a bit (AMA) is provided and is defined as follows:

AMA = 0 — office does not have LAMA

AMA = 1 — office has LAMA; there are no software message registers.

26.04 The 3E3 generic changes to the office option word are:

- (1) Delete the ALC (denotes old or new ALIT circuit) bit and the IDDD bit

- (2) Add the ICF (1 = printing of customer-dialed call forwarding changes are inhibited) bit

- (3) Another word has been added for future use. The first bit of this word indicates whether or not network analysis is taking place.

26.05 An 8-word MTI entry data table, OFFICE, is used for identification of the office. The table contains 6 to 14 teletypewriter codes for the office identification number. (The office identification could be from 6 to 14 characters in length.) The number of characters used for identification is stored in the first word of the table.

26.06 A 4-word MTI entry, NOTEST, contains odd and even TENS for both circuit and wire tests.

26.07 A 2-word MTI entry, FETL (far end test lines for ROTL), contains the first and second terminal equipment number assigned to ROTL 105 test lines. FETL is only found in the 3E3 generic.

26.08 A 1-word MTI entry, DTDADTA (dialtone detector alarm data), contains the distributor triplet address for the dial tone detector alarm. The DTDADTA is only in the 3E3 generic.

26.09 A 1-word MTI entry, TUTTL (trunk under test terminal for ROTL), contains the terminal equipment number assigned to the ROTL trunk under test terminal. TUTTL is only in the 3E3 generic.

27. LINE CLASS CODE

27.01 The line class code (LCC) translators are nonresident (tape stored) tables in Issue 4A of generic SO-2 and are resident tables in generic 3E3. These translators are used to provide a cross-reference between assigned telephone numbers and their associated line class codes. The telephone number to line class code conversion table (TNXLCC) contains two 8-bit LCC index entries per word, and the table may be as large as 16,384 words. TNXLCC does not exist in the 3E3 generic. The LCC index is in the NOCD hundreds group entry.

27.02 The LCC_TBL is a 256-word table containing 3-word entries. The entry contains ASCII format information, party number, rate, originating major class, terminating major class, and screening

class for a specific telephone number or numbers. The LCC_TBL is indexed by an 8-bit LCC index from TNXLCC in Issue 4 of the SO-2 generic or from NOCD hundreds group entry in the 3E3 generic.

28. SCAN POINT NUMBER TRANSLATION (XSLSPN) PROGRAM

28.01 The scan point number translation (XSLSPN) program translates SPNs. This program contains subroutines which obtain the originating line data, when given a TEN, and obtain data associated with the line data. These subroutines also obtain (given a SPN) the other SPN, the other scan points, and the other data in the scan point translator associated with a circuit.

28.02 The following paragraphs briefly describe the subroutines contained in the XSLSPN program.

A. Call Forwarding Expansion (CF_EXP) Subroutine

28.03 The CF_EXP subroutine is called to obtain a call forwarding entry. The entry condition for this subroutine in the call forwarding index is word 2 of SCRATCH (Fig. 35 for Issue 4 of the SO-2 generic and Fig. 36 for the 3E3 generic). The index is put into SCRATCH by calling the subroutine LNDATA. Therefore, CF_EXP can only be called after LNDATA is called. The exit condition from CF_EXP is the first word of the call forwarding entry in register 1 with register 0 indicating successful completion and registers 14 and 15 containing the address of the entry.

28.04 The CF_EXP subroutine obtains the call forwarding word from SCRATCH. A test is made to see if there is a call forwarding entry. If there is not an index, then an error is returned to the calling program; but if there is an index, then the subroutine continues. The index is isolated from the obtained word. For the 3E3 generic the index is then stored in CCDIGXLA (a temporary store in ATSD). The address and first word of the call forwarding MTI entry (CFLST) for generic 3E3 or entry (CFTBL) for generic SO-2 is obtained. The index, address, and first word are the entry conditions for the subroutine GNTRY3 in the XSLSUB program. The GNTRY3 subroutine then actually obtains the call forwarding entry indicated by the index.

28.05 Upon exit from GNTRY3 the check is made for any errors that GNTRY3 could have found in the translation. Then CF_EXP returns the address of the call forwarding entry and a return code of 1 for success.

B. Class A Line (CLASSA) Subroutine

28.06 The CLASSA subroutine is called to find the class of a line. This subroutine is called when dynamic service protection is in operation. The entry condition into CLASSA is the TEN of a line. The CLASSA subroutine exists giving an error or the type of line (either A or B).

28.07 The CLASSA subroutine calls the LNDATA subroutine and upon its return checks for the error return. If there is no error, then word 1 of SCRATCH (Fig. 35 for Issue 4 of the SO-2 generic and Fig. 36 for the 3E3 generic) is obtained and the essential line (EL) bit is tested. If this bit is set, then the line is a class A line. If the bit is 0, then the line is a class B line. A return code of 1 indicates class A and a return code of 2 indicates class B.

C. Call Forwarding (CLFRD) Subroutine

28.08 The subroutine CLFRD is called to obtain the call forwarding entry of a line. The entry conditions into CLFRD are the TEN of a line and the tip data request bit. The exit condition is a return code of 0 for error or a return code of 1 for success in register 0 with the return of the first word and address of the call forwarding entry in register 1, 14, and 15, respectively.

28.09 Upon entry into CLFRD the LNDATA subroutine is called using the entry conditions for CLFRD. After the return from LNDATA, CLFRD tests for an invalid line (error) or an unassigned line. If neither of these two errors occur, then CLFRD returns an error to the calling program. If the data of a line is obtained successfully, then CF_EXP is called expecting to find the call forwarding index in SCRATCH (Fig. 35 for Issue 4 of the SO-2 generic and Fig. 36 for the 3E3 generic) where LNDATA stored it. After CF_EXP returns, CLFRD tests for success or error. The possible error situations are: the line has no call forwarding index, the index is out of range, or the translation data for call forwarding is not right. If there is an error, then an error is returned

to the calling program. If the entry is obtained successfully, then the address of the entry is saved and returned with the first word of the entry and the successful return code is returned.

D. Coin Triplet Index Expansion (CNI_EXP) Subroutine

28.10 The CNI_EXP subroutine is called to obtain a coin line distributor triplet address entry. The entry condition is the coin triplet index in register 1. This index is part of the originating line data and is obtained through the subroutines TENDATA or LNDATA. The exit conditions are the coin triplet entry in register 1 and its address in registers 14 and 15.

28.11 Upon entry into CNI_EXP, CNI_EXP obtains the first word and address of COINTRIP. The 3-word MTI entry in TDATA for the coin triplet entries is COINTRIP. The first word of COINTRIP is in register 1 and its address is in register 14 and 15 for the call to GENTRY3. The GENTRY3 subroutine in program XSLSUB is then called and upon its return, CNI_EXP tests for a successful return code. If GENTRY3 was not successful, then the error return code of 0 is returned in register 0. The possible errors are: the index is out of range or the COINTRIP entry data is invalid. If GENTRY3 is successful, then the address of the coin triplet entry is saved in registers 14 and 15 and is passed back to the calling program. Also, the 1-word entry is in register 1 with a return code of 1 in register 0.

E. Coin Line Operated and Release (COINOPR, COINRL, and COINEN) Subroutines

28.12 The COINOPR, COINRL, and COINEN subroutines are called to operate or release the coin line circuit associated with the coin line TEN. The COINOPR subroutine operates the relay and COINRL and COINEN subroutines release the relay of the circuit. The entry condition for COINOPR and COINRL is the TEN in register 1 of the coin line. The entry condition for COINEN is that the coin triplet address stored in word 9 of SCRATCH (Fig. 35 for Issue 4 of the SO-2 generic and Fig. 36 for the 3E3 generic) by the TENDATA subroutine is loaded into register 3. The exit condition for all three subroutines is the same. Register 0 contains return code. When the return code is 0, then the TEN could not be translated into the line data or the subroutine GENTRY3 returned an error. When the return code

is 1, then the line has no coin triplet associated with it. When the return code is 2, then the distribute order failed. When the return code is 3, then the distribute order succeeded.

28.13 Upon entry into COINOPR, the OPERATE register is zeroed to indicate that the relay of the circuit is to be operated. Then subroutine TENDATA is called with the TEN in register 1. When TENDATA returns, COINOPR checks for the error return code. If it is an error, then error is returned to the calling program. If the line data is successfully obtained, then the triplet index word from SCRATCH is obtained. The high 4 bits of this word contain the identifying code of 9, the low 2 bits indicate which one of three points are associated with the circuit, and the other 10 bits are the index. The word is put into register 3. The register is tested to see if it contains an index for the line indicated. If the index does not exist, the return code of 1 is returned to the calling program. If it does exist, the subroutine continues.

28.14 Upon entry into COINRL, the OPERATE register is set to 15 to indicate that the relay of the circuit is to be released. The subroutine COINRL then continues as described for COINOPR with the call to TENDATA except where noted.

28.15 Upon entry into COINEN, the OPERATE register is set to 15 to indicate that the relay of the circuit is to be released. The subroutine COINEN then becomes identical to COINRL at the point where the triplet index is put into register 3.

28.16 The coin triplet index is used to index into two tables, the coin status table and the coin triplet address table. The data is obtained from the status table first. The status table contains the status of the coin line circuits. Each word contains the status of 12 circuits. The low 2 bits of the index and the 2 bits indicating the point are used to obtain a particular status bit. The other 8 bits of the index are used to obtain the word in the table. To obtain the status data, the 8 bits of the triplet index are isolated and loaded into register 1. The 3-word MTI entry COINSTAT is used to obtain the status table. The first word is loaded into register 0 and the address of COINSTAT is loaded into registers 14 and 15. The entry conditions for GENTRY3 have now been met. The subroutine is called and upon its return the return code is tested to see if GENTRY3 was

successful in obtaining the status word from temporary store. If it was not successful, then the error return code of 0 is returned to the calling program. If it was successful, then the status word is saved as it was obtained and the address of the status word is saved also. The OPERATE register is tested to see if the relay is to be operated or released. If the relay is to be operated, then the proper status bit indicated by the low 2 bits of the index and the point bits are set in the status word. If the relay is to be released, the status bit is zeroed. The status word is then stored back into the status table.

28.17 The data is obtained next from the triplet address table. The 10-bit index is used to obtain the 1-word entry from the table. The index is loaded into register 1 for the call to the CNI_EXP subroutine. When CNI_EXP returns, the error condition is tested as previously described. Register 1 now contains the coin distribution triplet address word. The address is then isolated in register 1.

28.18 Next the distribute order needs to be sent to the periphery. The status word is adjusted according to the 2 low bits of the triplet index. The status word is loaded into register 0. Registers 0 and 1 have now been set up for the call to the DSTRB0R subroutine in the PSUBS program. This subroutine issues the order for the distributor triplet address with the status of the three relays as indicated in register 0. Upon the return from DSTRB0R, a two is added to register 1 with the return code indicating success or failure. If the order did not fail, the status bits would indicate the state of the relays as they should be, not as they are.

F. Data Processing (DATAPR) Subroutine

28.19 The DATAPR subroutine obtains the line data from the expansion entries and puts the data into SCRATCH. Normally, the DATAPR subroutine is only called by LNDATA and TENDATA subroutines, but DATAPR can be called by any subroutine or program. The entry conditions are the word count in register 0 and the address of the block of data to be obtained and stored in SCRATCH in registers 14 and 15. The first word of the block of data is not used by DATAPR. The exit conditions are the return code in register 0 (which is 0 for error and 1 for normal) and the data in words 1 through 10 of SCRATCH.

28.20 The DATAPR subroutine obtains the address of SCRATCH in register 12 and 13. The word count in register 0 indicates the number of words to be processed. When the word count is zero, then a normal return is made. For Issue 4 of the SO-2 generic, when the word count is either one, three, or eight, there are either one or three words to process from a 2- or 4-word block, respectively. For the 3E3 generic, when the word count is either one or three, there are either one, three, or seven words to process from a 2-, 4-, or 8-word block, respectively. Each word to be processed contains an entry identification code (ENTRY_ID) in the four high bits. The ENTRY_ID (Table D) indicates the type of data in the word and usually indicates the word location in SCRATCH (Fig. 35 and 36) where the data is stored.

28.21 After the test for zero, the next data word is obtained and the ENTRY_ID is isolated and range-checked against the maximum, which is 11 for generic SO-2 and 12 for generic 3E3. If the ENTRY_ID is out of range, an error is returned. If the ENTRY_ID is in range, a test is made on the ENTRY_ID for 0, 1, and 11. If the ENTRY_ID is zero, then DATAPR loops back to the test for zero word count. If the ENTRY_ID is one, the entry contains a table number and an entry number. The DATAPR subroutine then calls the EXPAND subroutine in the XLSUB program with the numbers in register 1 for entry. For Issue 4 of the SO-2 generic the EXPAND subroutine can only return a 1-, 2-, or 4-word data block. For the 3E3 generic the EXPAND subroutine can only return a 1, 2-, 4-, or 8-word data block. After EXPAND returns, DATAPR checks for the zero error condition. If EXPAND is successful, the return code from EXPAND is tested for a three in Issue 4 of the SO-2 generic. If the return code from EXPAND is three, the first word of four words is to be processed. In the 3E3 generic, if EXPAND is successful, then the return code from EXPAND is tested for a four. If the return code from EXPAND is four, the first word of eight words is to be processed and the word count is charged to seven. The three is now the word count. Then DATAPR returns to the point of isolating the ENTRY_ID and continues to process the first word of the next three words. If the return code from EXPAND is not three, one is subtracted from the return code so that the word count becomes a one for the 2-word data block. The first word is to be processed next. The DATAPR subroutine returns to the point of isolating

SECTION 233-151-150

the ENTRY_ID and continues to process the first word and the next word.

28.22 If the ENTRY_ID is 11, the data is for an individual member of a PBX or multiline hunt group. The originating major class and the screening class are put into the appropriate bits in word 1 of SCRATCH. If the TOUCH-TONE indicator in SCRATCH is already set, it stays set; but if the data of the individual member has the TOUCH-TONE bit set and the SCRATCH bit is not set, then the bit in SCRATCH is also set. The DATAPR subroutine then continues to process the data words.

28.23 For the 3E3 generic, if the ENTRY_ID is 12, the data is miscellaneous bits which are put into word 11 of SCRATCH.

G. Directed Scan Point Number (DIR_SPN) Subroutine

28.24 The DIR_SPN subroutine obtains the directed SPN of TEN. This is a second entry point of the GET_DSPN subroutine. The entry condition is the TEN in register 1. The exit conditions are error, supervisory SPN returned, or directed SPN returned. These exit conditions are the same for the GET_DSPN subroutine. For the error return, register 0 and 1 are zero. For the supervisory SPN, return register 0 is zero, register 1 contains the supervisory SPN in bits 0 through 12, and bit 15 is set to one. For the directed SPN, return register 0 is one, register 1 contains the directed SPN in bits 0 through 12, and bit 15 is set to zero.

28.25 The DIR_SPN subroutine translates the TEN into the 2-word scanner entry by calling the TENXLA subroutine. The entry type is tested for the nonlinear entry. If the entry is not nonlinear, then the error return is returned. If the entry is a nonlinear entry, the test for supervisory scan point is made. If the entry contains a directed scan point, the directed SPN return is set up and returned.

H. Get Directed Scan Point Number (GET_DSPN) Subroutine

28.26 The GET_DSPN subroutine obtains the directed SPN of a circuit supervisory SPN. The entry condition is the supervisory SPN in register 1. The exit conditions are error return, supervisory SPN return, and directed SPN return. For the error return, registers 0 and 1 are zero.

For the supervisory SPN return, register 0 is zero and register 1 contains the supervisory SPN in bits 0 through 12 and bit 15 is set to one. For the directed SPN return, register 0 is one and register 1 contains the directed SPN in bits 0 through 12 and bit 15 is set to zero.

28.27 The 2-word entry associated with the TEN is obtained because the second word of the entry contains whatever is stored in register 1.

28.28 With the SPN, GET_DSPN first calls the TK_TEN subroutine. The TK_TEN subroutine translates the SPN into a TEN. If the scan point is not translatable, the error return is returned. Next the TENXLA subroutine is called to obtain the 2-word entry of the TEN. The entry identification is checked for a nonlinear entry. If the entry identification is not nonlinear, the error return is returned. Then the second word of the entry is obtained and put in register 1. This word contains either a supervisory or directed SPN. The high bit of the word is set indicating a supervisory SPN. Therefore, the return code of one in register 0 is returned. If the high bit is not set, there is a directed SPN and the return code of zero in register 0 is returned to the calling program.

I. Get Supervisory Scan Point Number (GET_SSPN) Subroutine

28.29 The GET_SSPN subroutine obtains the supervisory SPN of a circuit nonlinear TEN. The entry condition is the TEN in register 1. The exit conditions are error return of zero in register 0 and normal return of one in register 0 with the supervisory SPN in register 1.

28.30 The GET_SSPN subroutine calls the DIR_SPN subroutine with the SPN as the entry condition. Upon the return from DIR_SPN, the check is made for the error return. If there is an error, then it is passed on to the calling program. Next, the returned data in register 1 is checked for a supervisory SPN. If the supervisory SPN has already been obtained through DIR_SPN, the GET_SSPN subroutine returns to the calling program with a return code of one in register 0 and the SPN in register 1.

28.31 If the directed SPN was returned by the DIR_SPN subroutine, the directed SPN has to be translated into the supervisory SPN. The

SPNXLA subroutine is called with the directed SPN and the 2-word entry is obtained. There are four return codes from the SPNXLA subroutine. These return codes are:

- Zero—error
- One—miscellaneous subtranslator
- Two—universal subtranslator
- Three—line subtranslator.

The only valid return code (at this point) is the universal subtranslator return code. A check is made for a two. If it is not a two, then the error return code is returned; otherwise, the entry type is tested next. If the entry type indicates the two words are a directed SPN entry (the identification is four), then the first word of the entry contains the supervisory SPN. If the identification is not four, the error return code is returned.

28.32 When the SPNXLA subroutine returned to GET_SSPN, the first word of the entry was returned in register 1; therefore, the supervisory SPN is in register 1. The return code of one is returned in register 0 for the normal return.

J. Get Telephone Number (GET_TN) Subroutine

28.33 The GET_TN subroutine obtains the 7-digit telephone number associated with a TEN. The entry conditions are the TEN in register 1 and the party indicated in register 0. The party-indicated number indicates that a particular party of a multiparty line is wanted. If the party indicated is a zero, the telephone number of the lowest party number assigned is returned. For individual lines and multiline hunt groups (MLHG), the only numbers allowed are zero and one. For 2-party lines, the numbers allowed are from zero through two. For 4- and 8-party lines, the numbers allowed are zero through four and zero through eight, respectively. Also for the autoconnect line, which is an individual line but with party indication, the numbers allowed are from zero through six.

28.34 The exit conditions are return codes in register 0 and the telephone number in registers 1 and 2. The office code is in register 1 and the station number is in register 2. The digits are returned in binary coded decimal (BCD)

with BCD 0000 converted to BCD 1010. The return codes are:

- Zero—error
- One—multiline hunt group
- Two—8-party line
- Three—4-party line
- Four—2-party line
- Five—1-party line for a normal line
- Six—1-party line for an autoconnect line.

28.35 When there is an error return, register 1 indicates the type of error. The error types are:

- Zero—invalid TEN
- One—unassigned TEN
- Two—the party indicated is unassigned or out of range.

28.36 The GET_TN subroutine saves the TEN for later use. Also saved is the party indicated in the PTY_INDX register. If the party indicated is a two, the tip party data of a 2-party line may be wanted. The tip bit in register 1 for entry into LNDATA is set. If the line is not a 2-party line, the tip bit will not be checked by LNDATA. Next, LNDATA is called and LNDATA translates the TEN into the data for a line. The data associated with the line is stored in SCRATCH (Fig. 35 for Issue 4 of the SO-2 generic and Fig. 36 for the 3E3 generic). The 2-party and PBX indicators in SCRATCH are obtained and put into the PRI_INDC_R register for later testing. The return codes from LNDATA are tested next. Zero implies an assigned line and data stored in SCRATCH. One implies an unassigned TEN. Two implies that there was an error in translating the TEN, such as an invalid or out-of-range number. The error return code is returned with register 1 equaling zero.

28.37 When LNDATA returns an assigned line GET_TN, then it has to find the type of party line it is. Word 11 of SCRATCH contains the 4- and 8-party indicators which are then put

into the INDC_R register. Next the 4-party and 8-party bits are tested. If the TEN is for a 4-party line, the table number and entry number for the expansion entry is in word 11 of SCRATCH and is put in register 1. The PTY_INDX is checked to see if it is out of range. This could be any number greater than four. If the PTY_INDX is greater than four, the error return is returned with a two in register 1. If the PTY_INDX is within range, it is tested for zero. If it is zero, then any of the four lines that are assigned can be returned.

28.38 If the PTY_INDX is not zero, then one is subtracted from it. The modified PTY_INDX is used as the index into the expansion block. The table and entry number is in register 1, which was obtained through LNDATA. These numbers are used to obtain the 4-word expansion block associated with the TEN. Every 4-party line has a 4-word expansion block which indicates if the parties are assigned and which contains the packed directory number for each assigned party. With the table and entry numbers in register 1, the EXPAND subroutine in program XLSUB is called. Upon the return from EXPAND, GET_TN checks for the error return. If there is one, GET_TN returns error with register 1 containing a zero. With no error from EXPAND, registers 14 and 15 contain the address of the expansion block. The PTY_INDX is indexed into the block to obtain the party-indicated packed directory number. It is put in register 0. The party-indicated packed directory number is then tested to see if the telephone number exists or if the party is unassigned. If the party is unassigned, the error return code is returned with register 1 containing a two. If the party is assigned, the packed telephone number is to be unpacked.

28.39 Calling the UPK_DN subroutine is a common code used by all types of party lines. With the number in register 0, the UPK_DN subroutine is called. If UPK_DN returns an error, it is passed on with register 1 containing a zero. If there is no error, register 1 contains the office code in BCD in bits 0 through 12 for the unpacked number and register 2 contains the station number in BCD. Any BCD zeros (0000) have been converted to BCD tens (1010) as they would be found if dialed. Register 2 is then saved. The party bits in INDC_R and PRI_INDC_R (4 and 8, and 2 and PBX, respectively) and the originating major class for autoconnect are tested for the proper return code.

28.40 If the PTY_INDX is zero, then EXPAND is called and a search of the 4-word expansion entry is made to find an assigned line. If no assigned line is found, the error return code is returned with register 1 containing a one. If an assigned line is found, the packed directory number is obtained and GET_TN continues as described for unpacking the number.

28.41 If the TEN is for an 8-party line, then the table number and entry number for the expansion entry for the first four parties are in word 11 of SCRATCH and the table number and entry number for the expansion entry for the last four parties are in word 12 of SCRATCH. The PTY_INDX is tested. If the PTY_INDX is between five and eight, word 12 of SCRATCH is obtained and put into register 1 and five is subtracted from PTY_INDX for the index in the 4-word expansion entry. The GET_TN subroutine then continues as described for the 4-party expansion entry. If the PTY_INDX is between one and four, word 11 is obtained from SCRATCH containing the table and entry number. From this point, GET_TN continues the same as a 4-party line. The point at which the 8-party line differs from the 4-party line is when any assigned directory number is wanted. When the PTY_INDX is zero, EXPAND is called for 4-party line. Then the search for an assigned directory number is made using the table for word 11 and entry number if no assigned entry was found. Then the search is made using the table for word 12 and entry numbers. If there are no assigned lines, the error return is returned and register 1 contains a two.

28.42 If the TEN is not associated with a 4- or 8-party line, the packed directory number is obtained from word 11 of SCRATCH and put into register 0. The test for 2-party is then made. The 2-party bit in PRI_INDC_R is tested. If it is 2-party, the PTY_INDX is tested for an out-of-range party index. If the index is greater than two, the error return code is returned with a two in register 1.

28.43 Next there is a check to see if a directory number was obtained. If there is a packed directory number, the directory number is unpacked by calling the UPK_DN subroutine and the proper return code is returned. If the index is zero, the PRI_INDC_R is tested to see if either party is unassigned or denied service. If both parties are unassigned or either is unassigned and the other

denied service, the error return code is returned with a two in register 1. But if one of the parties is assigned, the next test is on the PTY_INDX. If PTY_INDX is not zero, then a particular party was wanted and it is unassigned, so the error return code is returned with a two in register 1.

28.44 The data of the other party is obtained next. The TEN is loaded into register 1 and the tip bit in register 1 is tested. If the ring party data has been obtained, then the tip party data is to be obtained so the tip bit in register 1 is set. If the tip party data has been obtained, then the ring party data is to be obtained so the tip bit in register 1 is zeroed. The LNDATA subroutine is called after the TEN is set up. If LNDATA returns an error, GET_TN returns the error return code with a two in register 1. If there is no error, the packed directory number is obtained from word 11 of SCRATCH and put into register 0. The register is then tested for zero. If it is zero, there is no directory number and the error return code is returned with a one in register 1. If there is a directory number, then it is to be unpacked as described before and the proper return code is returned.

28.45 If the line is not 2-party, the originating major class is tested for autoconnect. The autoconnect line is an individual line with the capability to have more than one directory number associated with it. The autoconnect line can have up to six directory numbers and one billing number. If the line is autoconnect, then the PTY_INDX is tested to see if a particular party is wanted. If a particular party is wanted, the PTY_INDX is range-checked for one or two. If it is either number, then the directory number for the assigned party is found in words 13 and 14 of SCRATCH, respectively. If a directory number is not assigned to a party number, the party unassigned returned code is returned. If there is a directory number, it is unpacked like any other directory number. If the party number in PTY_INDX is between three and six, word 15 of SCRATCH is processed by calling the subroutine EXPAND because word 15 contains a table and entry number pointing to a 4-word expansion entry. If there is a directory number and after indexing into and obtaining the directory number, it is unpacked.

28.46 If any directory number of a party is wanted, words 13 and 14 of SCRATCH are tested first. After testing words 13 and 14 of

SCRATCH, EXPAND is called and those four words tested for an assigned directory number. Again if no directory number is assigned, the party error return code is returned. If an assigned number is found, it is unpacked.

28.47 The two types of lines left are individual normal line and PBX/multiline hunt group (PBX/MLHG). The PTY_INDX should be either zero or one, so the test is made to check PTY_INDX. If it is greater than one, the error return code is returned with register 1 containing a two. If the PTY_INDX is within range, register 0 is checked for zero. Register 0 should contain the packed directory number from before. Therefore, if register 0 is zero, the error return code is returned with a one in register 1. If register 0 contains the directory number, then it is unpacked by calling the UPK_DN subroutine as previously described. The test for the return code is made. The PBX/MLHG bit is in PRI_INDC_R. If it is set, the proper return code is returned. If the PBX/MLHG bit is not set and the other party bits (8, 4, and 2) are not set, the line is an individual line and the test for the autoconnect major class is made and the proper return code is returned.

28.48 When LNDATA returns an unassigned line, GET_TN then has to find out if a particular party is wanted. First the 2-party bit is tested. If the line is 2-party, the PTY_INDX is tested. If the index is not zero, a particular party is wanted which is unassigned; so the error return code is returned with a two in register 1. If the index is zero, the GET_TN continues at the point at which an assigned line continues. If the line is not 2-party, the originating major class is obtained from SCRATCH and tested for zero. Zero implies unassigned. If the originating major class is zero, the error return code is returned with a one in register 1. If the originating major class is not zero, the PTY_INDX is tested for zero. If the PTY_INDX is not zero (a particular party wanted), then the error return code is returned with a two in register 1. If the PTY_INDX is zero, GET_TN continues at the point which an assigned line continues.

K. Get Billing Number (GETBIL) Subroutine

28.49 The GETBIL subroutine is called to obtain the billing number of a particular line TEN. The billing number may or may not be the same as the directory number. The entry condition is

SECTION 233-151-150

the line TEN in register 1 with bit 13 set if tip party data is being requested. The exit condition is a zero in register 0 for the error return code. The exit condition for a successful return code in register 0 is the office code in BCD of the billing number with bits 12, 14, and 15 indicating service observing, operator identify, and hotel-motel line, respectively. Also for a successful return, register 1 contains the last four digits (station number) of the billing number in BCD.

28.50 The GETBIL subroutine calls the LNDATA subroutine with the TEN in register 1. Upon the return from LNDATA, GETBIL checks for a successful return code from LNDATA. If LNDATA was not successful, then the error return code is returned by GETBIL to the calling program. If LNDATA was successful in obtaining the data of the line, the data is in SCRATCH. The GETBIL subroutine then obtains the word containing the billing indicators HOTEL operator identified (OI) and service observing (SO) from word 11 of SCRATCH. The OI bit is tested. If the line is OI, then there is no billing number; so if the OI bit is set, the three indicators are loaded into register 0 and GETBIL returns to the calling program. If the OI bit is not set, the packed billing number is obtained from word 12 of SCRATCH and loaded into register 0. Then the UPK_DN subroutine is called. The GETBIL subroutine returns the error return code to the calling program if UPK_DN was not able to unpack the billing number. If UPK_DN was successful, then register 0 contains the office code and register 1 contains the station number. All numbers are in BCD. Any BCD zeros were converted to BCD tens. Then GETBIL inserts the billing indicator into the high four bits of register 0 and returns to the calling program.

L. Get Group and Member Numbers (GETGRP) Subroutine

28.51 The GETGRP subroutine is called to obtain the group number and member number associated with a SPN. For a trunk circuit or a regular service circuit, the SPN is supervisory, which is found in the universal subtranslator. For a tone circuit or announcement circuit, the SPN is a line which is found in the line subtranslator. The entry condition is the SPN, which for a regular service circuit is the service circuit number. The exit conditions are a zero for error return code and a one for a successful return code in register

0. In register 1 for a successful return, the group number is in bits 0 through 7 and the member number is in bits 8 through 14. Also a successful return code in register 0 is bit 15 set and bits 0 and 1 contain the port number for the 3-port conference circuit. Register 1 would also contain the group number and member number.

28.52 The GETGRP subroutine calls the SPNXLA subroutine with the scan point number in register 1 as it is on entry into GETGRP. The return codes from SPNXLA are error, miscellaneous subtranslator, universal subtranslator, and line subtranslator. The first two (zero and one) are error conditions, so GETGRP returns error to the calling program. Two is returned for trunk and service circuits. Also the first word of the entry is in register 1.

28.53 For trunk and service circuits, GETGRP checks the ID of the 2-word entry of the universal subtranslator. If the ID in the first word is a one or a two (1-port supervisory entry or 2-port supervisory entry), then the second word of the entry contains the group number and member number which are obtained and loaded into register 2. Then GETGRP returns the success return code. If the ID is not a one or two, the error return code is returned.

28.54 For tone and announcement circuits, GETGRP checks the ID of the word entry of the line subtranslator. If the ID in the first word is an 11, then the second word of the entry contains a group number and a member number. In this case, they are obtained and loaded into register 1. The GETGRP subroutine then returns the success return code. If the ID is not 11, the error return code is returned. When the ID is 11, there will be a port number in the first word in bits 0 and 1. The two bits are put in register 0 before the second word is obtained. After the group number is obtained, it is tested for a 3-port circuit. If the group number is not for a 3-port circuit, register 0 returns a one. But if it is for a 3-port circuit, then bit 15 in register 0 is set to indicate the presence of a port number in bits 0 and 1. Then GETGRP returns register 0 the same as for success.

M. Line Data (LNDATA) and Terminal Equipment Number Data (TENDATA) Subroutines

28.55 Given the line TEN, all the data that is associated with a line TEN, including the

PBX/MLHG data and expansion data, is put into the base level SCRATCH area (Fig. 35 for Issue 4 of the SO-2 generic and Fig. 36 for the 3E3 generic) by the line data (LNDATA) subroutine. Ring party data for 2-party lines is retrieved unless tip party data is specifically requested. If the tip party is unassigned but not specified and the ring party is unassigned, then the return code is unassigned line. To alleviate this problem, the entry point TENDATA in LNDATA is used.

28.56 The terminal equipment number data (TENDATA) subroutine determines whether a TEN belongs to a line, an unassigned terminal, or neither. It is the same as the LNDATA subroutine except the tip party data is returned for 2-party lines in which the ring party is unassigned; otherwise, the ring party data is returned. The TENDATA subroutine should not be used for normal call processing. It is used to process unknown TENS or when sequencing through TENS.

28.57 The following subroutines are called by LNDATA and TENDATA subroutines: CLR_16A, DATAPR, EXPAND, PBXDAT, and TENXLA.

N. Local Numbering Plan Area (LOCALNPA) Subroutine

28.58 For generic 3E3 the LOCALNPA subroutine obtains the numbering plan area (NPA) for the office code given and returns the NPA and the AMA NPA indicator, which is the local NPA index in BCD.

28.59 For generic SO-2, the LOCALNPA subroutine obtains the NPA for the office code given and then obtains the other NPA. The LOCALNPA subroutine then compares the two NPAs. If the first NPA obtained is numerically less than the other NPA obtained, the AMA NPA indicator is zeroed. If the first NPA obtained is not numerically less than the other NPA obtained, the AMA NPA is set to one.

28.60 Subroutines called by LOCALNPA are ZEROTEN and BINXBCD.

O. Message Register Charge (MR_CH) Subroutine

28.61 The MR_CH subroutine increments the software message register by the message units and/or passes on the message units for the

hardware message register. For the 3E3 generic the message registers are recycled at 9999 (decimal).

28.62 The LNDATA and GNTRY3 subroutines are called by the MR_CH subroutine.

P. Normalized Office Code to Binary Coded Decimal (NOC_BCD) Subroutine

28.63 Given the normalized office code, the normalized office code to binary coded decimal (NOC_BCD) subroutine returns the office code in BCD and the local NPA index in Issue 4 of the SO-2 generic is one bit and in the 3E3 generic, two bits.

Q. Originating Line Translation (ORGXLA) Subroutine

28.64 Given the TEN of a line, the originating line translation (ORGXLA) subroutine returns the originating translation data of the line. The data returned by the LNDATA subroutine in SCRATCH is still intact.

28.65 When ORGXLA is called for 2-party data and if ring data is wanted and it is assigned, then ring data is obtained. If ring data is wanted and it is unassigned, tip data is obtained. If tip data is wanted and it is assigned, tip data is obtained. If tip data is wanted and it is unassigned, then unassigned return code is returned.

28.66 Subroutines called by the ORGXLA subroutine are LNDATA, PBX_BUSY, EXPAND, and DIG_ST.

R. PBX Data (PBXDAT) Subroutine

28.67 Given the group and member number of PBX/MLHG, the PBXDAT subroutine stores the PBX group and member list data in SCRATCH. This data is not modified by the line SPN entry data for the PBX/MLHG member.

28.68 The following subroutines are called by the PBXDAT subroutine: DATAPR, GRPX, and MBR_LIST.

S. Speed Call Expansion (SC_EXP) Subroutine

28.69 Given the speed call code, the SC_EXP subroutine returns the first word and address of the speed call list entry.

SECTION 233-151-150

28.70 The GENTRY3 subroutine is called by the SC_EXP subroutine.

T. Speed Call (SPDCL) Subroutine

28.71 The SPDCL subroutine, given the TEN and dialed code, converts the dialed code binary and returns the first word and address of the speed call list entry.

28.72 The following subroutines are called by the SPDCL subroutine: BCDXBIN, LNDATA, and SC_EXP.

U. Scan Point Number Translation (SPNXLA) Subroutine

28.73 Given a scan point number, the SPNXLA subroutine returns the address and the first word of the associated entry in the SPN translator. The SPN is used as two indexes (bits 12 through 6 and bits 5 through 0). The head table is a 1-word entry.

28.74 The GENTRY3 subroutine is called by the SPNXLA subroutine.

V. Service Circuit Number Conversion to Terminal Equipment Number (SV_TEN) Subroutine

28.75 The SV_TEN subroutine, given the service circuit number, returns the valid TEN if there is one for the service circuit number. If the service circuit is a 2-port circuit, the proper TEN is returned.

W. Terminal Equipment Number Translation (TENXLA) Subroutine

28.76 Given a TEN, the TENXLA subroutine converts the TEN to a line SPN and returns the address and the first word of the line subtranslators SPN entry.

28.77 The SPNXLA subroutine is called by the TENXLA subroutine.

X. Trunk or Service Circuit Scan Point Number Conversion to Terminal Equipment Number (TK_TEN) Subroutine

28.78 The TK_TEN subroutine, given the supervisory SPN for a trunk or service circuit, returns the TEN of the circuit. If the line SPN is received for a tone or an announcement circuit, the line

SPN is converted to its TEN. The 2-word entry of the trunk circuit associated with its scan point number is located in the universal subtranslator.

28.79 The SPNXLA subroutine is called by the TK_TEN subroutine.

Y. Tone Present Scan Point Number Conversion to First Tone Scan Point Number (TN_SPN) Subroutine

28.80 The TN_SPN subroutine, given the tone present SPN for a multifrequency (MF) or TOUCH-TONE receiver, returns the first tone SPN.

Z. Unpack Directory Number (UPK_DN) Subroutine

28.81 Given a packed billing or directory number, the UPK_DN subroutine returns a 7-digit (BCD) billing or directory number. The packed number may be a special billing index, in which case the 7-digit (BCD) outwats number is returned.

28.82 The following subroutines called by the UPK_DN subroutine are: BINXBCD, ZEROTEN, NOC_BCD, and GENTRY3.

29. COMMON TRANSLATION SUBROUTINES (XLSUB) PROGRAM

29.01 The common translation subroutines (XLSUB) program contains subroutines CLR_16A, GENTRY, GENTRY3, and EXPAND which are common to the translation programs. The XLSUB program also contains the subroutines LINESTATUS, FRMCHK, NXX_NOC, PCUTMAP, PRECUTST, and ANN_SPN which do not fit into a major classification of translations. For the 3E3 generic, XLSUB contains the EXPAND_C subroutine which is common to the translation programs. For the 3E3 generic, XLSUB also contains subroutines AUTO_CB, CBSERCH0, CBSERCH1, MLHG_SUB, RSCBS, TEN_TERM, and TN_UPTN which do not fit into a major classification of translations.

29.02 The following paragraphs briefly describe the subroutines contained in the XLSUB program.

A. Get Entry (GENTRY, GENTRY3) Subroutines

29.03 The get entry (GENTRY, GENTRY3) subroutines are really one subroutine with two entry points, GENTRY and GENTRY3. This subroutine

processes the 3-word master table index (MTI) entries. An entry describes a particular translation table, locates the table, and discloses the number of entries in the table. Given the particulars of the table, this subroutine locates the wanted entry in the table.

29.04 The entry conditions for GENTRY3 are the first word of the MTI entry in register 0, the translation table index or the search table match number in register 1, and the address of the MTI entry in registers 14 and 15. The GENTRY subroutine may be used when there is a table of 3-word MTI entries and when a particular entry is wanted. The entry conditions for GENTRY are the MTI table index in register 0 to select the particular entry, the translation table index or search table match number in register 1, and the address of the MTI table entry in registers 14 and 15.

29.05 The GENTRY subroutine first multiplies the MTI table index by three, then indexes the number into the MTI table and obtains the first word of the proper MTI entry. At this point, GENTRY and GENTRY3 become the code.

29.06 The subroutine then checks the XSLATOR bit which indicates if there is a translator associated with this entry or if the table does not exist, then an error condition exists and a return code of 0 is returned in register 0. If the translator does exist, the TYPE field is obtained next and saved as a return code. The TYPE field is usually one, but for the MTI table group table (GRPTBL), the type is either one for PBX/MLHG, two for service circuit group, or three for trunk circuit group. Then the address of the translator is obtained (the high 4 bits of the address are in the low 4 bits of the second word and the low 16 bits of the address are in the third word) and stored in registers 14 and 15.

29.07 The S_T bit is checked next to ascertain if the translator is a translation table or a search table. If it is a search table, then register 1 contains a match number. The first word of every entry in the table is compared to the match number until a match is found. If a match is not found, then the last entry is the entry wanted and the return code is two.

29.08 The entry size indicator (ESI) field indicates the number of words per entry. The

ENTRIES field indicates the number of entries to be searched. At this point a loop is set up for the search. If the compare is not equal, the address register is incremented by the number of words per entry and the first word of the next entry is compared. If there is a match or when there is a match before the last entry, the first word of the entry is returned via register 1 if there is only one word per entry; or the second word of the entry is returned if it is a 2-word entry. The address of the entry is returned in registers 14 and 15 with the return code in register 0. If the search is not successful, then the last entry in the table is returned with the return code of two returned disregarding the TYPE field which is one.

29.09 If the entry is a translation table, then register 1 contains the index. The index is compared to the number of entries in the table to check if it is within range. If it is not within range, an error code is returned. If it is within range, the ESI field is obtained which gives the LOG2 of the number of words per entry, or the LOG2 of the number of entries per word, or the number of words per entry minus two. These numbers make it easier to select the proper first word of an entry. Next the ROTATE bit is tested to see if there is more than one entry per word. If there is not more than one entry per word, then the multiply (MULT) bit is tested to see if the ESI contains the LOG2 of the number of words per entry or the number of words per entry minus two. If the MULT bit is not set, the ESI contains the LOG2 of the number of words per entry. The index is multiplied by the number of words per entry (index is rotated left by the ESI to obtain the proper word where the index number entry starts). With the index, the first word of the entry is obtained and the subroutine returns the return code and the address of the entry. If the MULT bit is set, the ESI contains the number of words per entry minus two. Basically, the index is multiplied by the number of words per entry and with the index, the first word of the entry is obtained. The subroutine returns the proper return code and the address of the entry.

29.10 If the ROTATE bit is set, then there are possibly 2, 4, 8, or 16 entries per word. The ESI contains the LOG2 of the number of entries per word. The subroutine then divides the index into two numbers. The first number is the word in the table that contains the entry and

SECTION 233-151-150

the second number is the position in the word where the entry starts. The particular entry is then obtained.

29.11 The temporary store (TS) bit is tested next.

If the table is a temporary store translator, the word that contains the entry is returned as it was obtained and the return code and the address of the entry is returned. If the translator is not a temporary store, the JUSTIFY bit is tested next. If the translation store entry is not to be right justified, the word that contains the entry is returned as it was obtained and the return code is returned along with the address of the entry. If the entry is to be right justified, the ESI field decides the number of entries in the word and the length of the entry. Then the entry is right justified, all other bits are zeroed, and the entry is returned. The return code and the address of the entry are also returned.

B. Expand (EXPAND EXPAND_C) Subroutine

29.12 The EXPAND subroutine is called to process 1-, 2-, or 4-word entries for Issue 4 of the SO-2 generic and 1-, 2-, 4-, or 8-word entries for the 3E3 generic which contain data associated with the SPN line subtranslator, or the 4-digit hundreds group table, or the route index expansion table. When an entry in one of these translators contains insufficient words for the optional data, one of the words is used to point to the additional data words. The pointer contains the table number and entry number used in the general purpose expansion translation table (EXPTBL). The translator has a head table which is indexed by the table number. Then the entry number indexes into the individual tables.

29.13 The EXPAND subroutine is entered with the table number and entry number. The entry number is saved and the table number is isolated in register 1 for entry into the GNTRY3 subroutine. The first word and address of the 3-word MTI EXPTBL is obtained for entry into GNTRY3. The GNTRY3 subroutine is called and returns (if without error) the address and the first word of a 2-word entry. The entry contains the store increment to the table, the number of entries in the table, and the size of the entries in the table which is 1, 2, or 4 for Issue 4 of the SO-2 generic and 1, 2, 4, or 8 for the 3E3 generic. The size (ESI) is isolated for later use.

29.14 The entry number is compared to the number of entries to see if it is within range. If not within range, a branch is made to an error return. If within range, then multiply the ESI by the entry number, obtain the store increment, and update the address. Then add the increment to the multiplied entry number and index this number from the address obtaining the first word in register 1 and the address in registers 14 and 15 of the desired entry. The subroutine then puts a number in register 0 as the return code to indicate the number of words per entry (a 1 equals one word, a 2 equals two words, a 3 equals four words and a 4 equals eight words).

29.15 The subroutine EXPAND_C is only used in the 3E3 generic. Subroutine EXPAND_C is called to process 2- and 4-word entries which contain data associated with the call forwarding and speed call translators. The data is BCD digits. Both call forwarding and speed call allow 8 digits to be stored in their translators. The speed call translators are SC1LST and SC2LST. The call forwarding translator is CFLST. If more than 8 digits are needed to be stored, the digits are stored in the code list table.

29.16 Subroutine EXPAND_C is entered the same as EXPAND, except that the 3-word MTI is CODLST instead of EXPTBL. The rest of the subroutine is the same.

C. Frame Check (FRMCHK) Subroutine

29.17 The FRMCHK subroutine tests the maintenance frame equipped (MT_FEM) MTI word in the TDATA program. Each bit in MT_FEM is associated with a network frame, except for bit 0. Bit 0 is always set because there is always the master scanner. The concentrator group number in the TEN indicates which bit is tested to see if a frame is equipped or the scanner number in the SPN indicates which bit. Therefore, the entry condition into FRMCHK is the TEN or SPN in register 1. The exit conditions are the TEN or SPN in register 1 unchanged and the return code in register 0. A return code of zero implies an unequipped frame and a one implies an equipped frame.

29.18 The FRMCHK subroutine first obtains the MT_FEM word and puts it into register 0. The FRMCHK subroutine then right justifies the concentrator number in the TEN in register

1. Next the check on the bit indicated in register 1 is made on MT_FEM. The TEN or SPN in register 1 is rotated back so that it is unchanged. Then, if the bit is set, a return code of one is returned. If the bit is 0, a return code of zero is returned.

D. Clear 16 Words (CLR_16A) Subroutine

29.19 The CLR_16A subroutine zeroes out a 16-word block of data. Registers 12 and 13 contain the address of the data block to be cleared. The CLR_16A subroutine exchanges the address registers with the other pair of address registers, then calls CLR_16 which requires the address to be in registers 14 and 15. The CLR_16A subroutine is a common system subroutine. It returns with the data block cleared and then CLR_16A returns to the calling routine with no return code.

E. Line Status Bits Test and Manipulation (LINESTATUS) Subroutine

29.20 The LINESTATUS subroutine has seven entry points (LINEBUSY, LINEHD, LINEIDLE, LINEJB, LINEMB, LINEPS, LINETEST) which test and change the line scanner status bits (SCANSTAT). The LINEBUSY entry changes the status bits to normal busy if the status was idle. The LINEIDLE entry changes the status bits to idle. The LINEHD entry changes the status bits to the permanent signal state. The LINEJB entry changes the status bits to normal busy regardless of the previous state. The LINEMB entry changes the status bits to maintenance busy. The LINEPS entry changes the status bits to the permanent signal state and puts the TEN in the high and wet list. The LINETEST entry does not change the status bits but tests them and returns an indicator as to the present status. This is also true of the other entries; they indicate the previous status before changing. So the return code in register 0 gives the status at the time the subroutine was called. A zero is idle, one is the maintenance busy state, two is the normal busy state, and three is the permanent signal state.

29.21 After entry into one of the subroutines, register 1 is set to a 4-bit code. Bits 0 and 1 indicate the new value of the line status bits. Bit 2 indicates if the setting of the line status bits is not to be changed, as for LINETEST.

The code and the scan point number, which are in register 0 from entry, are saved.

29.22 The entry LINEIDLE then loads the SPN into register 1 and calls the subroutine PBX_IDLE in program EQPSEL. This subroutine will idle the selection status bit for the line SPN if it is for a PBX/MLHG member. The LINEIDLE entry then merges with the main routine.

29.23 The LINEPS entry first calls the STUFFHW subroutine in program SCANS. The LINEPS entry then merges with the other entries.

29.24 The entries LINEBUSY, LINEHD, LINEJB, and LINEMB are merged together before the saves, so now the SPN is loaded in register 1 for the call to PBX_BUSY subroutine in program EQPSEL. This subroutine will busy the selection status bits for the line SPN if it is for the PBX/MLHG member. These entries are then merged with the main routine.

29.25 After the saves, the entry LINETEST merges with the main routine. The first thing the main routine does is to load the SPN into register 0. The subroutine then isolates the logical scanner number from the SPN and tests for zero. If the logical scanner number is zero, the translation error buffer (SXLERROR) is loaded with the not-a-line SPN (NOT_L_SPN) error code and the SPN which was given on entry. The subroutine returns a return code of one. If the logical scanner number is not zero, the subroutine SCANDISP in program SCANS is called which converts the logical scanner number to an index for 48 words per table.

29.26 The address of SCANSTAT, which is the beginning of the line status bits, is obtained. The converted logical scanner number is added to the address giving the first word of the logical scanner status bits. The scanner row is then multiplied by two because of 2-word entries. Lines are rows 2 through 23. If the row number is not within range, then the XSLERROR is loaded with the NOT_L_SPN error code and the SPN which was given on entry. The subroutine returns a return code of one. The XSLERROR is a 2-word entry which is outputted on the TTY every 30 seconds when there is data to be outputted. When the data is outputted, the XSLERROR is then zeroed.

SECTION 233-151-150

29.27 If the row number is within range, then the multiplied row number is added to the address. Both words of the 2-word entry containing one bit per word per point, which gives one row per two words, are then obtained. The two bits for the particular point are tested and the results are saved for return later. The code is then tested to see if the status bits are to be altered. If the status bits are not altered, the results of the status bits are returned. If the status bits are altered, is the alteration conditional? If the alteration is conditional, then the return code is tested for zero which indicates that the line is idle. If the line is not idle, the return code is returned as is. If the line is idle, continue as if the alteration is unconditional by clearing (zeroing) both bits, then setting or not setting the bits as indicated by the code, storing the updated line status bits, and then returning the code as it was.

F. Announcement Scan Point Number (ANN SPN) Subroutine

29.28 The ANN_SPN subroutine is called to obtain the announcement SPN of one of the specified announcement circuit groups. The announcement SPNs are in a data table in the XLSUB program. The announcement SPN indicates whether or not the announcement is in use.

29.29 The ANN_SPN subroutine is entered with the circuit group number in register 1. The base announcement group number, which is the first of possibly eight consecutive group numbers, is obtained. The difference between the base announcement group number and the announcement group number entered with is obtained. Then the address of the announcement SPN table is obtained and the size of the table is also obtained. Next, the group number is checked to see if it is within range. If it is not within range, an error code of zero is obtained. If the group number is within range, the difference between group numbers is used to index into the table, the proper announcement SPN is obtained and loaded in register 1, and a return code of one is returned in register 0.

G. Autoconnect Callback Block (AUTO_CB) Subroutine

29.30 The AUTO_CB subroutine is used only in the 3E3 generic. Subroutine AUTO_CB is called to obtain the callback block entry. The subroutine is used by call processing, maintenance,

and data base. It returns the callback block entry and address.

29.31 Subroutine AUTO_CB is entered with the index in register 1. It obtains the first word and address of the callback block (CALLBACK) MTI in registers 0, 14, and 15. The registers are set up for the call in subroutine GNTRY3. Subroutine GNTRY3 returns the first word and address of the entry if there is no error. If an error occurred, the error-return code of 0 is returned. With a successful return from subroutine GNTRY3, the address of the entry is saved in registers 14 and 15 and returned to the calling program with the first word in register 1 and the return code of 1 in register 0.

H. Convert Office Code to Normalized Office Code (NXX_NOC) Subroutine

29.32 The NXX_NOC subroutine searches the normalized office code to office code conversion table (NOCBCD) until a match on the office code is found. The office code is entered in register 1 as the entry condition. Then any zeros in the code are converted to binary 1010. The table has eight possible entries. Each entry contains the 3-digit office code in BCD. So the loop is set up to find a match. When the match is found, the index that points to the entry is the NOC. If no match is found, an error condition of zero is returned. If a match is found, then register 0 returns a one and register 1 contains the NOC.

I. Precut Status (PRECUTST) Subroutine

29.33 The PRECUTST subroutine is called when an office is in the precutover state and a 4-digit translation is taking place. The line cutoff contacts of lines not yet in the cutover state should not be closed, so PRECUTST returns the map of the precutover lines. Because this subroutine is called for every 4-digit translation when the office is in the precutover state, it takes a shortcut to get the data needed. When the map is to be written into the subroutine, the PCUTMAP subroutine is to be called which takes no shortcuts to obtain a location of a TEN in the map.

29.34 The entry condition for PRECUTST is the line TEN in register 1. The shortcut that is taken is not to call the subroutine GNTRY3. Instead the 3-word MTI entry cut status (CUTSTAT) is obtained first by getting the first word and

testing the XSLATOR bit to see if the map exists, because the map does not have to exist when the office is in the cutover state. So if the bit is not set, there is no map. Then PRECUTST returns to the calling subroutine an error code of zero in register 0. But if the bit is set, the map exists. Then the index into the CUTSTAT table has to be created. This is done by isolating the concentrator group (logical scanner number) number, subtracting one from it, and multiplying it by 24. Then the switch number group, concentrator number, and the high two bits of the switch number (scanner row number) as one field are added to the modified concentrator group number. Next the address in CUTSTAT is obtained and the index is added to the address. The word in the table is then obtained and the address is updated. The word is returned in register 0. This word is the 16 points for the row indicated by the TEN. Also the address is returned in registers 14 and 15. The TEN is still in register 1 untouched, so the low bit of the switch number and the level number (scanner point) as one field may be used to obtain the proper bit from the word in the map returned.

J. Precut Map (PCUTMAP) Subroutine

29.35 When an office is in the precutover state and a line is being put into or taken out of the cutover state, the PCUTMAP subroutine is called. If the state of a cutoff is only to be tested, the subroutine PRECUTST is to be called.

29.36 The entry condition for PCUTMAP is the line TEN in register 1. The TEN is used like the SPN layout to obtain the desired entry. The concentrator group number (logical scanner number) is first isolated, then one is subtracted from it. The number is then multiplied by 24. The scanner row number (the switch group number, the concentrator, and the high two bits of the switch number) is then added to the modified concentrator group number to give the index into the CUTSTAT table. The first word of the 3-word MTI CUTSTAT entry is put into register 0 and address of the entry is obtained. These are the entry conditions for the call to GENTRY3 which, upon returning, is checked for an error. If no error occurred, the word from the table which contains the scanner point bit (low bit of switch number and level number) is in register 1. The address of the entry is in registers 14 and 15, which is put into the hold/get area for the return

from PCUTMAP. The return code of one is in register 1.

K. ROTL Security Callback Status (RSCBS) Subroutine

29.37 The RSCBS subroutine is used only in the 3E3 generic. Subroutine RSCBS is called to check on the validity of the remote office test line (ROTL) user and return the authorization class. The subroutine searches through the CALLBACK block translator for the function ROTL security and compares for the member number. When the member number is found, the callback entry is returned.

29.38 The entry condition for RSCBS is the same member number in register 1. The member number is saved for the compare later. The number of entries in the translator is obtained from the second word of the CALLBACK MTI entry and is used as the parameter for the search loop. The subroutine AUTO_CB is called with register 1 equal to zero to obtain the first entry of the CALLBACK block translator. If subroutine AUTO_CB returns an error, the error return code is returned. If AUTO_CB is successful it returns the first word of the entry in register 1 and the address in registers 14 and 15. The second word of the entry is obtained which contains the function field. This field is compared with ETYP_ROT_L_SCUR which is a 10. If the function does not match, the first word and address of the next entry is obtained, if there is another entry. If there are no more entries, the error return code is compared to the member number field in the first word. If they do not match, the subroutine continues as if the function did not match. If the member number does match, the first word and address of the entry is returned with a successful return code.

L. Telephone Number to Unpack Telephone Number (TN_UPTN) Subroutine

29.39 The TN_UPTN subroutine is only used in the 3E3 generic. Subroutine TN_UPTN is called to unpack a telephone number from the packed 15 bits to the station number and the normalized office code (NOC). Subroutine TN_UPTN is used by program TERM. The packed telephone number is obtained from the series completion entry in SCRATCH. It was stored there by the 4DIGIT subroutine. This subroutine is general purpose and can convert any packed telephone number.

SECTION 233-151-150

29.40 The entry condition for subroutine TN_UPTN is the packed telephone number in register 0. Subroutine TN_UPTN calls the UPK_DN subroutine which takes the packed telephone number in register 0 and unpacks it into the office code and station number of the packed telephone number. The office code and station number are returned to subroutine TN_UPTN in registers 0 and 1 respectively. Subroutine TN_UPTN saves the station number for return later. It loads the office code into register 1 which is the entry condition for the subroutine NXX_NOC. Subroutine NXX_NOC converts the office code into the NOC and returns it in register 1. Subroutine TN_UPTN loads the station number into register 0. With registers 0 and 1 containing the data indicated, subroutine TN_UPTN returns to the calling program. If either subroutine UPK_DN or NXX_NOC returns to subroutine TN_UPTN with an error, TN_UPTN would return an error code to the calling program.

M. Callback Search (CBSERCH0, CBSERCH1) Subroutine

29.41 The callback search subroutine has two entry points, CBSERCH0 and CBSERCH1. The entry conditions for CBSERCH0 are:

- (a) Function and message class
- (b) Function and member number
- (c) Function only.

The entry conditions for CBSERCH1 are TTY controller number and port number. Also, both subroutines may be reentered with an entry count in R1 and callback block entry address in RA1. The subroutine searches the callback block until it finds a match or until the table is exhausted. If a match is found, an entry count is returned in R1 and the address of the entry selected is returned in RA1.

29.42 The entry flag (R5) is set to 0 if CBSERCH0 is called and set to 1 if CBSERCH1 is called. If R1 contains a -1, AUTO_CB is called to obtain the first word of the callback block in R1 and the address of the callback block entry in RA1. R1 is loaded with the number of entries -1 from the second word of the callback block MTI. Otherwise, if R1 does not contain a -1, it is the second pass. The next callback block is obtained and the search continues.

29.43 The subroutine checks the CBB bit which indicates an active callback block. If the callback block is inactive, the next callback block is checked until an active bit is found.

29.44 The subroutine then determines if the entry was CBSERCH0 or CBSERCH1.

29.45 If the entry was CBSERCH1, the TTYC number is compared to the CBB TTY. If a match occurs, the port number and the CBB port number are compared. If these match, it returns successfully; if no match, it continues searching.

29.46 If the entry was CBSERCH0, a series of tests are performed. The CBB function is compared to one function to see if they are equal. If they are not equal, continue searching. If they are equal, the function number is checked. If the function number has a value of 0 through 7, the entry conditions are function and message class. Then the message class is compared to the CBB message class to see if they match. If they match, it returns successfully. If they do not match, it continues searching. If the function has a value of 8 through 15, the entry condition is either function only or function and member number. A member number with a value of -1 indicates a function only and returns successfully. Therefore, the member number is compared to the CBB member number to see if they are equal. If they match, it returns successfully. If they do not match, it continues searching.

29.47 The search continues until the information is found or until all entries in the callback block have been examined.

N. Multiline Hunt Group (MLHG_SUB) Subroutine

29.48 The MLHG_SUB subroutine is only used in generic 3E3. Subroutine MLHG_SUB is called to store the eight words of multiline hunt group (MLHG) data in the scratch area for MLH called MSCRATCH. Subroutine MLHG_SUB is called by programs TERM, VRGRP, and ORGRP. The purpose of this subroutine is to store contiguously all the group data that was originally in the SO-2 generic 8 words of MLHG data.

29.49 The entry condition for MLHG_SUB is the group number in register 1. The subroutine calls GRPX with the group number in register 1.

GRPX returns the first word of the group data. The address of MSCRATCH is obtained. The last word of the group data may contain an expansion entry. If it does, then the NIT and STP indicators from this word and the NITE_STOP and STOP_HUNT member numbers from the first word of the expansion entry are put in word 7 (last word) of MSCRATCH. The 7 words (0 through 6) of group data are stored in MSCRATCH 0 through 6. If the last word does not contain an expansion entry item, all 8 words are copied into MSCRATCH. The address of MSCRATCH is then returned.

O. Terminal Equipment Number to Terminating Translation (TEN_TERM) Subroutine

29.50 The TEN_TERM subroutine is called to translate the given TEN through the originating translator and obtain the telephone number and translate that to give the terminating translation data. This subroutine is used by call processing and can be used by any program to do this translation.

29.51 The TEN_TERM subroutine, given the TEN in R1 and a party indicator in R0, calls GET_TN which translates the TEN and returns the telephone number unpacked and in BCD format in registers 1 and 2. Subroutine TEN_TERM then calls NXX_NOC with the office code (NXX) in R1, which converts the NXX into a normalized office code (NOC) and returns the NOC in R1. TEN_TERM loads the NOC into R0 and loads the station number (the last 4 digits of the telephone number) in R1. TEN_TERM then calls TXLNPC with registers 0 and 1. TXLNPC returns to TEN_TERM with the terminating translation data. This data is in registers 0 and 1 and in SCRATCH. This data varies depending on the type of line. TEN_TERM then returns the data as it is returned from TXLNPC to the calling program.

30. EQUIPMENT SELECT (EQPSEL) PROGRAM

30.01 The equipment select (EQPSEL) program contains subroutines which:

- Obtain the group data for trunk circuits and for service circuits;
- Select circuit members;
- Test, busy, and idle the selection status bits of PBX/MLHG members;

- Select an idle TCR.

30.02 The following paragraphs briefly describe the subroutines contained in the EQPSEL program.

A. All Busy Check (ABCK) Subroutine

30.03 The subroutine ABCK is called to find out if all the circuits in a group are busy. This subroutine is normally called when there is a no-action situation involving a service circuit. A no-action situation is when a service circuit is waiting for a supervisory signal that is being delayed, such as a transmitter waiting for the signals to send. The calling program will leave this circuit in the busy state if there are other idle circuits; but if the ABCK subroutine indicates all circuits are busy, then the program will terminate the call using this circuit so that the circuit is available for the call being set up. If ABCK indicates idle circuits, the circuit will stay in the busy state until the signal arrives, or the timer times out, or ABCK returns all busy.

30.04 The entry condition for ABCK is the circuit group number in register 1 bits 0 through 7. The exit condition is the contents of register 0. Register 0 will contain a zero if there are errors in the translation such as an unassigned group, not a circuit group number, or no selection status block; a one if all the circuits are busy; or a two if there are idle circuits.

30.05 The STATUS_G subroutine is called by subroutine ABCK with the same entry condition by which subroutine ABCK is called. When STATUS_G returns to ABCK, the check for errors is made on register 0. If there are no errors, the first word of the selection status block for the group indicated is in register 1. In this first word is the all busy bit which is checked to determine if all the members in the group are busy. If the bit is not set, then there are idle circuits in the group. But if the bit is set, all the circuits in the group are busy. The ABCK subroutine returns a one or two based on this bit.

B. Busy Circuit (BUSYCKT) and Busy Service Circuit (BUSYSVC) Subroutines

30.06 The busy circuit (BUSYCKT) and busy service circuit (BUSYSVC) subroutines are really one subroutine with two entry points,

BUSYCKT and BUSYSVC. When a service circuit or a trunk circuit is to be busied, the subroutine BUSYCKT or BUSYSVC is called. The subroutine sets the selection status bit of the circuit to one. The entry condition for BUSYCKT is the supervisory SPN in register 1 for trunk circuits and regular service trunks and the line SPN in register 1 for tone circuits and announcement circuits. The supervisory scan point for regular service circuits is the same as the service circuit number. If the calling program knows it has the service circuit number, then the subroutine BUSYSVC is the subroutine to call. The only difference between BUSYSVC and BUSYCKT is that BUSYSVC zeroes out the high bits because service circuits can only be in scanner 0, rows 0 through 15.

30.07 The exit conditions from BUSYCKT and BUSYSVC are the same. They are: if register 0 equals zero, then an error was found; if register 0 equals one, the selection status bit of a circuit was set; and if register 0 equals two, the bit was already set.

30.08 The subroutines call the STATX subroutine via the IGNORE macro. With the SPN in register 1, the status of the scan list IGNORE and LOOKAT bits are changed for the circuit specified. The IGNORE bit is set and the LOOKAT bit is zeroed. The STATX subroutine returns the SPN in register 1. The SPN is then saved for later use. With the SPN in register 1, the GRPADR subroutine is called.

30.09 The GRPADR subroutine is called to obtain the group and member number of the circuit indicated. If the group and member numbers are not obtained GRPADR returns error and, in turn, BUSYCKT returns error. If GRPADR returns the trunk circuit return code, then the SPN is put in register 1 for the call to STATUS_S. If GRPADR returns the service circuit return code, the group number is tested for 3-port circuit group. If it is not 3-port circuit group, the TESTCKT subroutine continues as for trunk circuits. If the group number is for 3-port circuit, the member number is isolated in register 1 and the PORTBUSY subroutine is called. The PORTBUSY subroutine sets (busies) the 3-port status bits. If PORTBUSY returns an error, then BUSYCKT returns error to the calling program. If there is no error, the bits are set and the subroutine continues as for trunk circuits.

30.10 The subroutine STATUS_S is then called returning the address of the selection status block in registers 14 and 15 and member number of the scan point in register 1. If there was an error, the error code would be zero; for success, register 0 equals one.

30.11 The selection status block has four header words and there is also one bit per member for as many members as are allocated. There are 16 members per word. Therefore, the 7-bit member number is broken into two numbers. The low four bits indicate the member in the word and the circuit position. The high three bits indicate the word and the index. So the member number is split, obtaining both numbers. Four is then added to INDEX to account for the header words.

30.12 Now the INDEX is used to obtain the word containing the status bit for the wanted circuit. That bit is then tested to see if it is set. If it is set, the circuit is already busy and the return code of two is returned in register 0 to indicate that it is busy. If the bit is not set, the idle is set and that word is stored back into the temporary store word from which it came. Also at this time, the group peg counter in word one of the selection status block is incremented by one. The return code of one is then returned in register 0 to indicate that the circuit is to be used.

C. Circuit List (CKTLST) Subroutine

30.13 The subroutine CKTLST obtains the member list data and the TEN for regular service circuits and trunk circuits and obtains the member list data for tone circuits and announcement circuits. The subroutine is entered with the circuit member number and the address of the group data of the circuit. The member number is saved for future use. It also needs to be checked to see if it is within the range of the group. The largest member number (LMN) is obtained from the group data and the member number of the circuit is subtracted from the LMN. If the member is not within range, the error return code is returned to the calling program. The error return code for CKTLST is a one in register 0. If the member is within range, the circuit code number from the group data is saved in the CKTCDHLD register. Next the MBR_LIST subroutine is called.

30.14 The MBR_LIST subroutine is called with the group data address of the group in

registers 14 and 15. If the subroutine is successful, it returns the member list data address of the group in registers 14 and 15, the first word of the list in register 1, and a one in register 0. If the subroutine is unsuccessful, the error return code is returned. The first word of the member list is a header word which contains a format (FORMAT) field, the number of members allocated in the group, and the number of spare members in the list. Also in the member list is the data needed to identify the circuit, considering the type of circuit it is.

30.15 The format in the first word indicates the type of circuit the list contains. The format of the list is dictated by the type of circuit within. The different types are PBX/MLHG which is not to be accessed by this subroutine, regular service circuits, tone and announcement circuits, and trunk circuits. So the first word is saved, the member number is returned to register 1, and the size of the list is put into the SIZE register. The FORMAT is tested for announcement and tone circuits, trunk circuits, and regular service circuits.

30.16 If the FORMAT indicates that the group is the regular service circuit group, then the service circuit number (SPN), distributor triplet address, and circuit code number are in the member list for the circuit. The SPN is only an 8-bit number for regular service circuits, so the first half of the list contains all the SPNs for that group with two SPNs per word. The second half of the list contains the distributor triplet addresses and circuit code numbers for that group. Each word contains a distributor triplet address in the low 11 bits and a circuit code number in the high 5 bits. To obtain the distributor triplet address and circuit code number, the member list SIZE is made even and divided by two. The SIZE is added to the saved member number in the MEMHLD register, then two is added to the MEMHLD register. The MEMHLD register is used to index from the member list address to obtain the distributor triplet address (DISTR_TRIPLET). Then to obtain the SPN, the saved number in the WRK3 register is evened and divided by two. A one is added to WRK3. Then WRK3 is used to index from the member list address to obtain the word containing the SPN. If the member number is even, the low eight bits contain the SPN. If the member number is odd, the high eight bits contain the SPN. With the word in the SCAN_POINT register and if the member is odd, the SCAN_POINT is rotated to put

the number in the low eight bits. The SPN is then isolated. It is now tested for zero. If it is zero, a return code of 2 is returned. If it is not zero, the SPN is put in register 1 for the call to the TK_TEN subroutine in program XSLSUB. The TK_TEN subroutine returns the TEN in register 1 and a one in register 0 if the subroutine is successful. If TK_TEN was not successful, register 0 contains a zero to indicate an error. The CKTLST subroutine then returns an error return code to the calling program. If the TK_TEN subroutine is successful, the CKTLST subroutine continues.

30.17 Next the TEN is isolated and stored in word 0 of SCRATCH (Fig. 35 for Issue 4 of the SO-2 generic and Fig. 36 for the 3E3 generic). Then the distributor triplet address and circuit code number are stored in word 1 of SCRATCH and the SPN is again put in register 1, this time for the return 1. The subroutine then returns to the calling program with a zero in register 0 for the successful return.

30.18 If the format indicates that the group is the trunk circuit group, the member list contains a 13-bit SPN in the first word and an 11-bit distributor triplet address and a 5-bit circuit code number in the second word of a 2-word entry. To obtain the scan point words, the member number is multiplied by two and a one is added to it for the header word. The MEMBER is then used as the index for the member list address to obtain the SPN. The address is updated at this point. To obtain the distributor triplet address and the circuit code number, 1 is the index to the address for the next word. The subroutine then branches to where the test for the zero SPN is and continues like the regular service circuit.

30.19 If the format indicates that the group is the tone and announcement circuit group, the member list contains the 13-bit TEN only. So one is added to the MEMBER number and then MEMBER is used to index from the member list address to obtain the TEN. The TEN is then isolated and tested for zero. If it is zero, this implies a spare and therefore a return code of two is returned to the calling program implying a nonworking member. If it is not zero, the SCAN_POINT and DISTR_TRIPLET registers are zeroed. This implies that there is no SPN or distributor triplet address but there is a circuit code number. This is obtained from CKTCDHLD and inserted into DISTR_TRIPLET. So the

subroutine then branches to where the TEN is stored into SCRATCH and continues as described before.

D. Data Circuit (DATCKT) and Data Circuit Group (DATCKTG) Subroutines

30.20 The subroutines data circuit (DATCKT) and data circuit group (DATCKTG) are the same subroutine but with different entry points, DATCKT and DATCKTG. The subroutines DATCKT and DATCKTG are called when the data of a particular member of a particular group is wanted. The DATCKT subroutine is entered with a SPN (supervisory for trunk and regular service circuits and line for tone and announcement circuits) in register 1. The GETGRP subroutine in program XSLSPN is called. If the group and member numbers were unobtainable, DATCKT returns the error return code of zero in register 0 to the calling program. But if the GETGRP subroutine is successful, the group and member numbers are returned to DATCKT in register 1 and a one in register 0. Also in register 2 is the port number of a 3-port conference circuit. The port number is stored in word 3 of SCRATCH. It is at this point that DATCKTG joins DATCKT.

30.21 The member number is saved for future use. The group and member numbers are stored in word 2 of SCRATCH. With the group number in register 1, the subroutine GRPX is called. Upon its return, the address of the group data is saved in the hold/get area. An error return or a PBX/MLHG return will result in DATCKT returning to the calling program with an error. If the return indicates service circuit or trunk circuit, the subroutine modifies and saves the return code for the return.

30.22 Upon a successful return, the first word of the group data is returned. Next the EXIST bit is tested. If it is not set, an error is returned. If the EXIST bit is set, the group is defined. Then the member (MEM) bit is tested. If it is not set, an error is returned. If the MEM bit is set, the group has working members in it. Then the member number is isolated and put into register 1 for the call to CKTLST. If the CKTLST subroutine is unsuccessful (return codes 1 and 2), an error is returned. Otherwise, the saved return code from GRPX is put into register 0 and DATCKT returns to the calling program. The data that is

returned to the calling program is the same as the data returned by CKTLST.

E. Get Distributor Triplet Address (GET_DTA) Subroutine

30.23 The GET_DTA subroutine is called only in the processing of a by-link trunk circuit. This subroutine obtains the trunk circuits distributor triplet address from the member list without the use of SCRATCH. None of the subroutines called by this subroutine use SCRATCH. The GET_DTA subroutine is entered with the by-link trunk supervisory SPN in register 1. The subroutine GETGRP in program XSLSPN is called to translate the SPN in the group and member numbers. If there is an error, the error code is passed from GET_DTA to the calling program as a return code of zero in register 0. If GETGRP is successful, the member number is isolated from the group number and saved in register MEMBER for future use.

30.24 With the group number in register 1, the subroutine GRPX is called. Any return code other than three is an error and GET_DTA will return error to the calling program. With a three in register 0, the first word of the group data is in register 1 and the address of the group data is in registers 14 and 15. Given the address of group data, the subroutine MBR_LIST is called. If an error occurred in obtaining the member list data, GET_DTA returns the error return code. If the member list data is obtained successfully, the first word and address of the member list are in registers 1, 14, and 15, respectively.

30.25 In the first word of the member list is the format code which indicates the type of circuit in the list. The only possible type is a trunk circuit for GET_DTA. If the list is not for a trunk circuit, GET_DTA returns an error return code to the calling program. If the format does imply trunk circuit, the rest of the list is in 2-word entries. The first word contains the SPN and the second word contains the distributor triplet address. To obtain the second word, two is added to MEMBER, one for the header word and one for the first word. The MEMBER register is then indexed from the member list address to obtain the proper distributor triplet address which is placed in register 1. The GET_DTA subroutine returns a one in register 0 for the return code.

F. Group Address (GRPADR) Subroutine

30.26 The GRPADR subroutine obtains the group data of the group associated with the SPN. The supervisory scan number is the entry condition for the data of a regular service circuit group or a trunk group. The line SPN is the entry condition for the data of a tone circuit group or an announcement circuit group. With the SPN in register 1, the subroutine GETGRP in program XSLSPN is called. If the subroutine is not successful, an error is returned and GRPADR in turn returns the error return code of zero in register 0. If the subroutine is successful, then register 0 is one. Register 1 contains the group and member numbers which are saved for later use. Register 2 contains the port number of a 3-port conference circuit.

30.27 Subroutine GRPX is called next. The group number is isolated in register 1. The GRPX subroutine returns the type of circuit in register 0 as the return code, the first word of the group data in register 1, and the address of the data in registers 14 and 15. The GRPADR subroutine then puts the group and member numbers in register 1 and puts the port number and group address in the hold/get area for the return. The GRPADR subroutine then returns to the calling subroutine with the same return code that was received from the GRPX subroutine.

G. Group Translation (GRPX) Subroutine

30.28 The subroutine GRPX obtains the data associated with a group number when given the group number in register 1. The group table (GRPTBL) MTI entry table contains four 3-word entries. The first entry is for PBX and multiline hunt groups (MLHG); the second entry is for service circuits and tone and announcement circuits; and the third and fourth entries are for trunk circuits. The address of GRPTBL is put in registers 14 and 15. The high two bits of the 8-bit group number indicate which of the four MTI entries is the proper entry and is put in register 0. The low six bits, which allow for 64 groups, are isolated in register 1. These are the entry conditions used to call subroutine GNTRY in program XSLSUB.

30.29 Upon return from subroutine GNTRY, register 0 is zero if there is an error in obtaining the data or the data for the group number specified is not allocated. Otherwise, register 0 contains a one through three indicating

PBX/MLHG, service circuit group, or trunk circuit group. Then register 1 contains the first word of the data of the group and registers 14 and 15 contain the address of the data. The GRPX subroutine then returns to the calling program with the same data in the same registers.

30.30 The data for service circuit groups is a 4-word block of data for each group while the trunk circuit group is an 8-word block of data. The data for the first four words of the trunk circuit is the same as the four words of the service circuit group. These four words contain the largest member number field, the exist bit, the member bit, the traffic schedule field, the selection status block index, the member list index, and the circuit code number. The other four words of the trunk circuit group data contain fields and bits indicating the type of trunk it is and what is expected to come over the trunk as far as dialed digits. The PBX/MLHG data is eight words long containing the originating translation data, the selection status block index, the member list index, the packed billing number, the speed call indexes, the line class code index, the traffic schedule, the largest member number, the hunt size, the stop hunt member, and the night stop member.

H. Idle Circuit (IDLECKT, IDLESVC) Subroutine

30.31 The idle circuit (IDLECKT, IDLESVC) subroutine is one subroutine with two entry points, IDLECKT and IDLESVC. When a service circuit or a trunk circuit is to be idled, the subroutine IDLECKT or IDLESVC is called. The subroutine zeroes the selection status bit of the circuit. The entry condition for IDLECKT is the supervisory SPN in register 1 for trunk circuits and regular service circuits and the line SPN in register 1 for tone circuits and announcement circuits. The supervisory scan point for regular service circuits is the same as the service circuit number. So if the calling program knows it has the service circuit number, then the subroutine IDLESVC is the subroutine to be called. The only difference between IDLESVC and IDLECKT is that IDLESVC zeroes out the high bits because service circuits can only be in scanner 0, rows 0 through 15.

30.32 The exit conditions from IDLECKT and IDLESVC are the same. They are: if register 0 equals zero, an error was found; if register 0 equals one, the selection status bit of

SECTION 233-151-150

the circuit was zeroed; and if register 0 equals two, the bit was already zero.

30.33 The subroutines save the SPN, then call subroutine GPRADR with the SPN in register 1 as the entry condition. The return from GRPADR results with the group number and member number in register 1, the address of the group data in registers 14 and 15, and register 0 containing the return code. The return codes are zero being error or unallocated group, one being PBX/MLHG, two being service circuit group, and three being trunk circuit group. A return code of zero or one is an error for IDLECKT and IDLESVC; so if either is returned, then the subroutine returns an error condition to the calling subroutine.

30.34 If the return code from GRPADR is three, then the ignore bit and last look bit of the scan point are zeroed via the LOOKAT macro which calls the subroutine STATS. Then the lock-out test is performed.

30.35 If the return code from GRPADR is two, the test is made to see if the service circuit is a 3-port conference circuit, a regular ringer, or a special ringer. If the circuit is a 3-port conference circuit, the subroutine PORTIDLE is called. The entry conditions for PORTIDLE are the port number in register 0 and the member number in register 1. After these numbers are put into their specified registers from their saving registers, PORTIDLE is called. The PORTIDLE subroutine zeroes the bit indicated by the port and member number. If the status bit of the port is not idled, then PORTIDLE returns error. The IDLECKT subroutine in turn returns error. If the bit is idle, PORTIDLE returns success. There are two success returns: one when port 0 is idled or when port 1 or 2 is idled but the other port is already idled, or two when port 1 or 2 is idled but the other port is still busy.

30.36 For a return of one, the port number is tested. If the number is zero, the selection status bit of the circuit is to be idled. So the subroutine continues by zeroing the ignore bit and the lookat bit for the SPN of the circuit.

30.37 For a return of two, the two busy ports of the circuit are reconnected. The subroutine RCONINIT in program TREWAY reconnects the ports. The low bit of the port

number is put in register 0 and the line SPN is put in register 1 for the entry into RCONINIT. There is no return code from RCONINIT. After subroutine RCONINIT returns, IDLECKT returns a return code of one to the calling program even though the selection status bit of the circuit is not idled.

30.38 If the circuit is either type of ringer, the ignore bit of the scan point is set and the lock-out test is performed. If the circuit is not a ringer, then the ignore bit and look-at bit of the scan point are zeroed. The lock-out test is then performed.

30.39 The lock-out test is the call to the subroutine LKO_OUT in program TSVSUB to see if the lock-out state is invoked. If the lock-out state is invoked, the status bit is not changed and camp on (TSV_CMP) subroutine in program TSVEA is called next. If the lock-out state is not invoked, the status bit is idled.

30.40 To idle the selection status bit, the STATUS_S subroutine is called. The SPN in register 1 is the entry condition for the call to the STATUS_S subroutine which returns an error indication if there is one. If successful, register 1 contains the member number and registers 14 and 15 contain the address of the selection status block for the group of the scan point. The first word of the status block is the first of four header words. This word contains the last circuit idled, the maintenance busy counter, the not-in-service bit, and the all-busy bit. So before idling a circuit, the first word is obtained, and the last circuit idled is updated with the member number of the scan point, and all busy bit is zeroed. The updated word is then stored back into the first word of the block. Next, the status word containing the bit of the scan point (of the member) is obtained. This occurs by taking the high three bits of the member number and shifting the bits right by four then adding four to the new number for the four header words. This gives the INDEX into the block. Taking the low four bits of the member number gives the position of the circuit in the word. So the word which contains the bit of the circuit is obtained by indexing from the selection status block address. The bit of the circuit is tested to see if it is idle (zero). If it is idle, then a return code of two is returned to the calling program. But if the bit is busy (one), the bit is idled (zeroed) and the word is stored back to its

word in the block. Then the camp-on test is performed by calling the subroutine TSV_CMP.

30.41 The entry condition for TSV_CMP is the SPN in register 8. The TSV_CMP subroutine does not do a hold-all. There are no exit conditions. The TSV_CMP subroutine camps on the SPN for error analysis and then returns. The subroutine then returns a return code of one to the calling program.

I. Member List (MBR_LIST) Subroutine

30.42 The MBR_LIST subroutine obtains the first word and address of a particular member list. This is done by obtaining the member list index from the group data indicated by the address given upon entry to MBR_LIST. The member list index word is put in register 1. Next the address of the member list table (MEMLST) is obtained and put into registers 14 and 15. The MEMLST is a 12-word MTI entry table having four 3-word entries. The high two bits of the member list index indicate which of the four entries is wanted. Then the two bits are isolated in register 0 and the member list index is isolated in register 1. With the address of MEMLST in registers 14 and 15, the subroutine GENTRY in program XLSUB is called.

30.43 If the subroutine GENTRY is unsuccessful in obtaining the first word and address of the member list, a zero returns in register 0. Therefore, MBR_LIST returns an error return code of zero in register 0 to the calling program. If the subroutine is successful, then register 0 equals one, register 1 contains the first word of the member list of the group indicated, and registers 14 and 15 contain the address of the first word of the member list. The first word of the member list is a header word containing the number of spares field, the total number of members field (that is, the number of working members and the number of spares), and a format field which indicates either regular service circuit, tone and announcement circuit, trunk circuit, or PBX/MLHG. The rest of the list or block of words contains SPNs and distributor triplet addresses for regular service circuits and trunk circuits. The tone and announcement list contains TENs and the PBX/MLHG list contains TENs and the remote busy field. With the successful return from GENTRY, the address obtained is saved in the hold/get area for

return. The MBR_LIST subroutine then returns a return code of one to the calling program.

J. Member Data (MBRDATA) Subroutine

30.44 The MBRDATA subroutine is called to obtain the member data of particular group and member number regardless of the status of the member. The data for a regular service circuit member, a tone circuit member, an announcement member, a trunk circuit member, or a multiline hunt group member can be obtained. Also, if a member is not a working member but a spare, MBRDATA indicates this.

30.45 The entry condition for MBRDATA is the group and member number in register 1. The member number is isolated and saved. The GRPX subroutine is called with the group number. The return code is checked for the error return. If there is an error, the return code is one from MBRDATA. If the return code from GRPX is for MLHGs, then the data of the member is obtained. This is done by obtaining the largest number from the group to range-check the member number. If the member is out of range, the error return code is returned. If the member is within range, the member list for the group is obtained by calling MBR_LIST given the group data address. The return code is checked again for error. If there is no error, the address of the member list is returned. Next the 1-word entry containing the TEN is obtained, if it exists, and a successful return code of zero is returned with the TEN stored in the first word of SCRATCH (Fig. 35 for Issue 4 of the SO-2 generic and Fig. 36 for the 3E3 generic). If there is not a TEN, the member number is a spare and the return code is a two.

30.46 If the return from GRPX indicates a service or trunk circuit, the subroutine CKTLST is called with the member number and the group data address. The return code from CKTLST is returned as the return code for the return from MBRDATA. Also returned by MBRDATA for a successful return is the SPN in register 1, the TEN in SCRATCH+0 (which is the same for multiline hunt groups), and the distributor triplet address and circuit code in SCRATCH+1. If the data was not obtainable, the return code for a spare member is a two or for an error, the return code is a one.

K. Next Service Circuit (NEXT_SVC) and Next Trunk Circuit (NEXT_TRK) Subroutine

30.47 The next service circuit (NEXT_SVC) and next trunk circuit (NEXT_TRK) subroutines are the same subroutine with two entry points, NEXT_SVC and NEXT_TRK. The subroutines NEXT_SVC and NEXT_TRK are called when the connection to a previously selected service circuit or trunk circuit cannot be established and a new selection is required. The NEXT_SVC subroutine uses the service circuit number, which is the supervisory SPN, as an entry condition. As an entry condition, the NEXT_TRK subroutine uses the supervisory SPN for trunk circuits and line SPNs for tone circuits and announcement circuits.

30.48 The NEXT_SVC subroutine first isolates the 8-bit service circuit number and sets the SERVICE_CKT register to indicate that NEXT_SVC was called. Then NEXT_SVC becomes part of the NEXT_TRK subroutine. Given the SPN, the GETGRP subroutine in program XSLSPN is called to obtain the group number and member number for the specified circuit.

30.49 The subroutine TRKSEL in program EQPSEL is then called to select and set the selection status bit of the next idle circuit of the group indicated. The SPN of the circuit is returned and saved, the address of the group is returned only if NEXT_SVC was not called, and the TEN is stored in SCRATCH (Fig. 35 for Issue 4 of the SO-2 generic and Fig. 36 for the 3E3 generic). Also stored in SCRATCH is the distributor triplet address for trunk and regular service circuits and the circuit code number and the group number. The return code from TRKSEL is saved as the return code for NEXT_SVC and NEXT_TRK. Regardless of the fact that TRKSEL may not have been able to select another idle circuit, the IDLECKT subroutine is called. So with the first SPN which the subroutine entered, IDLECKT zeroes the selection status bit for the circuit associated with the scan point. Then the new SPN and the return code obtained from TRKSEL are returned.

L. Number of Working Members (NUWKMEMS) Subroutine

30.50 The NUWKMEMS subroutine is called when the number of working members in a group is wanted. This subroutine can be used only for service circuit and trunk circuit groups and not

for PBX/MLHG. The NUWKMEMS subroutine is entered with the group number in register 1. The subroutine GRPX is called first. Upon its return to NUWKMEMS, register 1 contains the first word of the data of the group and registers 14 and 15 contain the address of the data if the return code from GRPX is a two or three. If the return code is a zero or one (zero for error and one for PBX/MLHG), NUWKMEMS returns the error return code of zero in register 0 to the calling program.

30.51 The word returned contains the largest member number, the group exists (EXIST) bit, the working members (MEM) bit, and the traffic schedule. So the next EXIST bit is tested. If it is not set, the group does not exist and the error return code is returned. If it is set, the MEM bit is tested. If this bit is not set, there are zero working members in the group. Register 1 is zeroed and the successful return code of one is returned in register 0. If the MEM bit is set, the largest member is isolated into the register SAVE_MEMBER_NO. A one is then added to it to give the total number of members defined in the group. With the address of the group data in registers 14 and 15, the subroutine MBR_LIST is called to return the member list data. If MBR_LIST is unsuccessful in obtaining the data, NUWKMEMS returns the error return code. If MRB_LIST is successful, it returns the first word and address of the member list for the group.

30.52 The first word of the member list is in register 1. This word is a header word to the list. It contains the number of spares in the list, the total number of members, and a format code. If the number of spares is greater than the number of members, an error has occurred and NUWKMEMS returns an error. If the number of members is greater than the number of spares, the spares are subtracted from the members to give the total number of working members. With this result in register 1, a return code of one is returned in register 0 to the calling program.

M. PBX Busy (PBX_BUSY) and PBX Idle (PBX_IDLE) Subroutines

30.53 The PBX busy (PBX_BUSY) and the PBX idle (PBX_IDLE) subroutines are the same subroutine with two entry points, PBX_BUSY and PBX_IDLE. The subroutine PBX_BUSY busies and the subroutine PBX_IDLE idles the selection status bit for a line scan point of a PBX or MLHG

member. The subroutines use common code. The PBX_BUSY sets the low bit of the PBX_INDI register to indicate that PBX_BUSY is the subroutine called. The PBX_IDLE subroutine zeroes the PBX_INDI register to indicate that PBX_IDLE is the subroutine called.

30.54 Both subroutines use common code and with the line SPN (which is the TEN) in register 1, the subroutine LNDATA in program XSLSPN is called. This subroutine returns all the TEN data (which is the originating line data) into SCRATCH. If the line data is not obtained properly, register 0 contains a one or two and the subroutine returns indicating error. If the line data is obtained successfully, the return code from LNDATA is a zero in register 0. The first word of SCRATCH is obtained and the PBX bit is tested. If set, the TEN belongs to a PBX or MLHG member. If it is not set, the subroutine returns to the calling program with an error. If the TEN is for a PBX/MLHG, the member number is saved and the group number is isolated. Both of these numbers were obtained when the PBX bit was obtained. The group number obtained is only a 6-bit number instead of the usual eight bits because PBX/MLHG numbers range from 0 through 63. The member number is only a 6-bit number instead of the usual seven bits because the total number of members per PBX/MLHG is only 64.

30.55 Next, with the group number in register 1, the subroutine STATUS_G is called. Upon successful return from STATUS_G, register 2 contains the first word of the selection status block and the address of the block is in registers 14 and 15. The high three bits of the saved member number are now obtained and by adding four to them, they are converted into the index in the block for the proper word. The low four bits of the saved member are converted into the bit position. Then with the index, the word is obtained. The PBX_INDI register is tested for busy and idle. If it is set, the bit position indicated is busied (set) and the group peg counter is incremented; but if it is zero, the bit position indicated is idled (zeroed). The word is then stored back into the selection status block. The return code of one is returned to the calling program.

N. PBX Key Status (PKEYSTAT) Subroutine

30.56 The PKEYSTAT subroutine will check the PBX key status and set the status on or

off according to the status of register 0 specified in the entry conditions. PKEYSTAT is only used in the 3E3 generic.

30.57 If register 0 contains a zero, the condition is normal and no current is passed. If register 0 contains a one, the condition is off normal and the current is passed. This bit is saved in register 3. Given the first word of the universal subtranslator entry, status key will be saved in register 2 to index into a word in the PBX/MLHG status block. The status key bit will be set (0 or 1) according to the status of register 3. The PBX group number is also saved from bits 0 through 5 of the universal subtranslator entry and masked into register 1 so that bits 0 through 7 may be used as an entry condition into the subroutine GRPX.

30.58 The GRPX subroutine obtains the first word of the group data associated with the PBX group number (register 1) and its address (registers 14 and 15). GRPX also returns a one (1) in register 0 indicating a PBX/MLHG. Subroutine PKEYSTAT then checks that the information returned is correct and uses the group data address to call subroutine STATUS_A. Given this information, subroutine STATUS_A returns the first word of the selection status block entry in register 1 and its address in registers 14 and 15. Subroutine PKEYSTAT again checks that the information is correct. The value in register 3 is now checked and status key bit is set either on or off according to that value.

O. Port Busy (PORTBUSY) Subroutine

30.59 The subroutine PORTBUSY busies the port status bits of the 3-port conference circuit. The PORTBUSY subroutine should be called only when a 3-port conference circuit is to be busied. This subroutine busies all three ports at one time. Register 1 contains the member number of the circuit to be busied into PORTBUSY. Register 0 contains the return code on exit from PORTBUSY. The return codes are zero for error and one for success.

30.60 The PORTBUSY subroutine saves the member number. The subroutine then calls PORTSTAT with the member number in register 1. The PORTSTAT subroutine obtains the word that contains the three status bits for the member indicated. The return code from PORTSTAT in

SECTION 233-151-150

register 0 is either zero for error or one for success. For success, register 1 contains the status word unchanged. Registers 14 and 15 also contain the address of the status word.

30.61 Next the three port bits for the member need to be set to ones. The low two bits of the member number are obtained and multiplied by four. The number just created is the bit position in the word for the particular member. The status word contains status bits for four members. The low three bits of every four bits are the status bits of the ports. All three port bits are to be busied. Register 4 is loaded with a seven (three bits set). Then register 4 is rotated left by the modified member number. The status word and register 4 are ORed together setting the port bits. Then the status word is stored back into the ports status block and PORTBUSY returns to the calling program with a return code of one.

P. Port Idle (PORTIDLE) Subroutine

30.62 The subroutine PORTIDLE idles the port status bits of the 3-port conference circuit. The PORTIDLE subroutine should be called only when a 3-port conference circuit is to be idled. This subroutine only idles the port indicated for the member indicated. Register 0 contains the port number and register 1 contains the member number of the circuit to be idled for entry into PORTIDLE. Register 0 contains the return code on exit from PORTIDLE. The return codes are zero for error, one for success for port 0 and for ports 1 and 2 when the other port is idle, and two for success for ports 1 and 2 when the other port is busy.

30.63 The port number and the member number are saved. With the member number, the PORTSTAT subroutine is called. The PORTSTAT subroutine returns error or success. For a successful return, register 1 contains the status word of the port bits for the member indicated and registers 14 and 15 contain the address of the status word. In each status word are three port bits for four members. Every fourth bit is the unused bit. Next save the two low bits of the member number and multiply them by four. Then test the port number for zero. If the port number is zero, then all three status bits are zeroed in register 1. The other bits are not changed. The word is stored back into temporary store idling the status bits of the port. Then a return code of one is returned.

30.64 If the port number is one or two, that port is to be idled. Port 0 is tested. If port 0 is idle, then the error return is returned. If it is busy, the port bit indicated (one or two) is tested. If the port indicated is idle, the error return is returned. If it is busy, it is idled. If port 1 is idled, port 2 is tested next. If port 2 is idled, port 1 is tested. If the tested port is idle, then the return code one is returned. If it is busy, the return code two is returned.

Q. Port Status (PORTSTAT) Subroutine

30.65 The PORTSTAT subroutine obtains the port status bits of the 3-port conference circuit. The PORTSTAT subroutine obtains the status bits through the 3-word MTI entry 3PCSTAT. The status bits are in temporary store. The entry condition is that register 1 contains the member number of a 3-port conference circuit. The exit conditions are that register 0 contains a zero for error and a one for success. When success is returned, register 1 contains the status word containing the status bit of the member unchanged. Registers 14 and 15 also contain the address of the status word.

30.66 The PORTSTAT subroutine obtains the address and first word of 3PCSTAT and loads them into registers 14 and 15 and into register 0, respectively. With the member number in register 1, the subroutine GNTRY3 in program XSLSUB is called. The subroutine GNTRY3 obtains the status word indicated by the member from the status block of the port. The subroutine GNTRY3 returns the status word in register 1 and the address of the word in registers 14 and 15. Also, in register 0 is the return code of one for success. If GNTRY3 is not able to obtain the word, then register 0 is zero for error. The subroutine PORTSTAT then tests for the error return. If it is error, PORTSTAT returns error. If the word is obtained, the address registers are saved. The subroutine PORTSTAT then returns success to the calling program.

R. Port Test (PORTTEST) Subroutine

30.67 The subroutine PORTTEST returns the three status bits for a particular member of a 3-port conference circuit. The PORTTEST subroutine is entered with the port 0 TEN in register 1. The exit condition from PORTTEST is zero for error or one for success in register 0.

For success, register 1 contains the status bits for the 3-port circuit. The three bits imply the status of each port. Zero is idle and one is busy.

30.68 The PORTTEST subroutine translates the port 0 TEN into the group number and member number by calling the GRPADR subroutine. Subroutine GRPADR also puts the port number in word 3 of SCRATCH. If GRPADR returns the error condition, PORTTEST returns error. With a successful return from GRPADR, the member number is isolated in register 1 for the call of PORTSTAT. The member number is then saved for later use. If PORTSTAT returns error, PORTTEST returns error also. If success is returned from PORTSTAT, then register 1 contains the status word containing the port status bits of the member number. Next the bits of the member number have to be isolated. The low two bits of the member number indicate which of the four sets of status bits is the wanted bit. By obtaining the low two bits of the member number and multiplying them by four, the starting position of the status bit is obtained. The word is then rotated right by the bit position to move the status bits to the low bits in the word. All the other bits are zeroed, leaving the three status bits in register 1 indicating the status of each port of the particular conference circuit. Subroutine PORTTEST then returns a successful return code.

S. Select an IDLE Circuit (SELECTS) Subroutine

30.69 The SELECTS subroutine is called only by subroutines TRKSEL and MEMSEL. The reason for this subroutine is that more registers were needed than were available. The entry conditions for SELECTS are the first member to be searched, register 8 (FIRST_MEMBER), and the last member to be searched, register 7 (LAST_MEMBER). If the search is starting from the last circuit idled, then the first member is the last circuit idled plus one and the LAST_MEMBER is the largest member number. If there were no idle members from the last circuit idled to the largest member number, then the next time SELECTS is called, the selection starts at the beginning of the list with FIRST_MEMBER equal to zero and LAST_MEMBER equal to the last circuit idled.

30.70 The exit conditions are register 0 equals zero (means an idle circuit was not found), register 0 equals one (means an idle circuit was

found), register 4 contains the status word, and register 7 contains the status bit. The status word is the INDEX into the selection status block. The status bit is the CKT_POSITION of the selected bit out of the possible 16 bits in the word.

30.71 Subroutine SELECTS subtracts the FIRST_MEMBER from LAST_MEMBER to give the total number of circuits to be searched in COUNTER. If FIRST_MEMBER is greater than LAST_MEMBER, the return code of no idle circuit found is returned. But if FIRST_MEMBER is less than LAST_MEMBER, then COUNTER is set up as to how many words are to be searched. There are a possible 128 circuits per group, so the member number is a 7-bit number. The high three bits indicate the total number of words to be searched.

30.72 The next two masks are needed for the first and last words to be searched. There are 16 status bits per word. The FIRST_MEMBER is the status bit of the first circuit to be tested for an idle state; therefore, the MASK is set up so that all bits less than the FIRST_MEMBER within the multiple of 16 are set to ones. These bits will not be scanned in the search. Next, a LAST_MASK is set up as to which bit is to be searched last via LAST_MEMBER. This mask is set up by the same method as MASK except that the circuits not to be searched are greater than LAST_MEMBER. Therefore, the bits set to one are those greater than LAST_MEMBER within the word.

30.73 The INDEX is set up next. The INDEX is set up the same way COUNTER was set up, but with INDEX being the high three bits of the first seven of FIRST_MEMBER. The status bits start in the fifth word of the block so three is added to INDEX and one is added just before the loading of the status word. If the COUNTER is zero, then there is only one word to be searched and both masks are ORed together for the first and last search argument with the results in LAST_MASK. If the COUNTER is not zero, the masks are not changed. Now, the loop for the search begins. A one is added to INDEX and the status word (STATUS) is obtained, given the status address (STATUS_ADDR) and the INDEX. The COUNTER is tested for zero. If it is zero, LAST_MASK is ORed to STATUS. If COUNTER is not zero, MASK is ORed to STATUS. The STATUS is then searched for a zero bit which

indicates an idle circuit. If a zero bit is not found, MASK is zeroed so that if the next word to be searched is not the last word, the MASK will allow all bits to be tested. If COUNTER is now zero and no idle circuit is found, the subroutine returns to the TRKSEL subroutine with a return code of zero indicating that no circuit was found. But if COUNTER is not zero, one is subtracted from it and the subroutine continues the loop to obtain the next status word. If an idle circuit is found not busy, it will be set busy by the TRKSEL subroutine. The INDEX and CKT_POSITION is returned to the TRKSEL subroutine with the return code of one.

T. Selection Status Block Address (STATUS_A) Subroutine

30.74 The subroutine selection status block address (STATUS_A) is called when the selection status block is to be obtained. The subroutine STATUS_A receives the address of the group in registers 14 and 15 and obtains the selection status block index from word 1 of the group data. The high two bits of this word are not part of the index. These bits are the index into the 4-entry GRPSTAT table. So the next GRPSTAT table address is obtained.

30.75 The GRPSTAT table is an MTI table in the program TDATA. There are four 3-word entries that can be accessed by the 2-bit field. A zero is for the PBX/MLHG, a one is for the service circuit groups, and a two and three are both for the trunk circuit groups. The 2-bit field is put in register 0, the 14-bit selection status block index is put in register 1, and the address of the GRPSTAT table is put in registers 14 and 15. This is the entry condition for the subroutine GNTRY in program XLSUB.

30.76 An error return from GNTRY is possible which returns a zero in register 0. If there is no error, then register 0 equals one and register 1 contains the first word of the selection status block. In this word is the last circuit idled field, the maintenance busy counter field, the group is not in service indicator, and the all circuits are busy indicator. Also, in registers 14 and 15 is the address of the selection status block which is in temporary store. The address registers are saved in the hold/get counter, then a return code of one is returned by STATUS_A to the calling program.

U. Get Selection Status Block With Group Number (STATUS_G) Subroutine

30.77 The subroutine STATUS_G is called when selection status block of the group is to be obtained. The subroutine STATUS_G is entered with the group number in register 1. The subroutine GRPX is called to return the first word of the group data in register 1 and the address of the data in registers 14 and 15. If an error is returned from GRPX, which is zero register 0, then STATUS_G returns the error return code to the calling program. The return codes GRPX 1 through 3 are valid. A one is for PBX/MLHG, a two is for service circuit groups, and a three is for trunk circuit groups.

30.78 Next the STATUS_A subroutine is called with the address of the group. If there was an error upon returning from STATUS_A, the 0 return code is returned and it is passed on to the calling program by STATUS_G. If there is no error, the selection status block address is put in registers 14 and 15, register 1 contains the first word of the block, and register 0 equals a one. Then the address registers are saved in the hold/get counters. Next, subroutine STATUS_G returns to the calling program with the return code from subroutine STATUS_A.

V. Get Selection Status Block With Scan Point Number (STATUS_S) Subroutine

30.79 The STATUS_S subroutine is called when the selection status block of the circuit (given the SPN) is wanted. The subroutine STATUS_S requires the SPN in register 1 as the entry condition. The SPN is translated into the group number and member number by the subroutine GETGRP in program XLSUB. If there is an error in translating the SPN to the group number and member number, GETGRP returns an error code of zero and, in turn, STATUS_S returns the zero error return code to the calling program. But if there is no error, the member number is saved to be returned later. So the group number is translated next.

30.80 With the group number still in register 1, the subroutine GRPX is called. The return code in register 0 is from zero through three where a zero is error, a one is the group number for a PBX/MLHG (which is an error in this case), a two is the group number for a service circuit, and a three is the group number for a trunk circuit.

So, subroutine STATUS_S returns to the calling program with the error code for zero and one, and continues for two and three. Also returned by subroutine GRPX is the first word of the group data and the address of the group data. Now the largest member number in the first word is isolated. If the member number which was saved from the GETGRP subroutine is greater than the largest member number, STATUS_S returns the error return code of zero. If the member number is within range, the subroutine STATUS_A is called with the group data address in registers 14 and 15. If subroutine STATUS_A is successful, the zero error return code is returned and STATUS_S passes it on. But if STATUS_A successfully obtains the selection status block of the group, the address is then in registers 14 and 15 and the first word of the block is in register 1; register 0 equals one for the return to STATUS_S. STATUS_S then stores the address in the hold/get counter for the return and puts the member number in register 1. Subroutine STATUS_S now returns to the calling program with the return code of one in register 0.

W. Transient Call Record Selection (TCRSEL) Subroutine

30.81 The subroutine TCRSEL tries to select an idle TCR. There are no entry conditions for the call to subroutine TCRSEL. The address of the TCRs is obtained from ATSD and saved. The size of a TCR is also obtained from ATSD. All TCRs are selectable except the last TCR which is used for maintenance and diagnostic purposes. After the address of the last selectable TCR is obtained, it is returned to the calling program.

30.82 Every TCR is tested until an idle one is found. This is done by testing the ACTIVE bit in the first word of the TCR. If the ACTIVE bit is set, then that TCR is in use. But if the bit is zero, that TCR is idle. If no idle TCRs are found, the return code of zero is returned in register 0. The search for an idle TCR is started at the address of the TCRs. The first word of a TCR is obtained and the ACTIVE bit is tested. If it is set, the search is ended. If the ACTIVE bit is not set, the search is continued until the beginning address is equal to the last address. When an idle TCR is found, the ACTIVE bit is set in the first word of the TCR. The contents of the first word were loaded into register 0 as the ACTIVE bit was set. So the ACTIVE bit is now checked to see if it is set. This is done because between the time the idle TCR was found and the ACTIVE

bit is set, the interrupt program may select and busy the TCR just obtained. To ensure that the TCR was not stolen, the bit is tested. If the bit is set, then the search for an idle TCR continues from where it left off. If the bit is not set, the rest of the word is zeroed.

30.83 The TCR number is obtained next. This is done, in effect, by subtracting the first TCR address from the selected TCR address and dividing the entry by the size of the TCR. The address of the TCR is saved in the hold/get counters for registers 14 and 15. Next, the rest of the selected TCR is cleared. This is done by adding one to the TCR address, loading the entry size of the TCR minus one in register 0, and calling the subroutine CLR_WRDS in program CSYSUB. The TCR number is then put in register 1 as part of the exit return. The TCR address is returned in registers 14 and 15 and the return code of one is returned in register 0.

X. Test Circuit (TESTCKT) and Test Service Circuit (TESTSVC) Subroutines

30.84 The test circuit (TESTCKT) and test service circuit (TESTSVC) subroutines are really just one subroutine with two entry points, TESTCKT and TESTSVC. The subroutine TESTCKT is called when a supervisory scan point of a trunk circuit or a regular service circuit is to be tested for busy or idle, or when a line scan point selection status bit of a tone circuit or announcement circuit is to be tested for busy or idle. The subroutine TESTSVC is called when the service circuit number selection status bit of a circuit is to be tested for busy or idle. Subroutine TESTSVC is called only when a service circuit number is present and not a SPN. So TESTSVC first zeroes the high bits beyond the service circuit number. Therefore, the SPN and the service circuit number are now the same. At this point, subroutines TESTCKT and TESTSVC continue as one subroutine.

30.85 The SPN is saved in register 1 and subroutine GRPADR is called. If error is returned to TESTCKT from GRPADR, then TESTCKT returns error to the calling program. If GRPADR returns the trunk circuit return code, the port number register (PORTNO_REG) is zeroed. This implies that the circuit tested is not a 3-port circuit and the subroutine continues. If subroutine GRPADR returns the service circuit return code, the group number is tested for the 3-port circuit

number. If the number is not for the 3-port circuit, TESTCKT continues as for a trunk circuit. But if the group number is for 3-port circuit, then the SPN is put into register 1 for the call to subroutine PORTTEST. Subroutine PORTTEST returns the 3-port status bits for 3-port circuit indicated. If PORTTEST returns error to TESTCKT, then TESTCKT returns error to the calling program. If PORTTEST returns successfully, then the status bits are in register 1. They are then saved in PORTNO_REG and subroutine TESTCKT continues.

30.86 Next, with the SPN in register 1, the subroutine STATUS_S is called and it returns the member number of the circuit in register 1 and the address of the selection status block in registers 14 and 15. In register 0, there is a one indicating a successful return. If the block was not obtainable, then register 0 would contain a zero and the subroutine would return an error to the calling program. Upon returning, with no error from STATUS_S, the 7-bit member number is divided into two numbers. The low four bits are the circuit position in the status word and the high three bits are the index into the words of the status bits. Four is added to the index to account for the four header words in the block and then a word is obtained from the block using the index and address of the block. The bit in that word indicated by the circuit position is now tested for a zero (busy) or a one (idle). If the bit of the circuit is idle, a return code of two is returned; but if the bit of the circuit is busy, a return code of one is returned to the calling program.

Y. Member Select (MBRSEL) Subroutine

30.87 The subroutine MBRSEL is usually called to get a particular type of test circuit from the test circuit group although it may be called whenever a particular member of any service circuit or trunk circuit group is wanted. Subroutine MBRSEL is a second entry into the subroutine TRKSEL.

30.88 The entry condition for MBRSEL is the group number and member number. The subroutine MBRSEL disregards the group-not-in-service indicator and selects the circuit only if it is idle. So if the circuit is idle, the selection status bit is set to busy and the ignore bit is set to ignore. The SPN of the circuit, TEN, distributor triplet address, and circuit code number associated with the group are returned. The group peg counter

is not incremented when MBRSEL is called. The return data previously mentioned is the same data as that returned by TRKSEL. The return codes are: a zero and a one equal error (a one is only used by TRKSEL); a two equals the circuit associated with the member number and is busy and no data is returned; a three equals the member number and is not defined but is a spare; and a four equals the circuit associated with the member number and is idle so it is busied and the data is obtained.

30.89 The MBRSEL subroutine sets up the register (MEMSEL) to indicate that MBRSEL is the subroutine called. The member number is then isolated and saved. The subroutine MBRSEL now merges with the TRKSEL subroutine.

Z. Trunk and Service Circuit Select (TRKSEL) Subroutine

30.90 The TRKSEL subroutine is called when an idle circuit of a particular group is wanted whether it is a regular service circuit group, a tone circuit group, an announcement circuit group, or a trunk circuit group.

30.91 The entry conditions for subroutine TRKSEL are the group number and a TCR address. An idle circuit is found by searching the selection status bits for the group. The search starts with the circuit after the last circuit idled and continues until a new idle circuit is found or until all bits have been searched and the last circuit idled is the only circuit idle; then it is selected. The selection status bit of the circuit is busied and the ignore bit is set. Also the group peg counter is incremented if this is a first try for trunk and service circuits and a second try for service circuits. If all the circuits are busy, the search fails; then the all-circuits-busy indicator is set. The group peg counter is incremented for trunk and service circuits on the first try only. The group overflow counter is incremented for service circuits on a first try only and for trunk circuits on a second try only. Also, anytime this counter is incremented and if the maintenance busy counter is not zero, the maintenance busy overflow counter (MMMBO) is incremented.

30.92 The exit condition for an unsuccessful circuit selection is a zero in register 0. This error code is returned when the group is unallocated or unassigned, when the group is a PBX/MLHG, when the selection status block is unobtainable, or when

the member number is out of the range of the group.

30.93 The exit condition for successful circuit selection is the SPN returned for trunk circuits or the service circuit number (which is the SPN for regular service circuits) or no SPN for tone or announcement circuits. Also, the address of the group data is returned if a trunk group number was inputted. In SCRATCH is the TEN for all circuits, the distributor triplet address for trunks and regular service circuits, the circuit code number for all circuits, and the group number for all circuits.

30.94 The subroutine TRKSEL zeroes the MEMSEL register to indicate that TRKSEL is the called subroutine and not MBRSEL. The second try (2NDTRY) bit from the TCR is obtained. This bit is tested as to whether or not the group peg counter and the group overflow counter are to be incremented. There is a trunk and service circuit (SVC_TRK) register to distinguish between trunk and service circuits which is used to tell which counter to peg. The register is zeroed at this time.

30.95 It is at this point that subroutine MBRSEL enters the TRKSEL subroutine. The subroutine then isolates the group number and stores it in SCRATCH. With the group number, GRPX is called which returns the address and the first word of the group data. The group data is an 8-word block for trunk circuits and a 4-word block for all service circuits. In the group data for both service and trunk circuits are the largest member number, the group exist bit, the members exist bit, the traffic schedule, the selection status block index, the member list index, and the circuit code number. The extra four words of the trunk circuit data contain bits and indicators as to the type of trunk that is found assigned to the group.

30.96 If a PBX/MLHG number was given, that would be an error. Also, if there was a problem in obtaining the MTI entry GRPTBL data, that would be an error. If the group number is for a trunk circuit, the address of the group data is saved for the return from this subroutine and the SVC_TRK register is set.

30.97 The circuit code number found in word 3 of the group data is saved for later testing. Next the group exist (EXIST) group bit is tested

to see if there is data associated with the number. If there is not any data associated with the number, the subroutine returns the error return code. If there is, the member (MEM) bit is tested to see if there are valid members associated with the group and not just spare members. If all the members in the group are defined as spares, the subroutine returns the no-members-return code. If there are valid members in the group, then the largest member number is put into LARGEST_MEMBER and saved and the group data address is put into registers 12 and 13 and saved.

30.98 The subroutine status address (STATUS_A) is entered with the group address in registers 14 and 15. Subroutine STATUS_A returns the first word in register 1 and the address of the selection status block for the particular group specified in registers 14 and 15. The address registers are exchanged with registers 12 and 13 becoming the STATUS_ADDR and registers 14 and 15 again containing the group data address. The selection status blocks are in temporary store. The block contains the last circuit idled, the maintenance busy counter, the all busy bit, the not-in-service bit, the group peg counter, the group usage counter, the group overflow counter or through switch peg counter, and the selection status bits for all the members in the group for service circuits and trunk circuits. The first word of the block contains the all busy (AB) bit which is tested to see if every member of the group is busy. If the group is all busy, then the all-circuits-busy return code is to be returned. Next the MEMSEL register is tested. If the MBRSEL subroutine is the subroutine called, then the all circuits busy return code is returned. If the TRKSEL subroutine is the subroutine called, then the AB bit is set regardless if it is set or not. The second try (2NDTRY) bit is then tested from the TCR. If this is the second time for trying to get a circuit, the SVC_TRK register is tested to see if it is a service or trunk circuit that is to be obtained. If it is a service circuit, the busy return code is returned. If it is a trunk circuit, the group overflow counter is incremented and the maintenance busy counter is tested for zero. If the counter is not zero, the MMMBO in ATSD is incremented and the busy return code is returned. If the counter is zero, the busy return code is returned immediately. If the 2NDTRY bit is not set, and if it is only the first try, the group peg counter is incremented and the SVC_TRK register is tested. If it is a trunk circuit, the busy return code is returned; but if it is a service

circuit, the group overflow counter is incremented. The group maintenance busy counter is tested for zero. If it is zero, the return code is returned. If it is not zero, the counter is first incremented and the busy return code is returned.

30.99 Upon testing for the AB condition, if it is not found, the subroutine continues. Next the MEMSEL register is tested for the subroutine called. If subroutine MBRSEL was called, the registers will be set up for the selection. If subroutine TRKSEL was called, the not-in-service (NIS) bit is tested. If the NIS bit is set, the group not-in-service return code is returned. If the NIS bit is not set, then the registers are set up for the selection.

30.100 Three registers are set up for the selection of an idle circuit: WHICH_HALF, FIRST_MEMBER, and LAST_MEMBER. The WHICH_HALF register indicates which half of the list is to be searched. A zero implies the first half, which is from the beginning of the list (first circuit) to the last circuit idled. A one implies the second half, which is from the last circuit idled to the end of the list (last circuit). Therefore, it is possible to have to set up the registers twice to try to select a circuit. If subroutine TRKSEL was called (and this is the first time for setting up the registers), the FIRST_MEMBER register is the next number after the last circuit idled number and the LAST_MEMBER register is the largest member number (LMN), which is the end of the list. At this point, WHICH_HALF is set to one. But if subroutine MBRSEL was called, then FIRST_MEMBER and LAST_MEMBER equal the member number and WHICH_HALF equals zero. Therefore, the registers are set up to search only one time.

30.101 The subroutine SELECTS is then called, which tries to select an idle circuit within the limits just mentioned. If successful on the return from subroutine SELECTS, register 0 equals one; register 4 equals the status word (INDEX), which is the number of words in the block where an idle circuit was found; and register 7 equals the status bit (CKT_POSITION), which is the bit number of the word that contains an idle circuit. If subroutine SELECTS was unsuccessful, then register 0 is equal to zero. If the search was unsuccessful, the WHICH_HALF register is tested to see if all the bits were tested; if they were, then the busy return code is to be returned. The

subroutine is continued as described above when the AB bit is set.

30.102 If the WHICH_HALF register indicates that all the bits were not tested, this indicates that from 0 to the last circuit idled is to be tested. The registers have to be set up to search and select an idle circuit from the first half of the selection bits, which means a call to subroutine SELECTS again. The WHICH_HALF register is zeroed so that the first half will be tested. The FIRST_MEMBER register is zeroed for the search to start at the beginning of the list with member 0. The LAST_MEMBER register is set to the last circuit idled as the last possible bit to select. Subroutine SELECTS is then called and the subroutine continues as described above when SELECTS was called.

30.103 If the selection was successful either time, then the circuit is to be marked busy. The word containing the selection bit is obtained, the bit is set, and the word is stored back into temporary store. Next the member number is created given the INDEX in the status block that was used to set the bit and the CKT_POSITION that was used to set the bit in the word indicated. There are four header words in the status block, so four is subtracted from the INDEX register. With 16 selection bits per word, the INDEX register is rotated left by four, then the CKT_POSITION register is inserted into the INDEX register to give the member number of the selected circuit.

30.104 Now the circuit code number previously saved is tested. If the number is for the 3-port conference circuit, then the port status bits are to be set. To set the status bits, the member number is put in register 1 as the entry condition into subroutine PORTBUSY. Subroutine PORTBUSY is called. This subroutine sets the three status bits of the member indicated implying that all three ports are busy. The return codes from PORTBUSY are error and success. Register 0 is tested for error. If there is an error, the error return is returned by subroutine TRKSEL/MBRSEL. If the bits were busied successfully, the data for the member is obtained next. If the circuit code number is not for the conference circuit, then PORTBUSY is not called and the data for the member is obtained next.

30.105 The member number is loaded into register 1 for the call to subroutine CKTLST. Also,

the member number is saved in FIRST_MEMBER for possible future use. The address of the group data is still in registers 14 and 15. Now subroutine CKTLST is called to obtain the data for the member specified in register 1. If the subroutine is successful, then the return code in register 0 is a zero and the scan point number is in register 1. The TEN, distributor triplet address, and circuit code number are returned in SCRATCH which subroutines TRKSEL and MBRSEL will return. Now the group peg counter is to be incremented. If subroutine MBRSEL is the subroutine called, then the counter is not incremented. Next, register 1 is tested for zero. If it is zero, the member number is for a tone or announcement group because tone and announcement circuits do not have scan points per circuits. Therefore, the ignore bit is not set for any scan point and a return code of four is returned. If register 1 is not zero, then there is a scan point and its ignore bit is set and a return code of four is returned. If subroutine MSRSEL is not the subroutine called, the group peg counter is to be incremented. The 2NDTRY bit is tested. If it is not set (first try), the counter is incremented and register 1 is tested for the SPN and the subroutine continues as described above. If it is set (second try) and SVC_TRK is set (trunk circuit), then the SPN test is done and continues as described above. But if it is a second try and SVC_TRK is not set (service circuit), the counter is incremented and the SPN test is done and continues as described above.

30.106 If a return code of one is returned from subroutine CKTLST, there was an error in obtaining the data and the subroutine should try to select another circuit. If subroutine MBRSEL was the called subroutine, a return code of zero is returned because the specific circuit was not obtainable. But if subroutine MBRSEL was not called, the registers are set up to search again. The FIRST_MEMBER register equals the last circuit idled plus one, the LAST_MEMBER register equals the largest member number, and the WHICH_HALF register equals one. Subroutine SELECTS is called and continued as described before.

30.107 If a return code of two is returned from subroutine CKTLST, the member selected is a spare member. The selection status bits of the spare member are selected and busied and cannot be reselected again until these bits in temporary store are cleared (zeroed) by the stable clear of the system. Therefore, when a spare

member is selected, subroutine CKTLST returns the return code to indicate it has been selected and the subroutine then continues the search to find an idle bit of a defined circuit. If subroutine MBRSEL was the called subroutine, then a return code of three is returned to indicate that the member is a spare. If subroutine TRKSEL was called, the search is to be continued from where it left off. The FIRST_MEMBER register is to start with the next member so one is added to the saved FIRST_MEMBER. Then the WHICH_HALF register is tested to see what the LAST_MEMBER register should contain. If it is zero, the first half of the list is being tested. The LAST_MEMBER register contains the last circuit idled and the subroutine then branches to select another idle member. If it is one, the last half of the list is being tested and the LAST_MEMBER register obtains the largest member number; the subroutine then branches to select another idle member.

31. 3-DIGIT TRANSLATION (XSL3DG) PROGRAM

31.01 The function of the 3-digit translation (XSL3DG) program is to provide routing and charging information for a call on the basis of the first three or six digits received from a customer plus the class of service of the customer.

31.02 The XSL3DG program is based on the subroutine 3DIGIT with the multiple entry points of 7DIGIT, 10DIGIT, and CODE_NDX. Needed to complete a 3-digit translation are the subroutines RI_EXP, GARI, and CH_EXP. Also in this program is subroutine 1DIGIT, used in the translation of one digit or incoming trunks; and subroutine TOTANLST, used to obtain an entry in the terminating office test access number table (TOTANTBL).

31.03 The following paragraphs briefly describe the subroutines contained in the XSL3DG program.

A. Code Index (CODE_NDX), 7-Digit (7DIGIT), and 10-Digit (10DIGIT) Subroutines

31.04 The code index (CODE_NDX), 7-digit (7DIGIT), and 10-digit (10DIGIT) subroutines are really entry points within the 3DIGIT subroutine.

31.05 The 7DIGIT entry into 3DIGIT sets the conflicting code register to zero and then obtains the code index from the TCR. Subroutine

7DIGIT branches into the 3DIGIT code where the code index has been obtained.

31.06 The 10DIGIT entry into 3DIGIT sets the conflicting code register and then obtains the code index from the TCR. Subroutine 10DIGIT branches into the 3DIGIT code where the code index has been obtained.

31.07 The CODE_NDX entry into 3DIGIT obtains the code index from the TCR and then branches into the 3DIGIT code where the code index has been obtained.

31.08 The subroutines 7DIGIT and 10DIGIT are called when there are 7 or 8 digits or 10 or 11 digits, respectively, in the TCR. A register is also set to indicate which subroutine is called. Either of these subroutines may be called after subroutine 3DIGIT is called and a return code of nine is returned. The TCR should contain the screening class number, the zero and one prefix indicators, the call forwarding indicator, and the code index that was stored when 3DIGIT was called.

31.09 The subroutine CODE_NDX is called for operator calls and when an office code is shared by two offices. The TCR contains the screening class number, the zero and one prefix indicators, and the call forwarding indicator. The code index for operator calls, which is one, is in the TCR. The code index for shared office codes, which was obtained from a 4-digit translation, is in the TCR also.

B. 3-Digit (3DIGIT) Subroutine

31.10 The subroutine 3DIGIT is called when there are three BCD digits in the TCR. If there are four digits in the TCR when 3DIGIT is called, then the first digit is a prefix digit of zero or one with indicators in the TCR implying the type of prefix. In the TCR the signal digit indicates the number of digits. Also in the TCR is the screening class number of the calling party indicating the type of service to be given and a call forwarding bit indicating if this call is being forwarded. When the subroutine 3DIGIT is called, a register is set to indicate that it has been called.

31.11 The 3DIGIT subroutine obtains the proper three BCD digits from the TCR, converts any 11X codes dialed to X11, and then converts the digits to binary and subtracts 200. This makes

it easier to use the number as an index because the lowest the three BCD digits can be is 200. The table FATTBL is the 12-word MTI table defined in program TDATA. The subroutine GNTRY3 is called each time MTI table, which is a 3-word entry type or a 3-word entry table type, is obtained. Subroutine GNTRY3 checks for a valid entry and index and returns the proper word and address. The foreign area translator (FAT) number selects one of the four 3-word entries in the table FATTBL and with the address from the 3-word entry, a local or FAT is selected. The conflicting code register is set to two to imply that 3DIGIT is the subroutine called. The first time through 3DIGIT, the FAT in the TCR is zero designating the local area translator. Both types of translators may be either a 400-word translation table using half-word entries or a 256-word search table using 2-word entries. The binary number is indexed into the table and the code index is obtained.

31.12 It is at this point that subroutines 7DIGIT, 10DIGIT, and CODE_NDX converge into the 3DIGIT subroutine.

31.13 The address of the code index expansion table in the CDIEXP entry and the 3-word MTI entry from TDATA is obtained next. The code index is then indexed into the table and a 2-word entry is obtained. There are eight types of entries:

- (1) *Vacant*
- (2) *Ignore any prefixing*
- (3) *A one prefix normally received*
- (4) *Normally no prefix is received*
- (5) *Ignore a one prefix*
- (6) *FAT required on next three digits*
- (7) *Conflict between area code and office code*
- (8) *Preroute peg count desired.*

31.14 The *vacant* entry is an error. The 2-word entry is zero-filled and the subroutine returns a return code of zero.

31.15 The *ignore any prefixing* entry contains a type of one, a screening table number, and a direct route index. As a type one, the subroutine then obtains the screening class number from the TCR and checks to see if screening is required. If screening is required, the screening table number is obtained and the address of the screening tables in the SCTBLS 3-word MTI entry is obtained. There are up to 32 screening tables of equal size up to 31 entries, so the screening table number is then indexed into the tables obtaining the proper table. The screening class number is next indexed into this table obtaining a 1-word entry containing a toll diversion (TDV) indicator, a charge index, and possibly another route index. If this route index is nonzero, it is stored in the TCR along with the TDV indicator and the charge index. If it is zero, then the direct route index is stored in the TCR along with the TDV indicator and the charge index.

31.16 Also in the TCR is a call forwarding (FORWARD) indicator. When set, this bit indicates that the digits in the TCR are for the forward-to number of the called party (forwarded number). Therefore, the calling party should not be charged for the forward-to number but charged only for the forwarded number (called party). The route is then to the forward-to number. Therefore, if the FORWARD bit is set, neither the charge index in the TCR nor the TDV bit is changed. The new route index, obtained by reentry into the 3DIGIT program after the storing of the call forwarding digits, is stored in the TCR.

31.17 If after obtaining the screening class number it is decided screening is not required, then the direct route index is stored in the TCR. The TDV indicator and charge index, stored previously before entry into the 3DIGIT subroutine, remain unchanged.

31.18 The subroutine then calls the subroutine RI_EXP. Subroutine 3DIGIT passes to RI_EXP the same route index that was stored in the TCR. Subroutine RI_EXP then returns to 3DIGIT as return codes the type of entry ranging from zero through seven, the first word of the route index expansion entry, and the address of the entry. In turn, subroutine 3DIGIT returns to the calling program the same return codes, word entry, and address.

31.19 The *one prefix normally received* entry contains a type two, a screening table number, and a direct route index. It also contains a zero prefix code index and a no prefix code index. Therefore, this entry is used when one prefix is normally received. The entry also allows or denies zero prefixing or no prefixing.

31.20 If there is a one prefix, the type two entry is processed identically to a type one entry from the point of obtaining the screening class number from the TCR.

31.21 If there is no one prefix but there is a zero prefix, then the zero prefix code index is obtained and the address in CDIEXP is again obtained. This new code index is indexed into the table again resulting with a new 2-word entry and a new type. There may not be a zero prefix code index in the entry. If there is not a zero prefix code index in the entry, it is due to the fact that zero prefixing is not allowed in this particular office, as indicated by an office option word.

31.22 If there is not a one prefix or a zero prefix, then the no prefix code index is obtained from the entry. The address of the CDIEXP is also obtained. The code index is then indexed into the table and a new 2-word entry and type are obtained.

31.23 The *normally no prefix is received* entry contains a type three, screening table number, and direct route index. This entry also contains a zero prefix code index and a one prefix code index. This entry is normally used when no prefix is received although code indexes are provided to allow or deny a zero or a one prefix.

31.24 If there is a one prefix, then the one prefix code index is obtained and it is indexed into the CDIEXP table for a new entry and a new type.

31.25 If there is not a one prefix but there is a zero prefix, then the zero prefix code index from this entry is obtained and it is indexed into the CDIEXP table for a new entry with a new type. This is, of course, dependent on the office option with respect to zero prefixing.

SECTION 233-151-150

31.26 If there is no prefix at all, the call is processed like a type one starting with the screening class number.

31.27 The *ignore a one prefix* entry contains a type four, screening table number, and a direct route index. This entry also contains a zero prefix code index. This entry allows or denies a zero prefix.

31.28 So if there is a zero prefix and a zero prefix code index, the code index is obtained and it is indexed into the CDIEXP table for a new entry and a new type.

31.29 If there is no zero prefix, then the entry is processed like a type one starting with the screening class number.

31.30 The *FAT required on next three digits* entry contains a type five and a FAT number ranging from one through three. This entry implies that the three digits dialed are for a different area code and that a new code index from a FAT is needed to route this type of call to the proper destination. Therefore, the FAT number is stored in the TCR and a return code of eight is returned implying to the calling program to call 3DIGIT again to process this call.

31.31 The *conflict between area code and office code* entry contains a type six, an area code index, and an office code index. If there are three digits in the TCR with or without a zero or one prefix, there is a need for seven or ten digits in the TCR in order to specify which code index in the entry is to be used. So the code index that was obtained from the FAT table is stored in the TCR and a return code of nine is returned.

31.32 The *preroute peg count desired* entry contains a type seven, a peg counter number, and a code index. The peg counter is incremented and a new code index is obtained and used to index the CDIEXP table for a new entry and a new type.

31.33 Subroutines 3DIGIT, 7DIGIT, 10DIGIT, and CODE_NDX all return the same data. The return codes are:

- (1) One—Intraoffice call

- (2) Two through five—7- and 10-digit interoffice calls both with overlap outpulsing permitted and with no overlap outpulsing

- (3) Six—Destination determined by the DEST code in the route index expansion entry

- (4) Seven—Interoffice call with outpulsed received digits

- (5) Eight—10-digit call with foreign area translation required

- (6) Nine—Conflicting office and area code.

For return codes one through seven, the first word of the route index expansion entry and the address are returned. For return codes one through seven, the TCR contains the toll diversion indicator, the charge index, and the route index. For return code eight, the TCR contains a FAT number and for return code nine, the TCR contains a code index.

C. Route Index Expansion (RI_EXP) Subroutine

31.34 The subroutine RI_EXP is called by subroutine 3DIGIT and is also called by other subroutines for routing information. Subroutine RI_EXP translates a route index to return one through seven return codes which are the same as 3DIGIT return codes one through seven. Also returned is the first word and address of the route index expansion entry.

31.35 After subroutine RI_EXP is called by subroutine 3DIGIT, the route index is obtained by RI_EXP from 3DIGIT. Subroutine RI_EXP then obtains the address of the route index expansion table from the RIEXP 3-word MTI entry. This table has a maximum of 512 words with 2-word entries. The route index is then indexed into the table and a 2-word entry is obtained. There are basically eight types of entries. They are:

- Type 0—Unassigned
- Type 1—Intraoffice (route index is the NOC)
- Types 2 and 3—10-digit interoffice (overlap outpulsing and no overlap outpulsing permitted)

- Types 4 and 5—7-digit interoffice (overlap outpulsing and no overlap outpulsing permitted)
- Type 6—Destination determined by the DEST code in the route index expansion entry
- Type 7—Interoffice (outpulse received digits).

31.36 The **type** number is returned as return code. Type 0 is for unassigned or error return. Type 1 for intraoffice contains a route index which is the NOC.

31.37 Types 2 through 5 for 7- and 10-digit interoffice with or without overlap outpulsing contain a possible three digits which are used to prefix the digits that are sent. The entry also contains a delete number for the number of digits to be deleted (if any), a free bit indicating if there is to be charging, the trunk group number for a trunk to be used, and an alternate route index provided (APR) bit to indicate that if all the members of the trunk group to be used are busy, then there is or is not another route index to use.

31.38 Type 6 contains a destination code number which indicates a particular destination. The numbers range from 0 through 20 and correspond to:

- 0—Vacant circuit group
- 1—Vacant code operator
- 2—No outpulsing trunk group
- 3—Tones with time-out
- 4—Tones without time-out
- 5—Announcement local
- 6—Route to reorder
- 7—Station ringer test
- 8—Code conversion
- 9—Balance test line
- 10—Interrupt milliwatt test line
- 11—Synchronous test line

- 12—Loop around test line 0
- 13—Loop around test line 1
- 14—Short circuit test line
- 15—Open circuit test line
- 16—Charge test line
- 17—Continuous milliwatt test line
- 18—Trunk and line test panel access trunk 1
- 19—Trunk and line test panel access trunk 2
- 20—Autoconnect
- 21—ROTL input terminal (for 3E3 generic only).

The entries are identical except for DEST code 8 which is code conversion that contains the table and entry numbers used to index into the expansion tables and a digit count. The valid numbers in digit count are four through seven which means a 2-word expansion entry. The other entries all contain a trunk group number which is optional, a free bit, a class of service bit, and a tone bit for high or low tone.

31.39 Type 7 is for interoffice with outpulse-received digits. Type 7 contains the same fields of data as types 2 through 5.

D. Get the Alternate Route Index (GARI) Subroutine

31.40 The subroutine GARI may be called by the OUTCAL program if OUTCAL is not able to select a member of the trunk indicated by the route index expansion entry. If there is an alternate route indicated in the RIEXP entry, then a new route index is selected.

31.41 The route index that was stored in the TCR by subroutine 3DIGIT is used by GARI. Subroutine GARI obtains the address of the alternate route index expansion table from the ARIEXP 3-word MTI entry. The route index is indexed into the alternate route index table. Then the alternate route index is returned.

SECTION 233-151-150

E. Charge Expansion (CH_EXP) Subroutine

31.42 The CH_EXP subroutine is called to get the charging data for the calling party. The charge index that was obtained from the screening table entry and stored in the TCR is now used.

31.43 The subroutine CH_EXP is entered with the charge index. The CHARGETB is a 16-word MTI table, with 1-word entries, which is directly indexed by the charge index. Subroutine CH_EXP is only called if the charge index is between 16 and 31. Also, only the four low bits of the index are used. The only other valid charge index is one, which implies a free call. Subroutine CH_EXP returns the type of charging (coin or message rate) and the charging data (initial period, initial charge, overtime period, overtime charge).

F. Country Code Digit (CCDIGIT) Subroutine

31.44 The CCDIGIT subroutine gets the country code digits from the TCR. A maximum of four digits are obtained. When four digits are obtained, the first digit is the second or third digit of the IDDD prefix digits. When three digits are obtained, the digits are the IDDD prefix digit 010. The IDDD prefix digits are 01 (TSPS operator assistance), 010 (overseas operator assistance), and 011 (station-to-station). The country code digit is from one to three digits. If 010 was obtained, the digits are right justified, then CCDIGIT calls CCEXP to obtain a valid entry. When four digits are obtained, all four digits may not be used in obtaining the valid terminal entry. For example, if CAC 47 is dialed, CCDIGIT obtained 147x from the TCR, where x is the first digit of the foreign national number and 147 translates into a valid terminal entry. The CCDIGIT subroutine is called by DNTRP in call processing and by TLTPC in trunk maintenance.

31.45 Subroutine CCDIGIT is called by the DNTRP program. The TCR address is obtained by CCDIGIT from DNTRP. The data stored in the TCR used by CCDIGIT are the digits dialed and an indicator as to the position of the last digit received (SIGDIG) in the TCR. CCDIGIT obtains the last digits, up to four, using SIGDIG and puts them in register 1.

31.46 A test is made for operator. When operator is dialed which is 010, A1A0 (in BCD) digits

are obtained and the operator test compares for A0 to match. If the digits dialed are A1A0, then the digits are rotated to right-adjust the high three digits. This positions the digits for the call to CCEXP. If the digits dialed are not operator, then the digits obtained should have a 1 in the first position with the variable country access code (CAC) in the next three positions. The digits are in the proper format for CCEXP. The dialed digits in register 1 are the entry condition for the call to CCEXP.

31.47 For call processing, CCEXP is the entry point into subroutine CCEXP. The CCEXP subroutine translates the four digits until a valid entry or a vacant entry is obtained. If a valid or vacant entry is not obtained after translating the four digits, then error is returned to CCDIGIT. If a valid or vacant entry is not obtained after translating the four digits, then error is returned to CCDIGIT. If a valid entry is obtained, then register 1 contains the maximum number of digits expected, the minimum number of digits expected, and the code index. Also, a successful return code is returned. If a vacant entry is obtained, then register 1 may contain a code index. If there is no code, the entry contains zeroes. Also, a successful return code is returned.

31.48 Subroutine CCDIGIT returns the data in register 1 with the successful return code to DNTRP if CCEXP was successful or returns no data in register 1 with the error return code to DNTRP.

G. Country Code Expansion (CCEXP, CCEXP) Subroutine

31.49 The CCEXP subroutine is called by the office record program to sequence through the country codes for the output of the 3307 form. The subroutine, given the binary number for the country code digits right justified in register 1, translates and returns the proper entry. CCEXP is the general purpose subroutine which translates the digits and returns the results. When CCEXP is called and the digit entered is zero, which is for overseas operator, the digit and the table numbers are both set to zero. The subroutine CCLST is called which indexes into the country code translator. The table 0 entry 0 slot is obtained and the data is returned to CCEXP. This entry may contain all zeroes (vacant) or a code index (invalid digits).

31.50 When CCEXP is called and a country code is entered, the digit or digits of the code are in binary. CCEXP has at most three digits to translate using the subroutine CCLST. Before CCEXP can translate the digits, they are converted from binary to BCD. The first digit is translated using Table 1. Table 0 is not used by CCEXP except for operator as previously mentioned. The first digit indexes into the table and the first 1-word entry is obtained. This entry may be any of the four types; vacant, invalid digit, next table, or valid terminal entry. If the entry type is vacant, invalid digit, or valid terminal entry, and there are still digits to translate, CCEXP returns the invalid digit return code and zeroes the entry data, indicating a code index of zero which is for vacant code treatment. If the entry type is vacant, invalid digit, or valid terminal entry and there are no more digits to translate, then the return code indicating the type of entry is returned with the data of the entry. If the entry is "next table", a check is made for another digit. If there is no other digit, then the entry and the next table return code is returned. If there is another digit, the next digit (second) is obtained and the table number from the first entry is obtained. This table number and digit are then translated and a new entry is obtained. This entry is handled the same as the first digit. If the second-digit entry contains a table number and there is a third digit, then the third digit is processed the same as the previous two digits.

31.51 The CCEXP subrountine, given the country code digits right justified in register 1, translates them and returns the proper entry. CCEXP is only called by CCDIGIT for call processing and maintenance, and it translates the digits to obtain a valid or invalid digit entry. When CCEXP is called and the digits entered are 010, which is for overseas operator, the digit and the table number are both set to zero. The subroutine CCLST is called which indexes into the country code translator. Table 0 entry 0 is obtained and the data is returned to CCEXP. The data in the entry is the same as described before. This is the same entry that CCEXP obtains when it is given a zero.

31.52 When CCEXP is called and a country code is entered, the code is preceded by a one and the digits are in BCD. To see how the digit one precedes the country code, see the description of the CCDIGIT subroutine (paragraph 31.44

through 31.48). CCEXP translates up to four digits using the subroutine CCLST. CCEXP indexes the first digit into table zero, obtaining entry one. This first entry is a next table entry containing table number one. The second digit indexes into table one. This digit is the first digit of the country code. The second entry is obtained; it may be any of the four types; vacant, invalid digit, next table, or valid terminal entry. If the entry is vacant, invalid digit, or valid terminal entry, the entry is returned with the return code indicating the type of entry. The remaining digits are assumed to be part of the national number and they are not translated. If the entry is "next table," a check is made for another digit. If there is not another digit, then the entry and the return code is returned. If there is another digit, the next digit is obtained and the table number from the second table entry is obtained. The table number and digit are then translated and the third entry is obtained. This entry is handled the same as the second entry. If the third entry contains a table number, then the fourth digit is processed the same as the previous two digits.

31.53 If IDDD is not allowed in the office and the IDDD prefix and country code are dialed, the proper data (one plus three digits) is given to CCEXP from CCDIGIT. CCEXP checks the table 0 entry 1 slot for all zeroes, which, when IDDD is not allowed, is zero. If the entry is zero then the table 0 entry 0 slot is returned. The data in this entry contains the code index for the appropriate action to be taken by call processing.

31.54 The subroutines called by CCEXP and CCEXP are BINXBCD, TENZERO, and CCLST.

H. Country Code List Translation (CCLST) Subroutine

31.55 The CCLST subroutine is called to obtain a 1-word entry of the table indicated from the country code translator. The country code translator contains up to a maximum of 28 tables with each table containing 10 words. The CCLST subroutine is only used in the 3E3 generic.

31.56 The country code table (CCTBL) is a 3-word MTI entry. The address of CCTBL is obtained. The country code table number which is from a previous entry (or if first time in it is zero) is indexed into the country code translator which obtains the 10-word table. The digit that

SECTION 233-151-150

was saved upon entry is then used to index into the table to obtain the 1-word entry. Subroutine CCLST returns the entry and its address.

31.57 The entry will contain a code index if the digit is invalid for that table, a table number for reentry if there are more digits to be translated or a maximum and a minimum number of digits expected, and a code index if a valid country code number has been translated.

31.58 The subroutine called by CCLST is GENTRY3.

I. One-Digit Table Translation (ODIGLST) Subroutine

31.59 The subroutine ODIGLST is called to specifically return the 1-word entry of the 1-digit table (1DGTBL) given the specific table number and entry number. The ODIGLST subroutine is only used in the 3E3 generic.

31.60 The subroutine ODIGLST saves the entry number and obtains the 1DGTBL MTI entry first word and address. The entry conditions for the call to GENTRY3 have been set up (registers 0, 1, 14, and 15). Subroutine GENTRY3 is called and the error return is checked. If there is an error, the error return code of zero is returned. If there is no error, the table is indexed with the entry number and the 1-word entry is obtained and put into register 1, with registers 14 and 15 being updated. The entry and the address are returned to the calling program with a return code of 1.

J. 1-Digit Translation (1DIGIT) Subroutine

31.61 The 1DIGIT subroutine is called when the data of an incoming trunk implies that a translation is needed on one digit. A 1-digit translation routes calls through the office. The necessary trunk data is a table selector number from the trunk group which is stored in the TCR. Also in the TCR are the digits received and the number of digits received. Subroutine 1DIGIT may be called more than once for a particular call if the data in the 1-digit entry proves to be inconclusive (ie, another table selector number in the entry).

31.62 The 1DIGTBL is a 3-word MTI entry. The address of 1DGTBL is obtained and the table selector number is then indexed into the tables obtaining a particular table. Each table consists of ten 1-word entries. There may be up

to 64 tables. The one digit most recently received is then indexed into this table and the 1-word entry is obtained. The entry contains a type, number of digits expected, and auxiliary information.

31.63 If type equals zero, then the digit received is invalid. The auxiliary information (AUXINFO) equals zero, but a route index is supplied so that the call can be completed to reorder tone or an operator. The number of digits expected (NDE) is zero.

31.64 If type equals one, then NDE is zero and AUXINFO equals a table selector number for another table to be used in translating the next digit.

31.65 If type equals two, then NDE equals the number of expected digits over the trunk and AUXINFO equals the NOC for this locally terminating call.

31.66 If type equals three, then this is a tandem call. The NDE is zero and the AUXINFO equals a route index.

31.67 The type is returned as a return code and the AUXINFO data is stored in the TCR.

K. Terminating Office Test Access Number List (TOTANLST) Subroutine

31.68 The subroutine TOTANLST is called when diagnostic tests on outgoing or 2-way trunks are being performed. This terminating office test access list or table contains basically a permanent busy test number and an operational test number.

31.69 The TOTANTBL is a 3-word MTI entry. The address of TOTANTBL is obtained and the terminating office test access number index from the group data is then indexed into the table obtaining a particular entry. The first word of the entry is returned.

31.70 The index of zero obtains the billing number for maintenance functions and up to three area codes. All other indexes obtain the previously mentioned data.

32. 4-DIGIT TRANSLATION (XSL4DG) PROGRAM

32.01 The 4-digit translator translates the last four digits of the dialed number into

terminating data. These digits are received from a distant office on an incoming call or from the dial pulse receiver on an intraoffice call.

32.02 The 4-digit translation (XSL4DG) program is based on 4DIGIT subroutine with a second entry point, 4DSCX. Also, a part of the 4-digit translation is the subroutine TERMXL, which is called by both 4DIGIT and 4DSCX.

32.03 The following paragraphs briefly describe the subroutines contained in XSL4DG program.

A. 4-Digit Subroutine Using Series Completion Directory Number (4DSCX)

32.04 The 4DSCX subroutine is called when a 4-digit translation is needed a second time. This type of call implies that the customer has a series completion directory number. Therefore, the NOC and the four BCD digits are in the registers when 4DSCX is called.

32.05 The 4DSCX entry into 4DIGIT subroutine converts the BCD ten to zero. If there are zeroes in the four BCD digits that were stored in the entry, they were stored as 1010 (binary 10) instead of 0000 (binary 0). Therefore, the digits need to be converted to zero. Subroutine 4DSCX then branches into 4DIGIT where subroutine TERMXL is called.

B. 4-Digit (4DIGIT) Subroutine

32.06 The subroutine 4DIGIT is called when the TCR contains the NOC, the stored digits, and the signal digit indicating the number of digits stored. The digits should be four BCD digits.

32.07 The 4DIGIT subroutine obtains the four BCD digits from the TCR by calling the subroutine DIGACC in the program OUTCAL. Subroutine 4DIGIT then calls subroutine TERMXL which returns the return code in register 0 and in register 1. The code index is for return code two, the PBX/MLHG and member number are for return code three, and the TEN is for return codes four through seven (see paragraph 32.10). The rest of the data of the directory number is stored in SCRATCH.

32.08 The subroutine 4DIGIT then checks for an error from the subroutine TERMXL. If

an error is returned, then subroutine 4DIGIT in turn returns an error to the calling program. If the error return code is one, word 8 in the TCR is zeroed and word 7 in the TCR contains a zero, one, or two which is the type of unassignment from register 1. Then, if the return code is two, the third word in SCRATCH is obtained and stored in word 8 of the TCR. If the return code is three through seven, the data in register 1 is stored in word 7 of the TCR. For a return code of three, the data is the PBX/MLHG and member numbers. For return codes four through seven, the data is the TEN.

32.09 Next word 3 of SCRATCH is obtained. In word 3, there is the route index (if any), the originating bit, the call waiting bit, the trace bit, and the terminating major class. The class is tested for autoconnect. If the class is for autoconnect, then what was stored as the route index is actually a table index. So the subroutine ACXREQ in program ACPORT is called, which obtains the table index and sets up for autoconnect using the table index, and then returns a route index into the route index field in SCRATCH. Word 3 of SCRATCH is again obtained, now containing a route index. Now, regardless of autoconnect, word 3 of SCRATCH is stored in word 8 of the TCR. Subroutines 4DIGIT and 4DSCX now return to the calling program.

32.10 Subroutines 4DIGIT and 4DSCX return the same data. These return codes are returned from the subroutine TERMXL. The return codes are:

- Zero—Error (proper translation data could not be found)
- One—Unassigned or blank number
- Two—Route to another office
- Three—PBX or multiline hunt group
- Four—Line has a series completion number
- Five—Line has a key scan point number
- Six—Line has both a series completion number and a key scan point number
- Seven—Normal line data.

32.11 The TCR contains a code index and a screening class number for a return code of two. Register 1 contains the code index also.

32.12 For a return code of three, the TCR contains the PBX/MLHG number, the first and last hunt member numbers, the terminating major class, a trace bit indicating if tracing is to be done, and an originating bit indicating if there is more data in the originating translation data for this line. Register 1 contains the PBX/MLHG number and the first hunt member number. If the line is denied termination, the TCR contains the route index and not the last hunt member number.

32.13 For a return code of four through seven, the TCR contains the TEN, the terminating major class, the route index, a trace bit, an originating bit, and a call waiting bit. Register 1 contains the TEN. For return code four, the series completion directory number is in SCRATCH. For return code five, the key scan point number and a tone bit is in SCRATCH (Fig. 37). For return code six, the key scan point number, tone bit, and series completion directory number is stored in SCRATCH (Fig. 38). Also in SCRATCH is the number group table entry and address, the thousand group table entry and address, the hundred group table entry address, and the second word of the entry for return codes three through seven. Also for return codes four through six, the first word of the hundred group table entry is in SCRATCH. For return code two, only the number group table entry and address is stored in SCRATCH.

C. Termination Translation No Precut (TXLNPC) Subroutine

32.14 The subroutine TXLNPC is a second entry point into subroutine TERMXL. Subroutine TXLNPC does not test for the precut condition for which TERMXL does test. To do this, subroutine TXLNPC sets the no precut register and then becomes part of subroutine TERMXL.

D. Termination Translation (TERMXL) Subroutine

32.15 The TERMXL subroutine zeroes the no precut register and then saves the NOC and the four BCD digits which are given in registers 0 and 1. The subroutine TERMXL calls subroutine CLR_16A in program XLSUB which zeroes the 16 words of SCRATCH. Next the cutover route

index register is zeroed. Then the no precut register is tested. If it is not zero, subroutine TXLNPC is called and the NOC is to be translated next. If it is zero, then subroutine TERMXL is called and the precut test is next. The PRECUT bit in the office option word is tested. If the PRECUT bit is set, the office is in the precut state. So given the route index, which is the NOC, the subroutine RI_EXP is called. It returns the first word of the route index entry which may contain a cutover route index and the CORI bit. If there is a cutover route index, then the bit is set. If the CORI bit is set, the cutover route index is isolated and tested. If the route index is between zero and seven, it is a new NOC and is stored in place of the old NOC. If the route index is greater than seven, it is a route index that will provide a special route for the call. The route index is saved for future use and a register is set to indicate that it is saved. The original NOC is then translated. If there is a new NOC, the new NOC is to be translated. If the PRECUT bit is not set, then the original NOC (as it was saved) is to be translated.

32.16 The NOCD table is the 24-word MTI table defined in the program TDATA, which is eight 3-word entries. Given the NOC (which is now put in register 0), the first BCD digit (the thousand digit which is now put in register 1), and the address of the NOCD table (which is now put in registers 14 and 15), the subroutine GNTRY in program XLSUB is called. The 3-word entry associated with the NOC is located. The address in the entry is obtained and the table, which is a number group table, is then indexed by the thousand digit. The 1-word number group table entry is returned in register 1 and the address is returned in registers 14 and 15. The type, which is one, is returned in register 0 unless GNTRY located a vacant NOC entry or had any other problems with the translation. Then a zero is returned in register 0 indicating an error.

32.17 If there is no error, the number group table entry is stored in word 8 of SCRATCH for generic SO-2 and word 9 of SCRATCH for generic 3E3 and address of the entry is stored in words 6 and 7 of SCRATCH for generic SO-2 and words 7 and 8 of SCRATCH for generic 3E3. The high bit (bit 15) in the entry is tested. If this bit (NG_ID) is set, this indicates that there is a store increment in the entry. If the bit is not set, then the next bit (bit 14) is tested. If this bit (RI_ID)

is set, this indicates that there is a code index in the entry. If the bit is not set, this indicates that the thousand group does not exist. In this case TERMXL returns to the calling program with a zero in register 0 to indicate unassigned thousand group.

32.18 When the RI_ID bit is set, the code index in the entry is stored in word 3 of SCRATCH (Fig. 39 for generic SO-2 and Fig. 40 for generic 3E3) and a return code of two is returned in register 0 indicating that the call is to be routed to another office.

32.19 When the NG_ID bit is set, the store increment is isolated, the hundred digit is obtained and isolated, and they are added together to give the index for the thousand group table. This index is then indexed from the number group table entry to obtain the thousand-digit table entry, which is one word. The address registers (14 and 15) are updated to point to the new entry. The thousand group table entry is stored in word 11 of SCRATCH (Fig. 41) and the address of the entry is stored in words 9 and 10 of SCRATCH for Issue 4 of the SO-2 generic. For the 3E3 generic the thousand group table entry is stored in word 12 of SCRATCH (Fig. 42) and the address of the entry is stored in words 10 and 11 of SCRATCH.

32.20 In Issue 4 of the SO-2 generic the high bit (bit 15) in the entry is tested. If this bit (TG_ID) is set, it indicates that the hundred group table exists. If it is not set, it indicates a nonexistent hundred group table which subroutine TERMXL returns to the calling program with a zero in register 0 to indicate unassigned. A one is returned in register 1 to indicate unassigned hundred group. But if the bit is set, then the entry contains the store increment that points to the hundred group table and the increment is isolated.

32.21 In the 3E3 generic the high bit (bit 15) in the entry is tested. If this bit (TG_ID) is set, it indicates that the hundred group table exists. If it is not set, the TG_RI bit (bit 14) in the entry is tested. If this bit is set, it indicates that there is a hundred group route index. The thousands group entry containing the route index is stored in word 3 of SCRATCH and a return code of 8 is returned in register 0 indicating that the call is to be routed on a hundred group basis out of the office. If the TG_RI bit is not set, it indicates a

nonexistent hundred group table which subroutine TERMXL returns to the calling program with a zero in register 0 to indicate unassigned. A one is returned in register 1 to indicate unassigned hundred group. If the TG_ID bit is set, the entry contains the store increment that points to the hundred group table and the increment is isolated.

32.22 The ten and unit digits are then converted to a binary number by the subroutine BCDXBIN in program CSYSUB. The binary number is then multiplied by two in Issue 4 of the SO-2 generic and multiplied by three in the 3E3 generic. This number is part of the index for the hundred group table which is a 2-word entry table in Issue 4 of the SO-2 generic and a 3-word entry table in the 3E3 generic. The number is added to the store increment, which is then added to the thousand digit table entry address to obtain the hundred group table entry. The address is updated. The first and second words in Issue 4 of the SO-2 generic and the first, second, and third words in the 3E3 generic of the entry are obtained, then the entry ID in the first word is tested for a vacant entry. If the entry is vacant, then both words are zeroed in Issue 4 of the SO-2 generic and the three words are zeroed in the 3E3 generic because the entry itself may contain data in it which are comment-oriented for office records. For generic 3E3 the third word is then stored in word 3 of SCRATCH regardless of the fact that the entry may be vacant. Also, the address of the entry is stored in words 4 and 5 of SCRATCH for generic SO-2 and words 5 and 6 of SCRATCH for generic 3E3.

32.23 As indicated before, the first word of the entry contains an ID. The ID is in the high three bits. The following are the possible entry IDs:

- 0—Vacant or unassigned
- 2—PBX or multiline hunt group
- 6—Normal line data
- 7—Expanded line data.

Now the ID is tested for expansion data. If there is expansion data, then the first word also contains the table and entry numbers for the expansion tables. This word is then stored in word 2 of SCRATCH. This word is in register 1, which is

the entry condition for the subroutine EXPAND which is in program XLSUB. Subroutine EXPAND uses the table and entry numbers to return the address of the entry indicated by the numbers in registers 14 and 15. Also, in register 1 is the first word of the entry and in register 0 is the return code of zero for error (the numbers were not valid or else there was a translator problem) or one for successfully obtaining the data. The entry returned may be either a 2- or 4-word entry. The first word of the entry also contains an entry ID in the high three bits. These entry IDs are:

- 3—The line has a series completion directory number.
- 4—The line has a key scan point number.
- 5—The line has both a series completion directory number and a key scan point number.

The entry ID for 5 is a 4-word entry and the entry IDs for 3 and 4 are 2-word entries.

32.24 Next the second word of the expansion entry is obtained and then stored in word 0 of SCRATCH. For an ID of 3, the second word contains the series completion directory number. For an ID of 4, the second word contains the key SPN. If this is a 4-word entry, the third word of the entry is obtained and stored in word 1 of SCRATCH. For an ID of 5, the second word contains key SPN, the third word contains the series completion directory number, and the fourth word of the entry contains no data.

32.25 The first word of the expansion entry contains the TEN which is now in register 1. If there was no expansion data, the TEN is in the first word of the hundred group table entry and therefore, it is also in register 1. It is at this point that the subroutine continues, regardless of whether there is or is not expansion data.

32.26 Now the ID is tested for the PBX/MLHG entry. If this entry is for PBX/MLHG, a one is added to the ID number and this is returned as a return code of three. If the entry is not PBX/MLHG, then the test is made for a cutover route index. If there is not a route index, the return code is created by adding one to the ID which may be from 3 through 6 and then subroutine TERMXL returns to the calling program.

32.27 If there is a cutover route index, then with the TEN in register 1, the subroutine PRECUTST in program XLSUB is called. This subroutine obtains the cutover status bit of the TEN. The bit is in register 0 in the same position as found in the bit map. But if the office was not in the precut state, then register 0 returns a zero to subroutine TERMXL and subroutine TERMXL continues and returns the saved ID plus one as the return code with register 1 still containing the TEN.

32.28 If the office is in the precut state, subroutine PRECUTST returns the status bit in register 0 and the TEN in register 1. Subroutine TERMXL then tests the bit to see if the line of the TEN has been cut over. If the line is not cut over, TERMXL returns the return code as described before. If the line has been cut over, the major class for special routing is stored in word 3 of SCRATCH along with the cutover route index. Then subroutine TERMXL returns the return code as described previously.

33. GLOSSARY

33.01 A glossary of terms is provided to aid in understanding this section.

Bylink Trunk—An incoming immediate start trunk from a step-by-step office.

Call Forwarding—When activated by a station user, this feature automatically routes calls intended for the user station line to another station line that has been designated.

Call Tracing—On both intraoffice and interoffice calls, provides originating line identification or incoming trunk identification on calls completed to a specified line or number.

Call Waiting—Permits a customer in the talking state of an established call to be notified of an incoming call. The customer may then elect to receive the second call, via switchhook flash, without abandoning the existing connection.

Delay Dial—The call office signals off-hook toward the calling end when in the idle condition (loop supervision). The called office remains in the off-hook condition until a register sender is attached and ready to receive pulses. At that time, the called office changes to an on-hook

condition which is a start dialing indication to the calling office.

Ground Start—Usually, lines that originate at a PBX or coin station, whereby the presence of ground provides a method of signaling for originations.

Expansion Table—Used to expand the memory used for a translation entry when extra storage space is needed.

Hot Line—A line with manual class of service for which the machine automatically dials preprogrammed 7- or 10-digit numbers (does not require an operator).

Immediate Start—The originating distant switching system (usually SXS) does not wait for a start signal before sending dial pulses. The terminating switching system must be ready to receive pulsing in less than 65 ms.

Inband Coin and Rering—Coin collect, coin return, and rering signaled by a wink, followed by multifrequency tones.

Line—Anything that connects to a network terminal that is not classified as a trunk or service circuit. Usually, a pair of wires that serve to connect a customer telephone to a terminal on the network.

Local Coin Overtime—This feature is used when local coin overtime charging is provided.

Long Loop Pulsing—Pulsing that requires additional equipment to boost the signal to the terminal.

Make Busy—Permits the business customer to make lines appear busy to the central office for any subsequent terminating calls.

Member—One of a number of lines or trunks in a group.

Message Rate—Permits completion of calls to a group of unselected single and multimessage unit destinations which are assigned initial and overtime message periods with charges made accordingly.

Multiline Hunt—Allows calls to hunt over a preselected group of customer facilities in order to connect a calling party with an idle facility within the group.

Network—The fabric consisting of network links and switches that serves to provide a 2-wire metallic talking path between any two network terminals. Remreed switches are used in No. 3 ESS.

No Hunt—A switching machine should be capable of associating directory numbers with individual facilities which may also be part of a multiline hunt group. Calls to these directory numbers should not cause hunting to occur.

No-Test Access—A trunk is bridged onto an existing connection to enable an operator or maintenance personnel to complete calls to facilities which may be busy.

Outpulsing—Generation of pulses to match the stored digit information and of the proper type to be used by the next switching office.

Peg Counter—Used to count the number of times a particular event occurs.

QZ Billing—Toll calls from lines having this feature are routed to a CAMA operator for identification.

Recorded Announcement—One or more small magnetic recorders and associated amplifiers that produce previously recorded voice messages.

Remote Make Busy (RMB)—A preselected set of members associated with an RMB key is made to appear busy when the RMB is operated.

Scan Point—One of many current sensing elements that form the scanner. It is used to detect the on-hook or off-hook status of lines, to supervise talking paths, and to monitor many other miscellaneous points. The ferrod is the scan point device used in No. 3 ESS.

Series Completion—Allows calls to be routed to any designated directory number within the same office code if the original number is busy.

Service Circuit (SVC)—A circuit that connects to a network terminal and provides one of several specialized functions such as digit reception and transmission, alerting coin control, testing, and others. A service circuit communicates with the system control via scan points and peripheral decoders.

SECTION 233-151-150

Service Observing—Means of determining the quality of service given by a telephone office.

Sleeve Lead—Third wire (in addition to tip and ring) used for line service observing or busy indication.

Speed Calling—Permits a station user to originate calls to frequently called numbers by dialing predesignated codes consisting of fewer digits than normally required. Speed call lists may contain 8 or 30 entries. Optionally, individual entries may be changed by the subscriber.

Stop-Go—A momentary off-hook signal, given during an interdigital interval to indicate that dial pulsing should stop until a transition from off-hook to on-hook, or “go” signal, is returned.

Subtranslator—A table to supplement a translator table when more space is needed.

TEN—The network terminal equipment number of a line or trunk.

Terminal—A point at which information can enter or leave a communication network.

Three-Way Calling—A station user can add a third party to any established call for a 3-party conference without the assistance of an operator.

Translation—Conversion of information from one form to another. Specifically, the translation of digits received to those required to make a call.

Translator—A group of tables containing data pertinent to a specific type of translation.

Trunk—The circuits and transmission facility that connect a network terminal of one office to a network terminal of another. A trunk circuit communicates with the system control via scan points and peripheral decoders. In direction of control, trunks are classified as incoming, outgoing, or 2-way. Trunks may also be classified by method of supervision (loop, reverse battery, E- and M-lead, etc) and signaling language (multifrequency and dial pulse). Some special trunks terminate on switchboards and test circuits.

Trunk and Line Test Panel (TLTP)—A manual test facility for performing various

transmission and operational tests on trunks, service circuits, lines, and junctors.

Wink Start—A momentary off-hook signal sent from the terminating office to the originating office as an indication that outpulsing may begin at the end of the timed off-hook wink.

34. ABBREVIATIONS

34.01 The following abbreviations and acronyms are used frequently in this section.

TERM	MEANING
ABCK	All Busy Check
ALIT	Automatic Line Insulation Test
ADO	Automatic Dial Out Data
ANN_SPN	Announcement Scan Point Number
ARI	Alternate Route Index
AUTO_CB	Autoconnect Callback Block
AUXINFO	Auxiliary Information
BCD	Binary Coded Decimal
BUSYSVC	Busy Service Circuit
BUSYCKT	Busy Circuit
CCEXP	Country Code Expansion
CCLST	Country Code List Translation
CCTBL	Country Code Table
CDIEXP	Code Index Expansion
CDPR	Customer Dial Pulse Receiver
CF_EXP	Call Forwarding Expansion
CH_EXP	Charge Expansion
CHI	Charge Index
CI	Code Index
CKTSCAN	Circuit Scan

TERM	MEANING	TERM	MEANING
CKTLST	Circuit List	ESS	Electronic Switching System
CLASSA	Class A Line	EXPAND	Expand
CLFRD	Call Forwarding	EXPAND_C	Expand
CLR_16A	Clear 16 Words	EXPTBL	Expansion Translation Table
CNI_EXP	Coin Triplet Index Expansion	FAT	Foreign Area Translator
CODE_NDX	Code Index	FATTBL	Foreign Area Translator Table
COINEN	Release a Coin Line Circuit (Second Entry Point)	FETL	Far End Test Lines For ROTL
COINOPR	Operate a Coin Line Circuit	FRMCHK	Frame Check
COINRL	Release a Coin Line Circuit	GARI	Get the Alternate Route Index
COINSTAT	Coin Status	GETBIL	Get Billing Number
COINTRIP	Coin Triplet	GET_DSPN	Get Directed Scan Point Number
DATAPR	Data Processing	GET_DTA	Get Distributor Triplet Address
DATCKT	Data Circuit	GETGRP	Group and Member Numbers
DATCKTG	Data Circuit Group	GET_SSPN	Get Supervisory Scan Point Number
DIR_SPN	Directed Scan Point Number	GET_TN	Get Telephone Number
DISCON	Disconnect Progress Marks	GENTRY	Get Entry
DNTRP	Digit Interpretation Progress Marks	GENTRY3	Get Entry
DP	Dial Pulse	GRPADR	Group Address
DRI	Direct Route Index	GRPTBL	Group Table
DSTRBOR	Distribute Order Routine	GRPX	Group Translation
DTA	Distributor Triplet Address	ID	Identification
DTDADTA	Dial Tone Detector Alarm Data	IDLECKT	Idle Circuit
EL	Essential Line	IDLESVC	Idle Circuit
ENTRY_ID	Entry Identification Code	LCC	Line Class Code
EQPSEL	Equipment Selection Subroutines	LINESTATUS	Line Status Bits Test and Manipulation
ESI	Entry Size Indicator	LMN	Largest Member Number

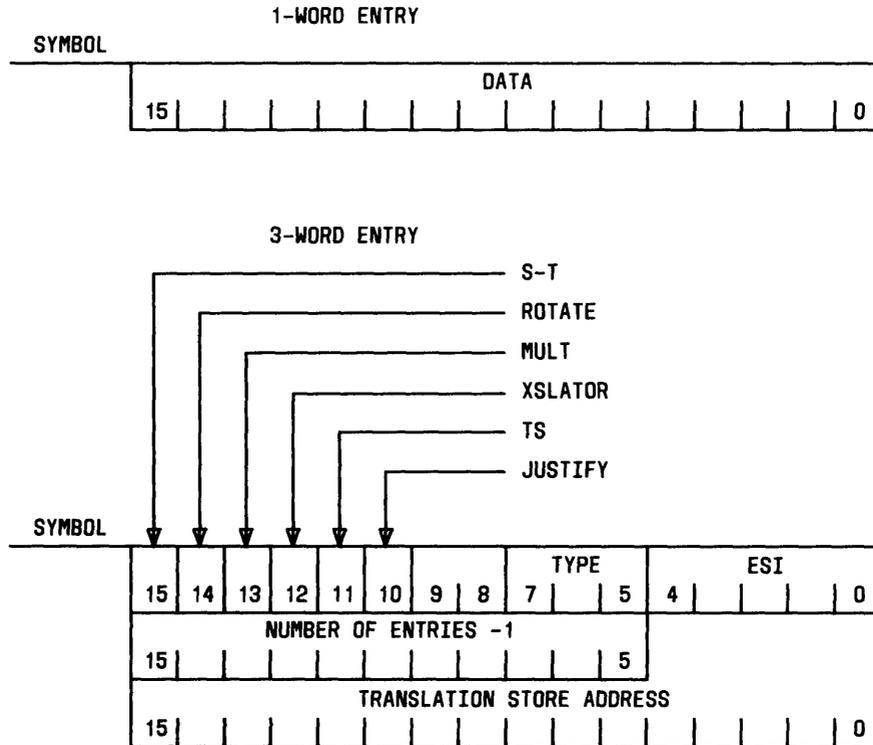
SECTION 233-151-150

TERM	MEANING	TERM	MEANING
LNDATA	Line Data	OI	Operator Identified
LNORIG	Line Origination Program	OMAJ	Originating Major Class
LOCALNPA	Local Numbering Plan Area	ORGXLA	Originating Line Translation
MAJ	Major	PBX	Private Branch Exchange
MBJ	Message Billing Index	PBX_BUSY	PBX Busy
MBRDATA	Member Data	PBXDAT	PBX Data
MBR_LIST	Member List	PBX_IDLE	PBX Idle
MBRSEL	Member Select	PBX/MLHG	PBX Multiline Hunt Group
MCNO	Message Class Number	PKEYSTAT	PBX Key Status
MEMLST	Member List Table	PCUTMAP	Precut Map
MF	Multifrequency	PORTBUSY	Port Busy
MLHG	Multiline Hunt Group	PORTIDLE	Port Idle
MR_CH	Message Register Charge	PORTSTAT	Port Status
MSGCL	Message Class Table	PORTTEST	Port Test
MTI	Master Table Index	PRECUTST	Precut Status
NDB	No Dial Back Information	RC	Recent Change
NDE	Number of Digits to Expect	RI	Route Index
NEXT_SVC	Next Service Circuit	RI_EXP	Route Index Expansion
NEXT_TRK	Next Trunk Circuit	ROH	Receiver Off-Hook
NOC	Normalized Office Code	ROTL	Remote Office Test Line
NOCBCD	Normalized Office Code to Office Code Conversion Table	RSCBS	ROTL Security Callback Status
NOC_BCD	Normalized Office Code to Binary Coded Decimal	SC	Speed Call
NPA	Numbering Plan Area	SC_EXP	Speed Call Expansion
NUWKMEMS	Number of Working Members	SCR	Screen Class Number
NXX_NOC	Convert Office Code to Normalized Office Code	SCRCLASS	Screening Class Number
ODIGLST	1-Digit Table Translation	SCR_TBL	Screening Table Number
		SELECTS	Select an Idle Circuit

TERM	MEANING	TERM	MEANING
SIGDIG	Signal Digit	TLTP	Trunk and Line Test Panel
SO	Service Observing	TMAJ	Terminating Major Class
SPDCL	Speed Call	TMR	Terminal Call Record
SPN	Scan Point Number	TN_SPN	Tone Present Scan Point Number Conversion to First Tone Scan Point Number
SPNXLA	Scan Point Number Translation	TN_UPTN	Telephone Number to Unpack Telephone Number
STATUS_A	Selection Status Block Address	TOTANLST	Terminating Office Test Access Number List
STATUS_G	Get Selection Status Block With Group Number	TOTANTBL	Terminating Office Test Access Number Table
STATUS_S	Get Selection Status Block With Scan Point Number	TRKSCAN	Trunk Scan
SVCKKT	Service Circuit	TRKSEL	Trunk and Service Circuit Select
SV_TEN	Service Circuit Number Conversion to Terminal Equipment Number	TT	TOUCH-TONE
TCR	Transient Call Record	TUTTL	Trunk Under Test Terminal for ROTL
TCRSEL	Transient Call Record Selection	TXLNPC	Termination Translation No Precut
TDATA	Translation Data	UPK_DN	Unpack Directoy Number
TDV	Toll Diversion	XSLSPN	Scan Point Number Translation Program
TEN	Terminal Equipment Number	XSLSUB	Basic Translation Subroutines
TENDATA	Terminal Equipment Number Data	XSL3DG	3-Digit Translation Program
TENXLA	Terminal Equipment Number Translation	XSL4DG	4-Digit Translation Program
TERMXL	Termination Translation	1DGTBL	One-Digit Table
TESTCKT	Test Circuit	1DIGIT	One Digit
TESTSVC	Test Service Circuit	3A CC	3A Central Control
TGN	Trunk Group Number	3DIGIT	Three Digit
TKORIG	Trunk Originations Program	4DIGIT	Four Digit
TK_TEN	Trunk or Service Circuit Scan Point Number Conversion to Terminal Equipment Number		

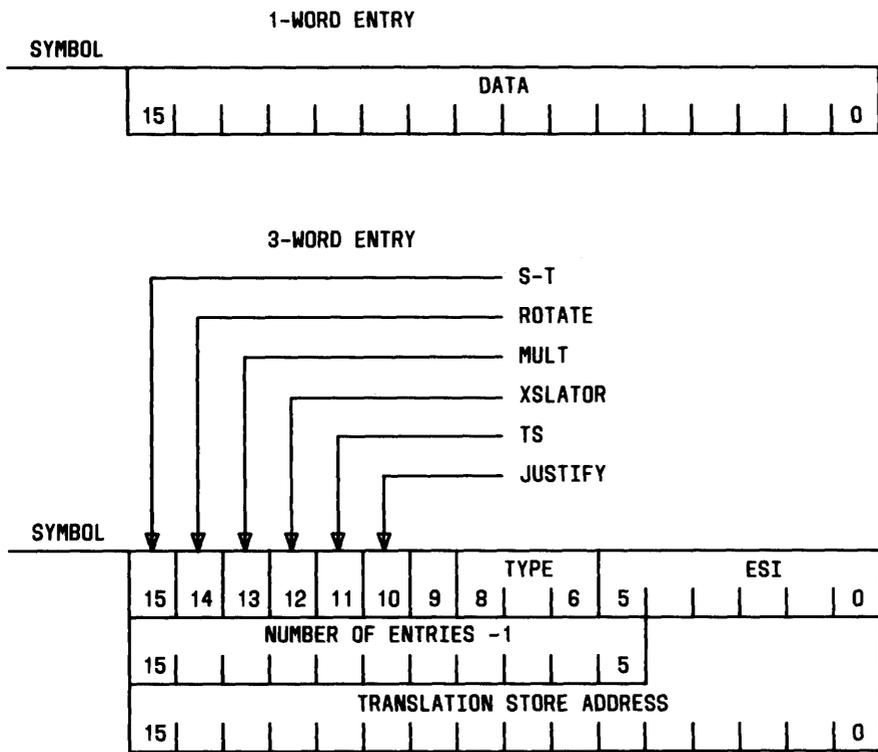
SECTION 233-151-150

TERM	MEANING	TERM	MEANING
4DSCX	4-Digit Subroutine Using Series Completion Directory	7DIGIT	Seven Digit
		10DIGIT	Ten Digit



- ESI = ENTRY SIZE INDICATOR (SEE MULT)
- TYPE = TYPE OF TRANSLATION TABLE IF THERE IS ONLY ONE TYPE. TYPE = 1
- JUSTIFY = USED WHEN ROTATE=1
 - = 0 DO NOT RIGHT JUSTIFY THE ENTRY
 - = 1 RIGHT JUSTIFY THE ENTRY
- TS = 0 ADDRESS POINTS TO PROGRAM STORE
 - = 1 ADDRESS POINTS TO TEMPORARY STORE
- XSLATOR = 0 TRANSLATOR IS NOT DEFINED
 - = 1 TRANSLATOR EXISTS
- MULT = 0 IF ROTATE=0 THEN ESI = LOG₂ (NO. OF WORDS PER ENTRY)
 - = 1 THEN ESI = LOG₂ (NO. OF ENTRIES PER WORD)
 - 1 MULTIPLY - ESI = NO. WORDS PER TABLE ENTRY - 2
- ROTATE = (SEE MULT)
- S-T = 0 TRANSLATION TABLE - ESI = (SEE MULT)
 - = 1 SEARCH TABLE - ESI = NO. OF WORDS PER TABLE ENTRY

Fig. 1—Master Table Index Entry Formats for Generic SO-2



- ESI = ENTRY SIZE INDICATOR (SEE MULT)
- TYPE = TYPE OF TRANSLATION TABLE IF THERE IS ONLY ONE TYPE. TYPE = 1
- JUSTIFY = USED WHEN ROTATE=1
 - = 0 DO NOT RIGHT JUSTIFY THE ENTRY
 - = 1 RIGHT JUSTIFY THE ENTRY
- TS = 0 ADDRESS POINTS TO PROGRAM STORE
 - = 1 ADDRESS POINTS TO TEMPORARY STORE
- XSLATOR = 0 TRANSLATOR IS NOT DEFINED
 - = 1 TRANSLATOR EXISTS
- MULT = 0 IF ROTATE=0 THEN ESI = LOG₂ (NO. OF WORDS PER ENTRY)
 - = 1 THEN ESI = LOG₂ (NO. OF ENTRIES PER WORD)
 - 1 MULTIPLY - ESI = NO. WORDS PER TABLE ENTRY - 2
- ROTATE = (SEE MULT)
- S-T = 0 TRANSLATION TABLE - ESI = (SEE MULT)
 - = 1 SEARCH TABLE - ESI = NO. OF WORDS PER TABLE ENTRY

Fig. 2—Master Table Index Entry Formats for Generic 3E3

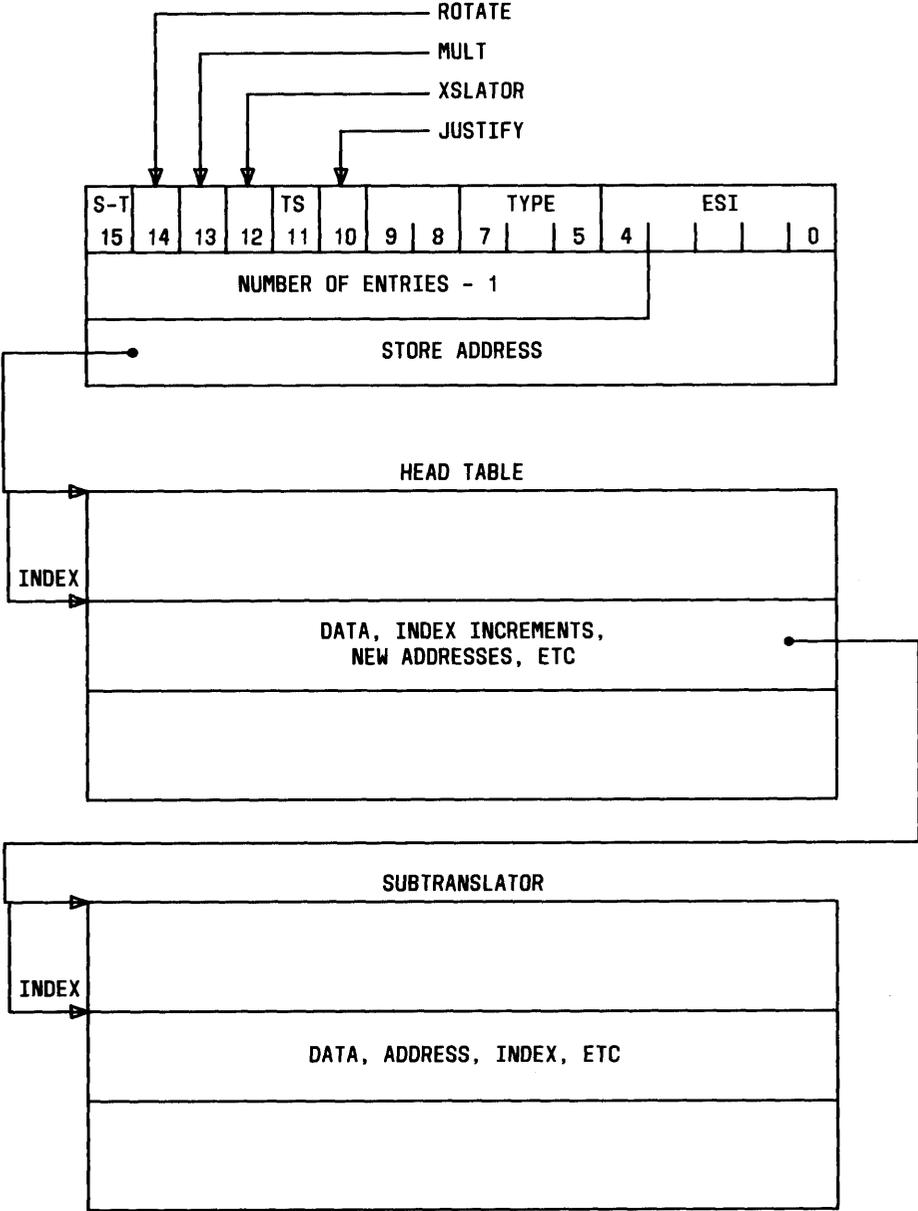


Fig. 3—Translator Accessing for Generic SO-2

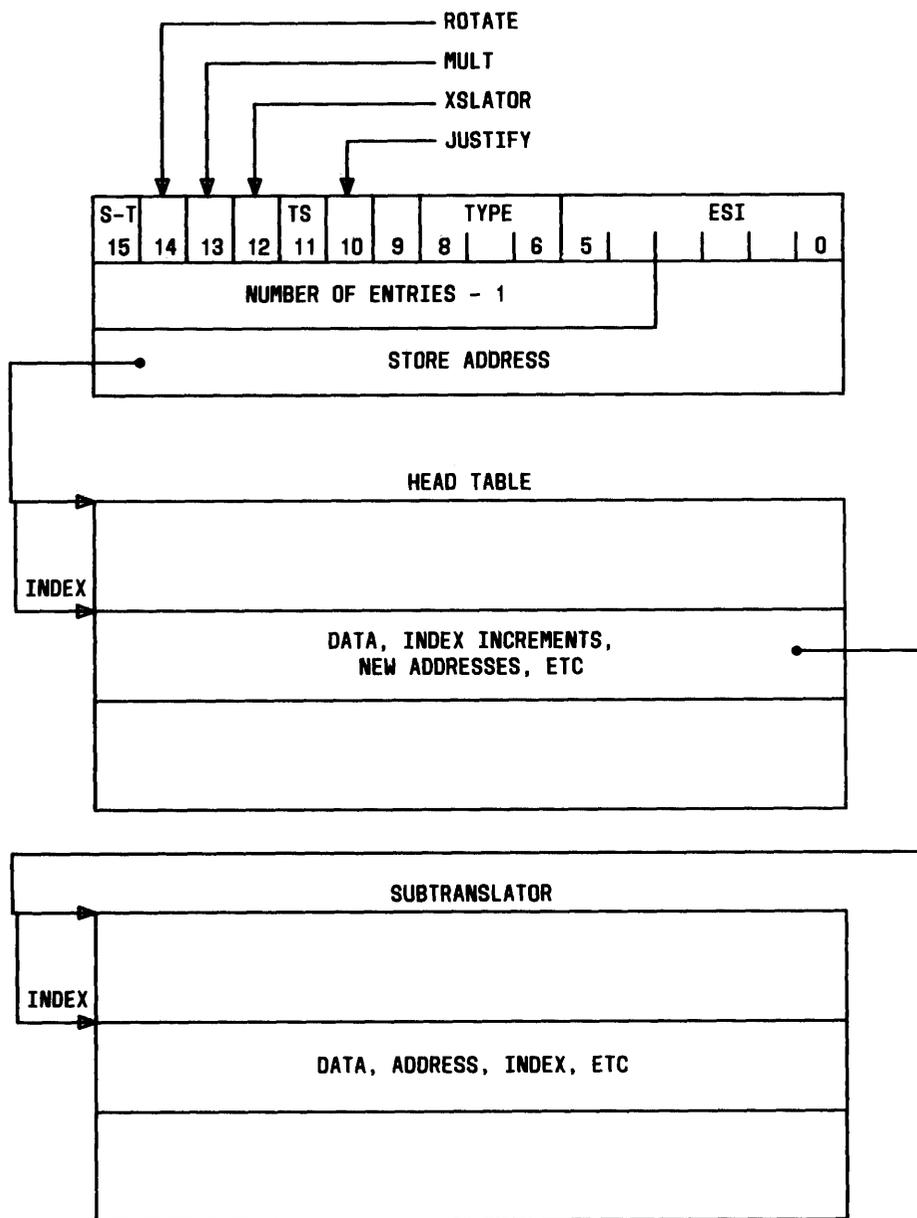


Fig. 4—Translator Accessing for Generic 3E3

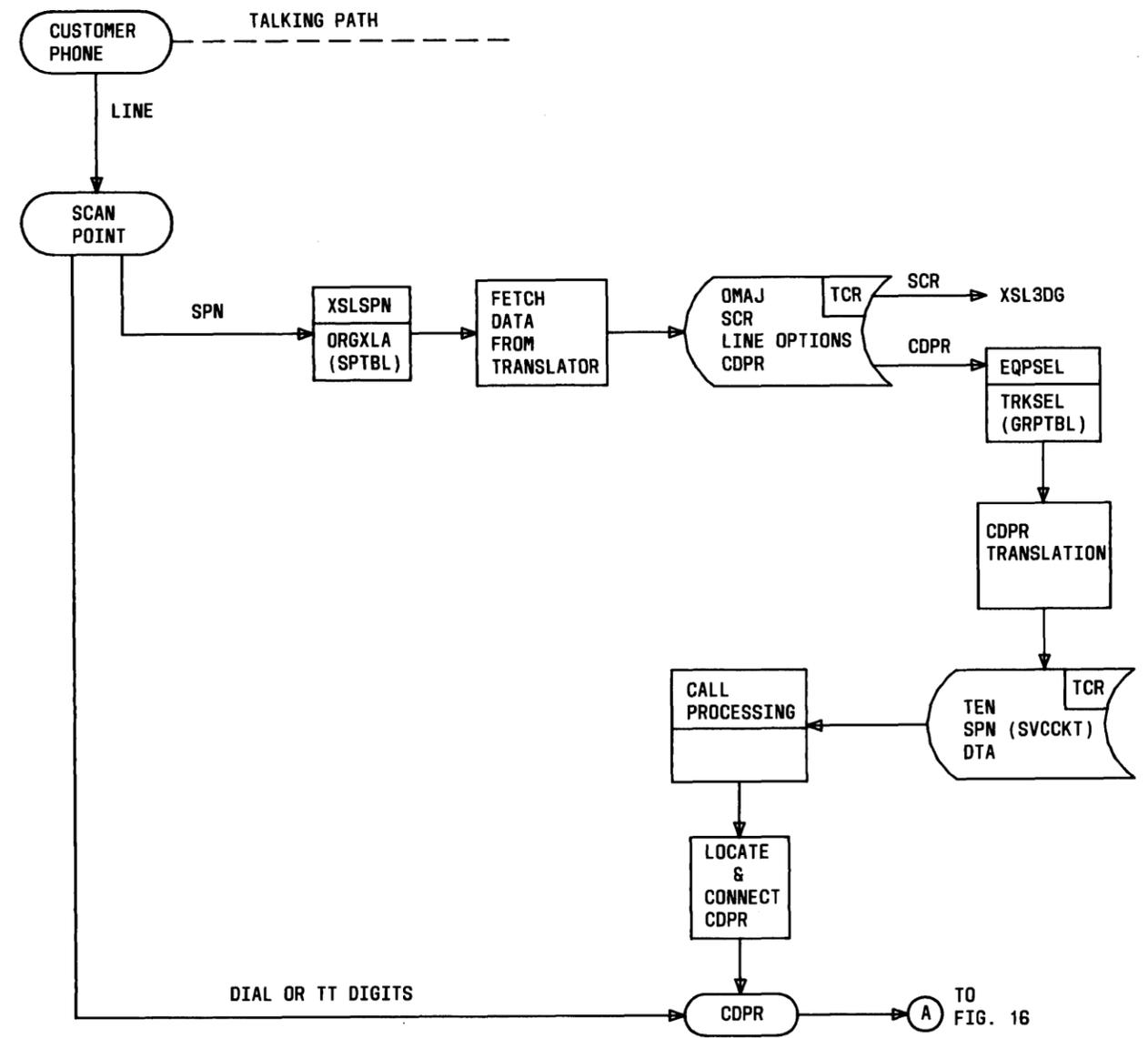


Fig. 5—Line Origination Translations

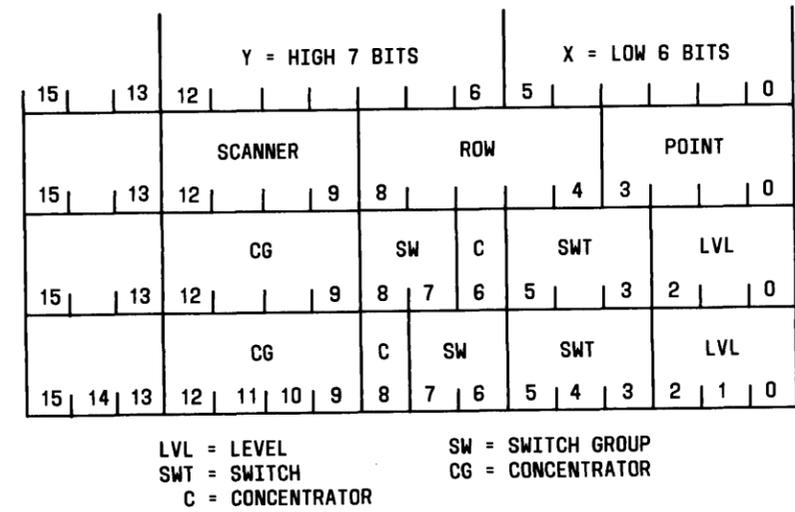


Fig. 6—SPN, TEN, and OE Formats

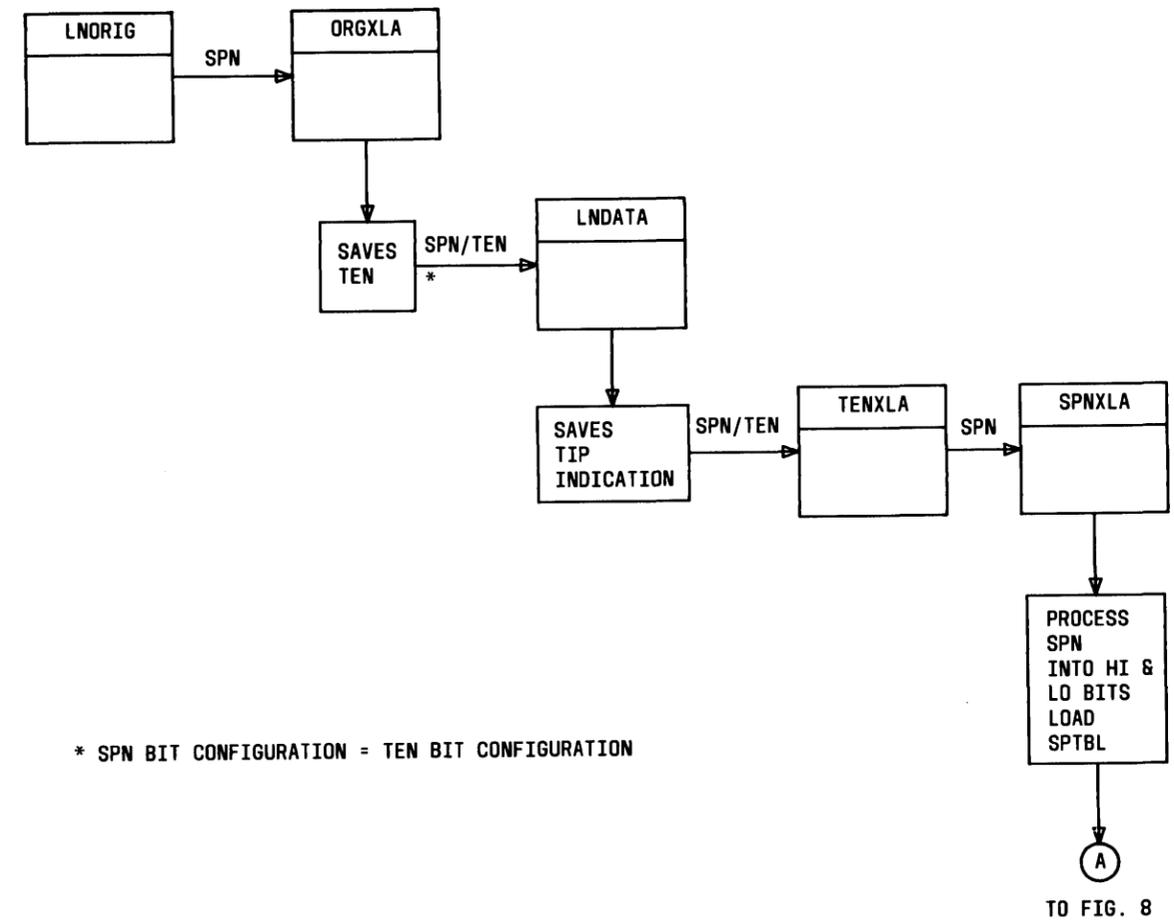


Fig. 7—SPN Processing

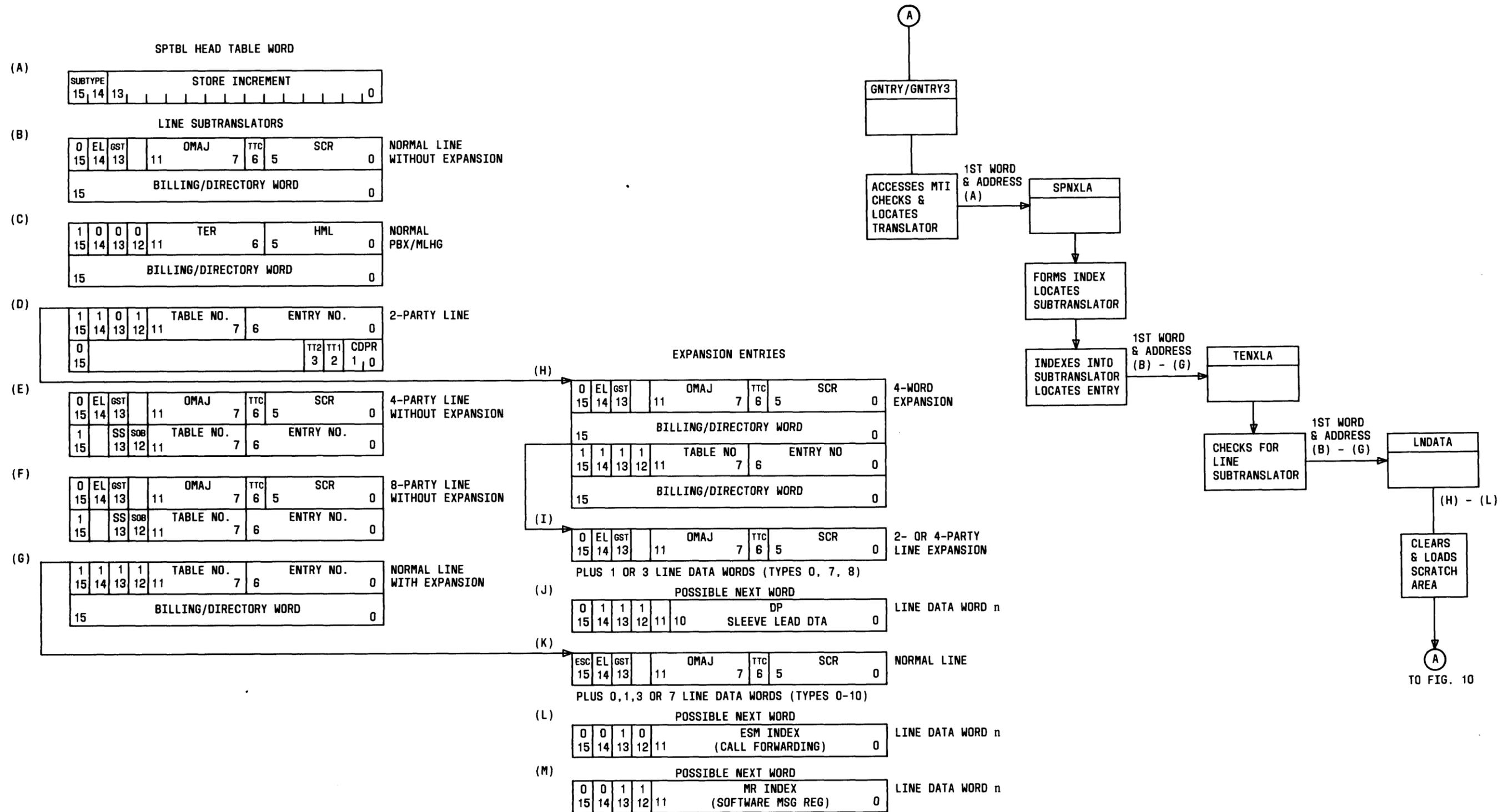


Fig. 8—Line Translation Processing for Generic SO-2

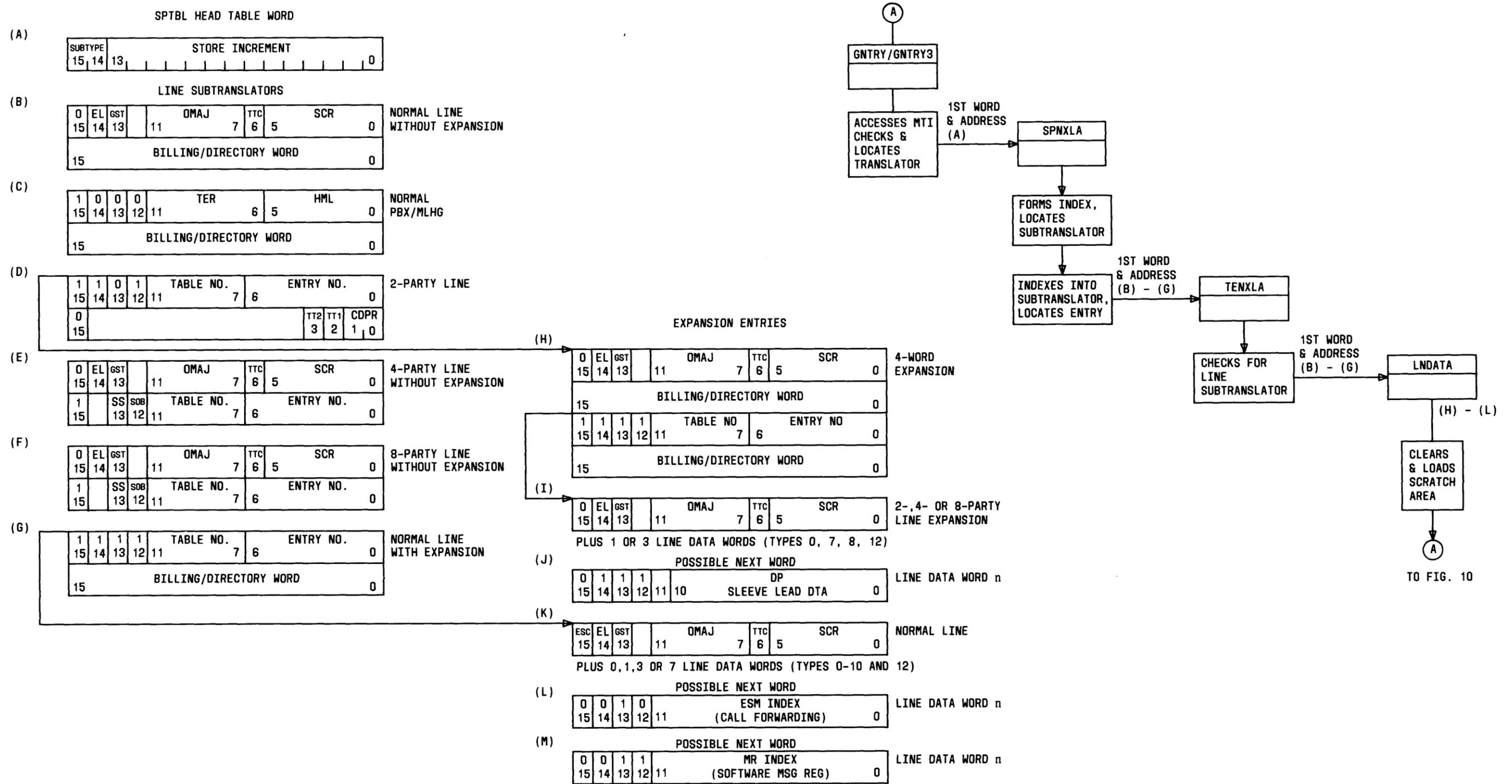
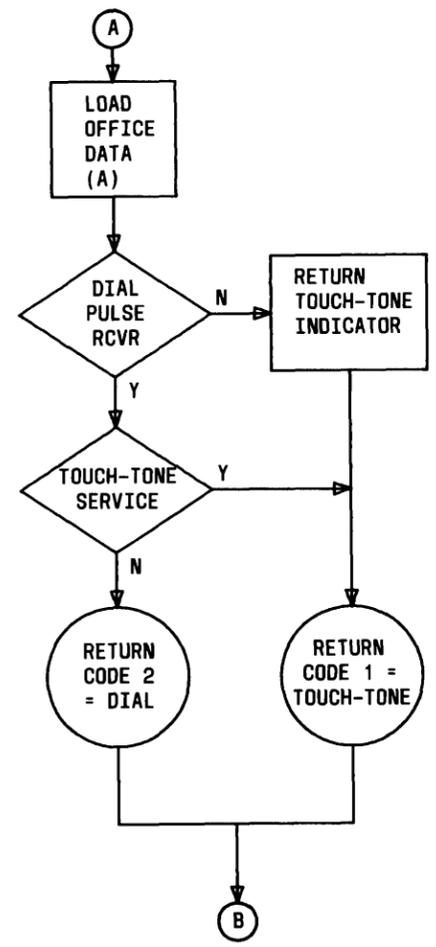
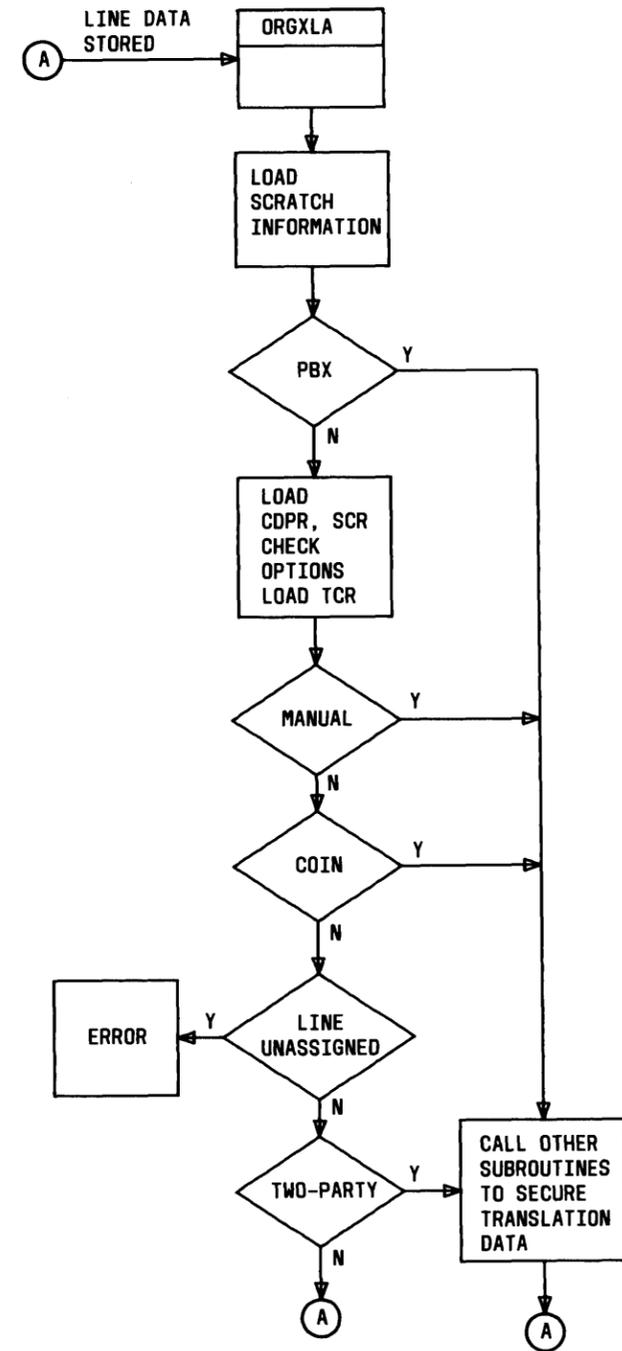


Fig. 9—Line Translation Processing for Generic 3E3



TO FIG. 11

(A)

OFFICE OPTION WORD FOR GENERIC SO-2															
DPG	MOT	SUP	ZPLUS												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

OFFICE OPTION WORD FOR GENERIC 3E3																
DPG	MOT	SUP	ZPLUS													ICF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

DPG
1 = DIAL PULSE RCVR PROVIDED

Fig. 10—Line Data Processing

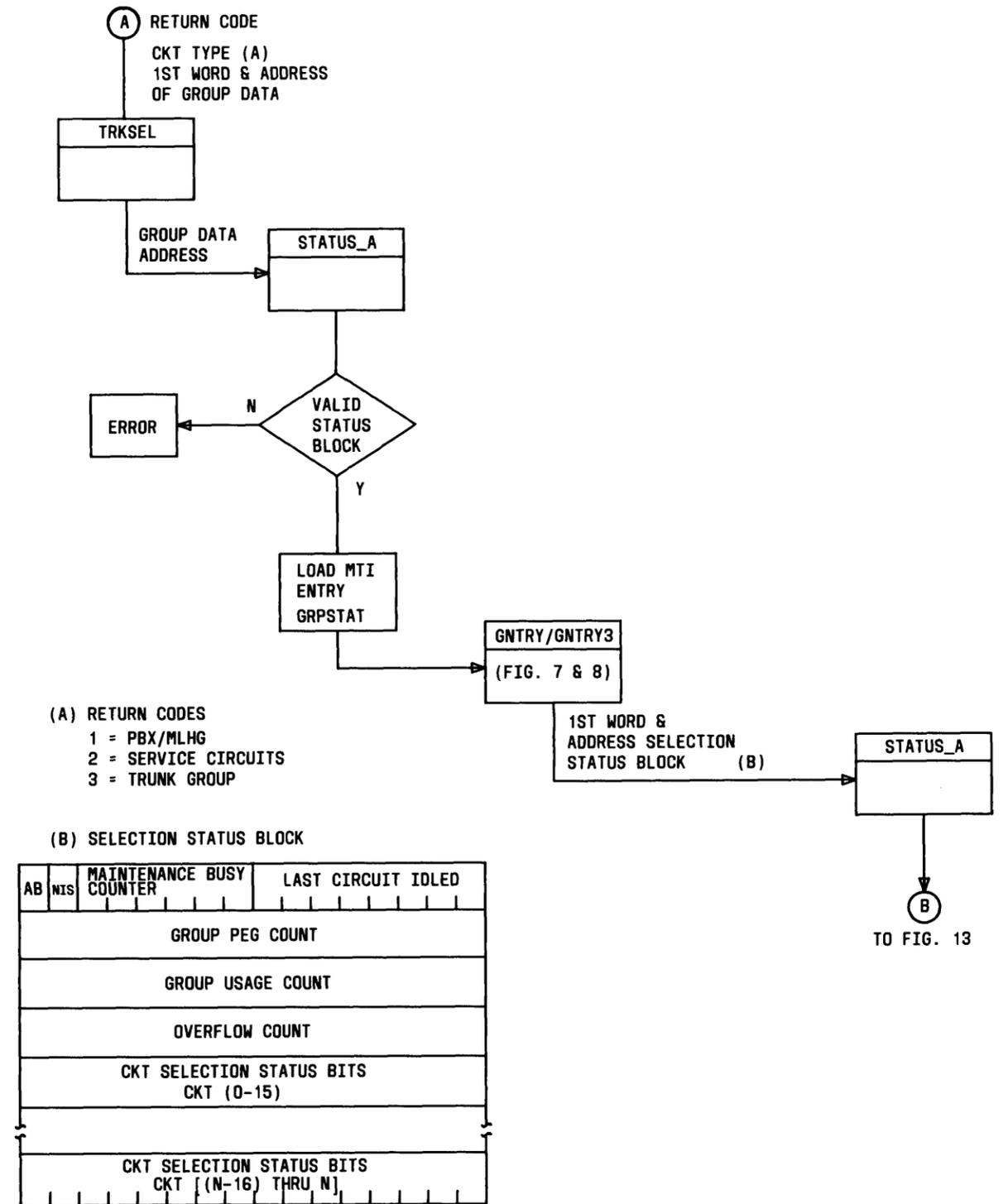
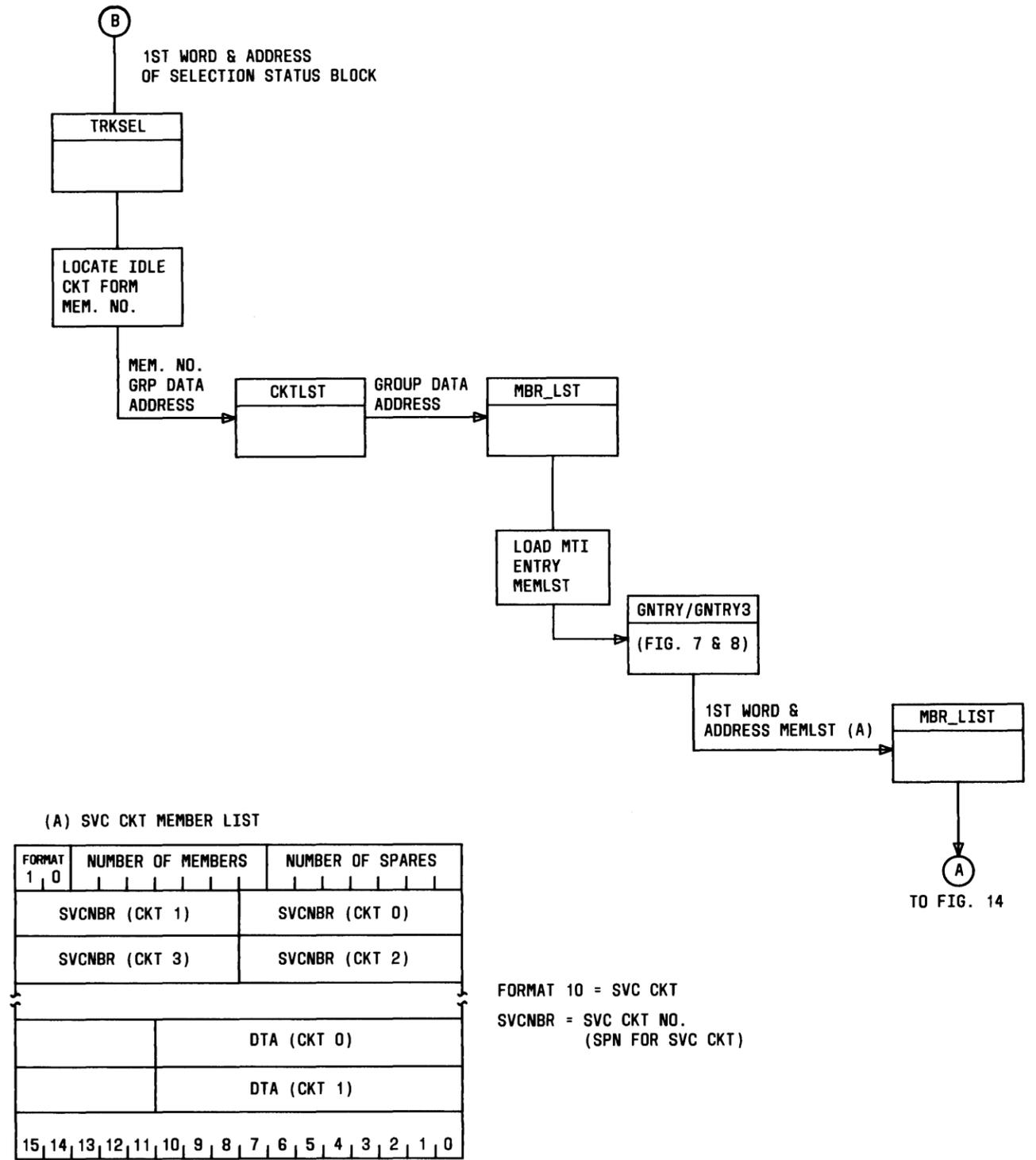


Fig. 12—Selection Status



(A) SVC CKT MEMBER LIST

FORMAT 1, 0	NUMBER OF MEMBERS	NUMBER OF SPARES
	SVCNBR (CKT 1)	SVCNBR (CKT 0)
	SVCNBR (CKT 3)	SVCNBR (CKT 2)
	DTA (CKT 0)	
	DTA (CKT 1)	
	15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0	

FORMAT 10 = SVC CKT
 SVCNBR = SVC CKT NO.
 (SPN FOR SVC CKT)

Fig. 13—MEMLIST Access

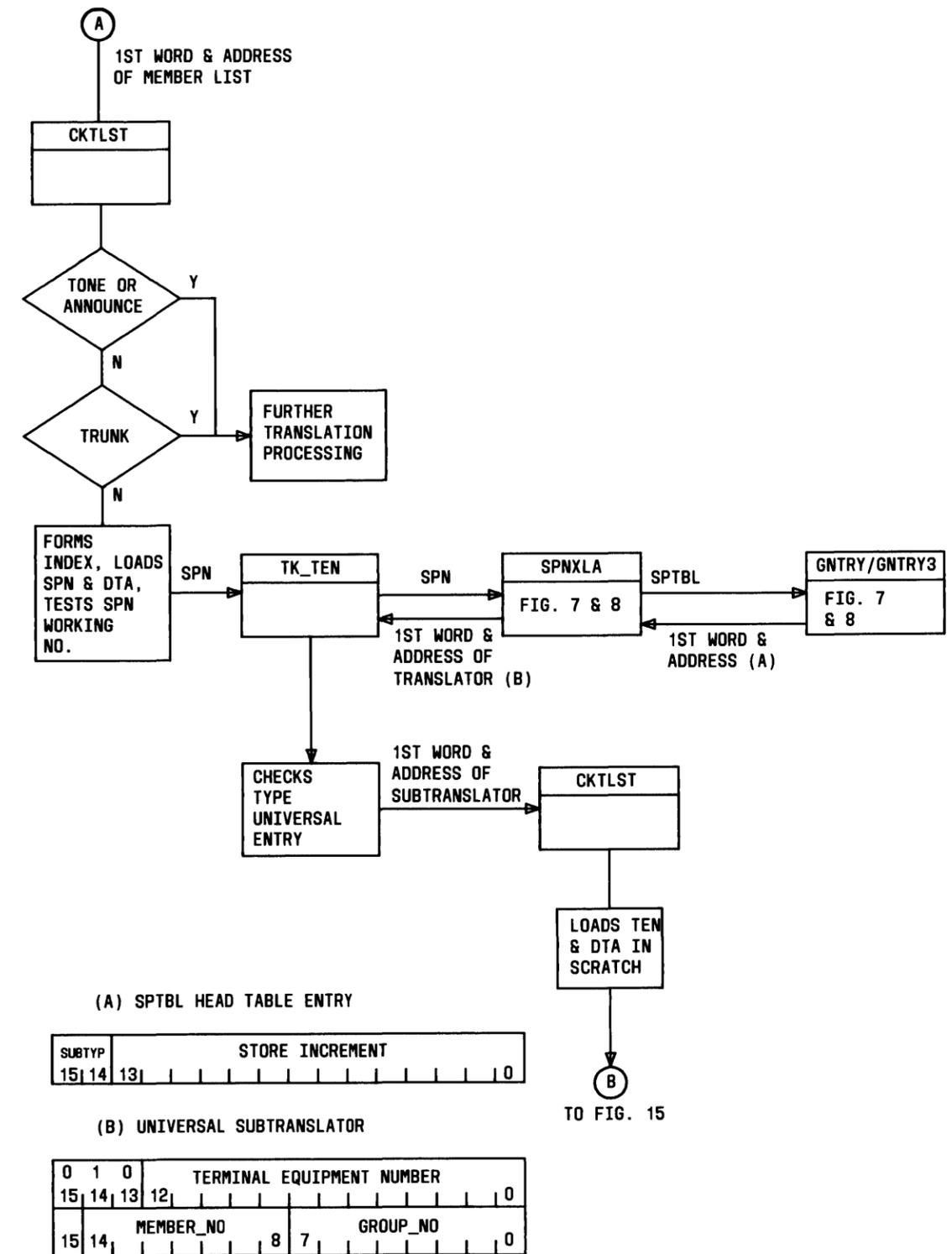
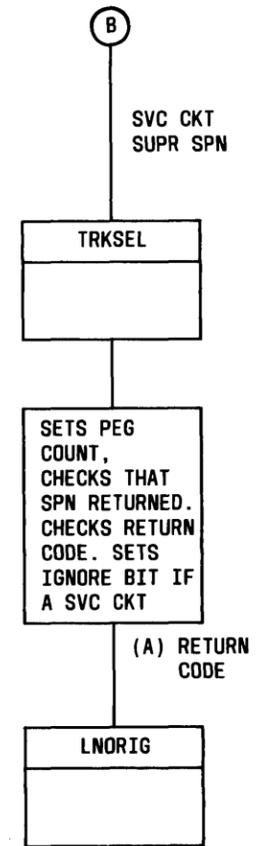


Fig. 14—SPN to TEN Conversion



(A) RETURN CODE: 0 = ERROR
1 = PBX/MLHG
2 = SVC CKT
3 = TRUNK

Fig. 15—Determination of Trunk or Service Circuit

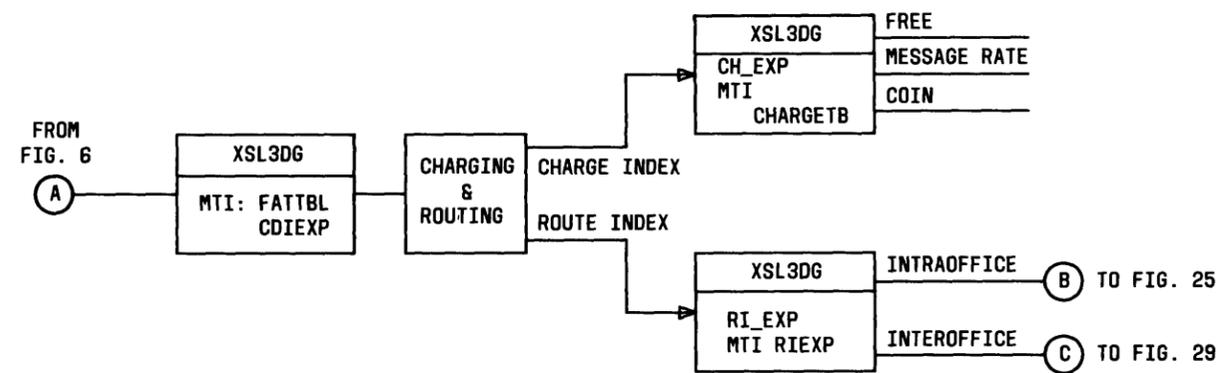


Fig. 16—3-Digit Translations

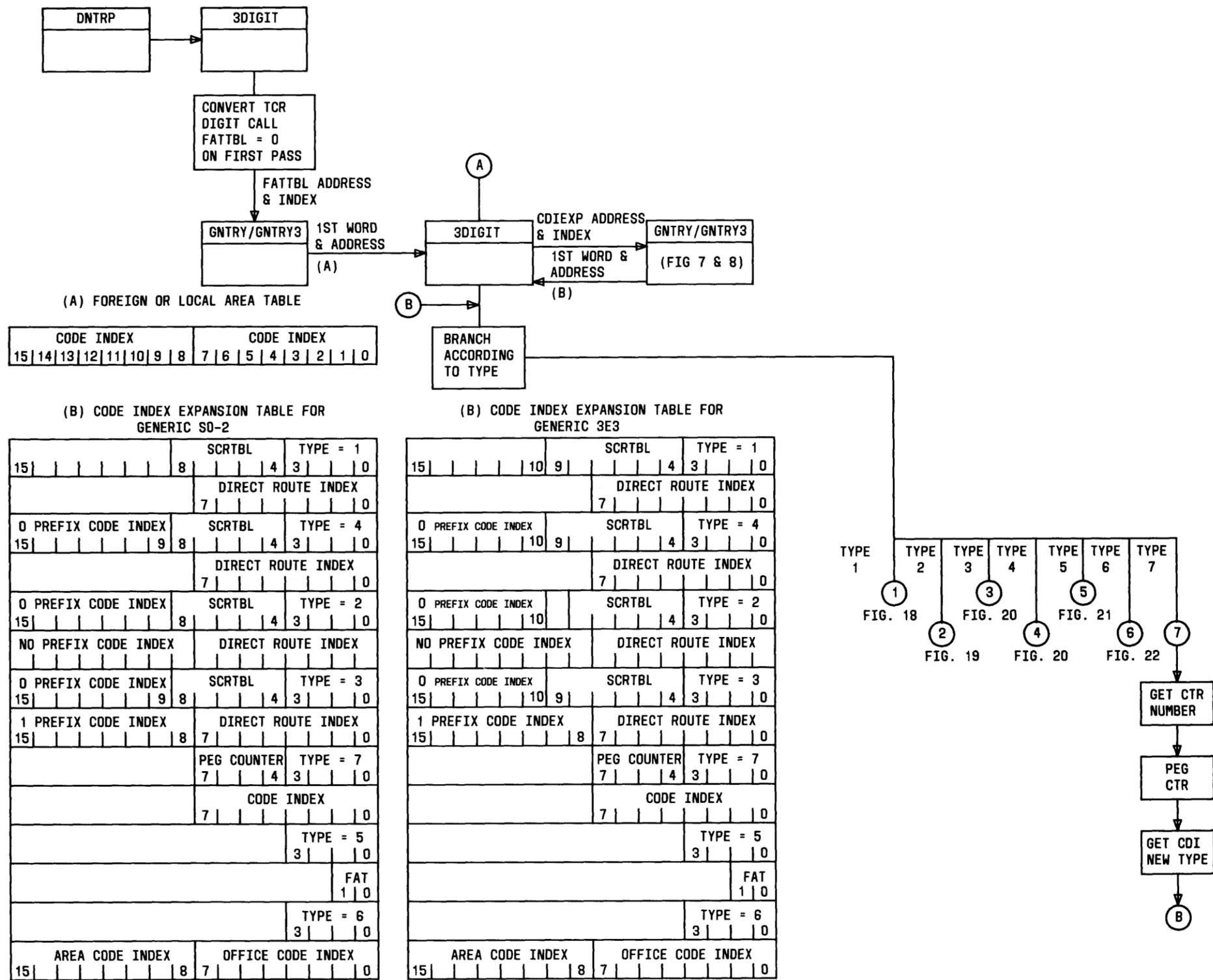


Fig. 17—Code Index Processing

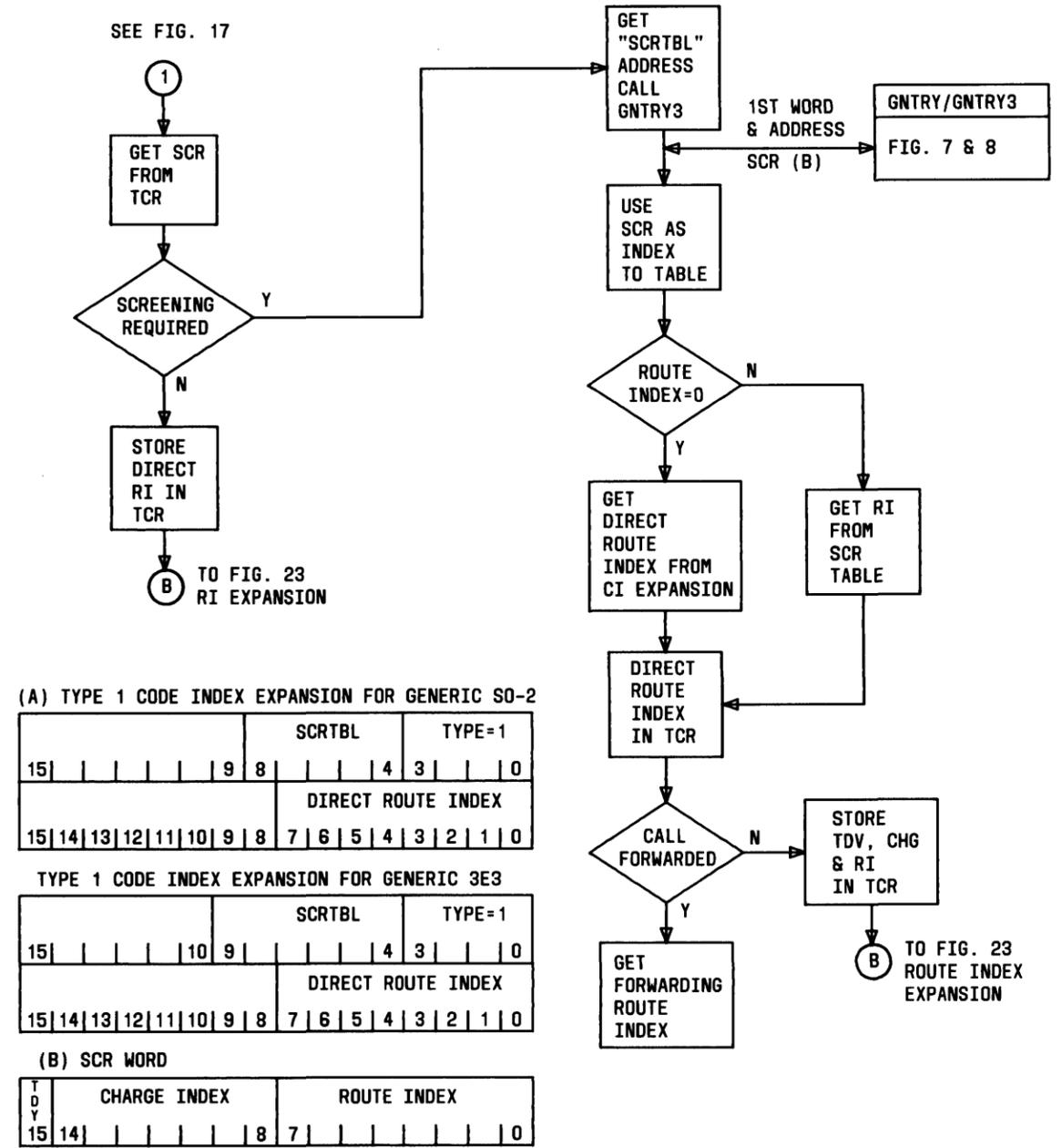
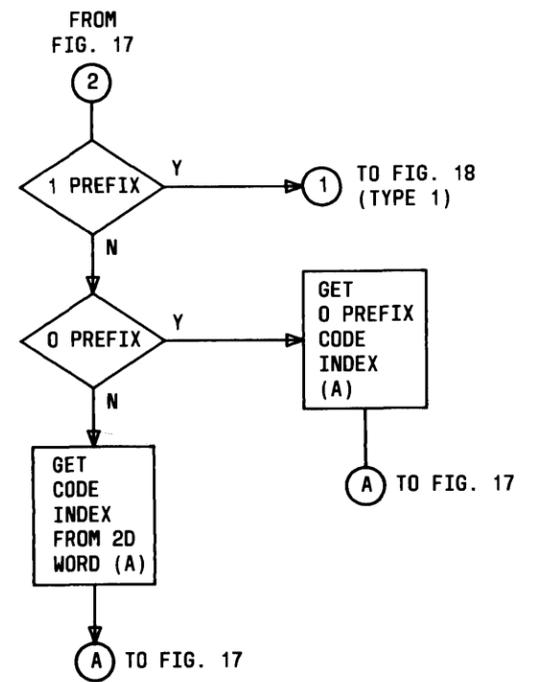


Fig. 18—Type 1 Code Index Expansion Processing



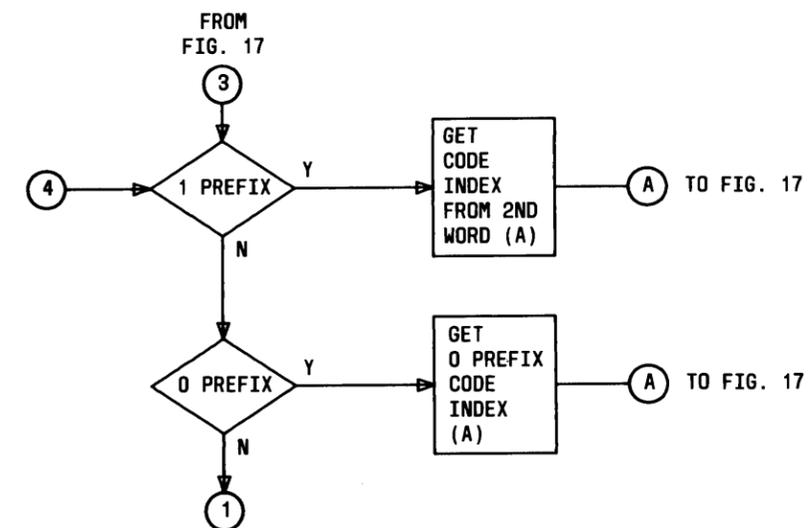
(A) TYPE 2 CODE INDEX EXPANSION WORD FOR GENERIC S0-2

0 PREFIX CODE INDEX								SCRTBL				TYPE=2				
15							9	8				4	3			0
NO PREFIX CODE INDEX								DIRECT ROUTE INDEX								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

TYPE 2 CODE INDEX EXPANSION WORD FOR GENERIC 3E3

0 PREFIX CODE INDEX								SCRTBL				TYPE=2				
15						10	9					4	3			0
NO PREFIX CODE INDEX								DIRECT ROUTE INDEX								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

Fig. 19—Type 2 Code Index Expansion Processing



(A) TYPE 3 CODE INDEX EXPANSION FOR GENERIC SO-2

0 PREFIX CODE INDEX					SCRTBL				TYPE=3		
15				9	8			4	3		0
1 PREFIX CODE INDEX					DIRECT ROUTE INDEX						
15					8	7					0

TYPE 3 CODE INDEX EXPANSION FOR GENERIC 3E3

0 PREFIX CODE INDEX					SCRTBL				TYPE=3		
15				10	9			4	3		0
1 PREFIX CODE INDEX					DIRECT ROUTE INDEX						
15					8	7					0

TYPE 4 CODE INDEX EXPANSION FOR GENERIC SO-2

0 PREFIX CODE INDEX					SCRTBL				TYPE=4		
15				9	8			4	3		0
1 PREFIX CODE INDEX					DIRECT ROUTE INDEX						
15					8	7					0

TYPE 4 CODE INDEX EXPANSION FOR GENERIC 3E3

0 PREFIX CODE INDEX					SCRTBL				TYPE=4						
15				10	9			4	3		0				
1 PREFIX CODE INDEX					DIRECT ROUTE INDEX										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Fig. 20—Type 3 Code Index Expansion Processing

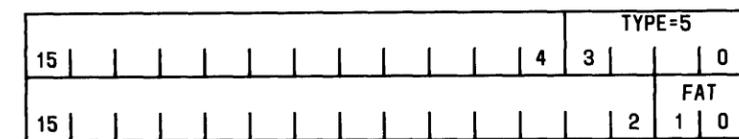
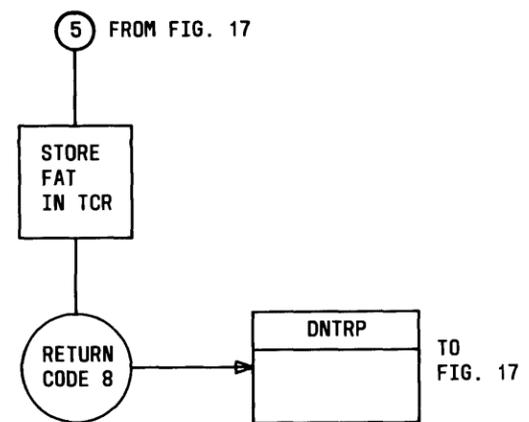
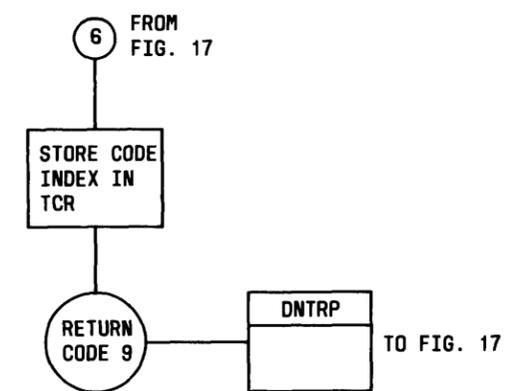


Fig. 21—Type 5 Code Index Expansion Processing



											TYPE 6		
15										4	3		0
AREA CODE INDEX								OFFICE CODE INDEX					
15						8	7						0

Fig. 22—Type 6 Code Index Expansion Processing

FROM FIG. 18

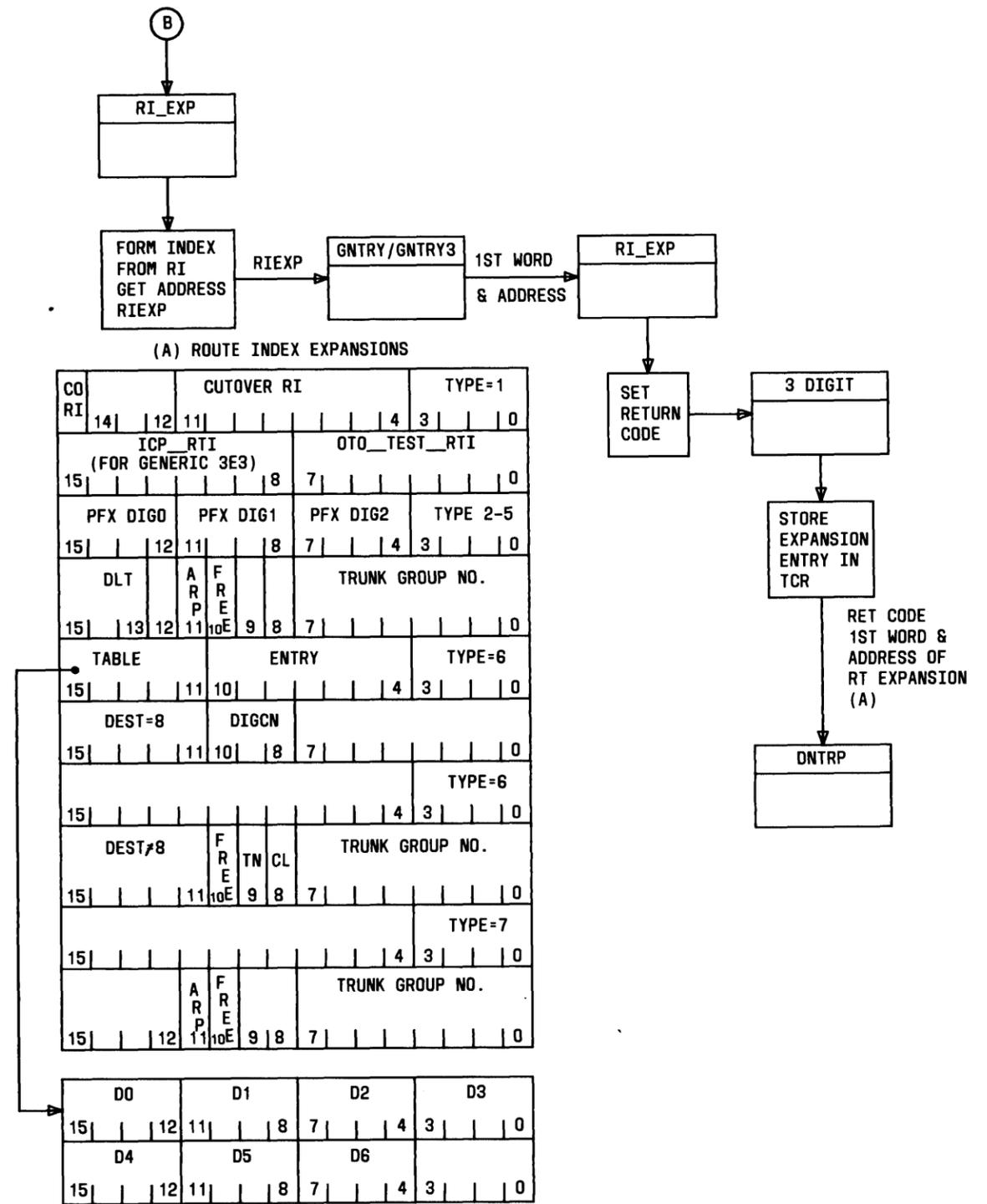
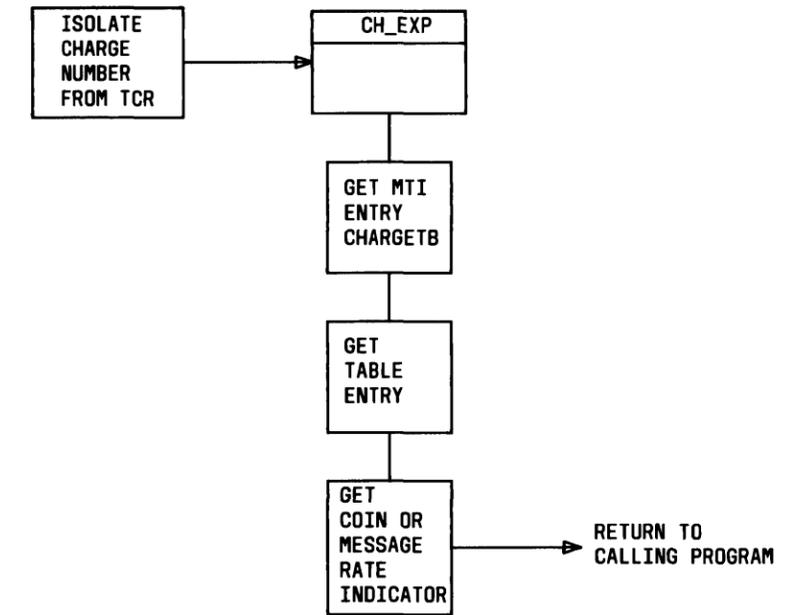


Fig. 23—Route Index Expansion



TYPE	OVERTIME							INITIAL								
0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 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												
0 1	TIME	CHARGE = 0					TIME	CHARGE = 0								
1 0	TIME	CHARGE					TIME	CHARGE								
~ ~ ~ ~ ~																
15																

Fig. 24—Charge Index Expansion

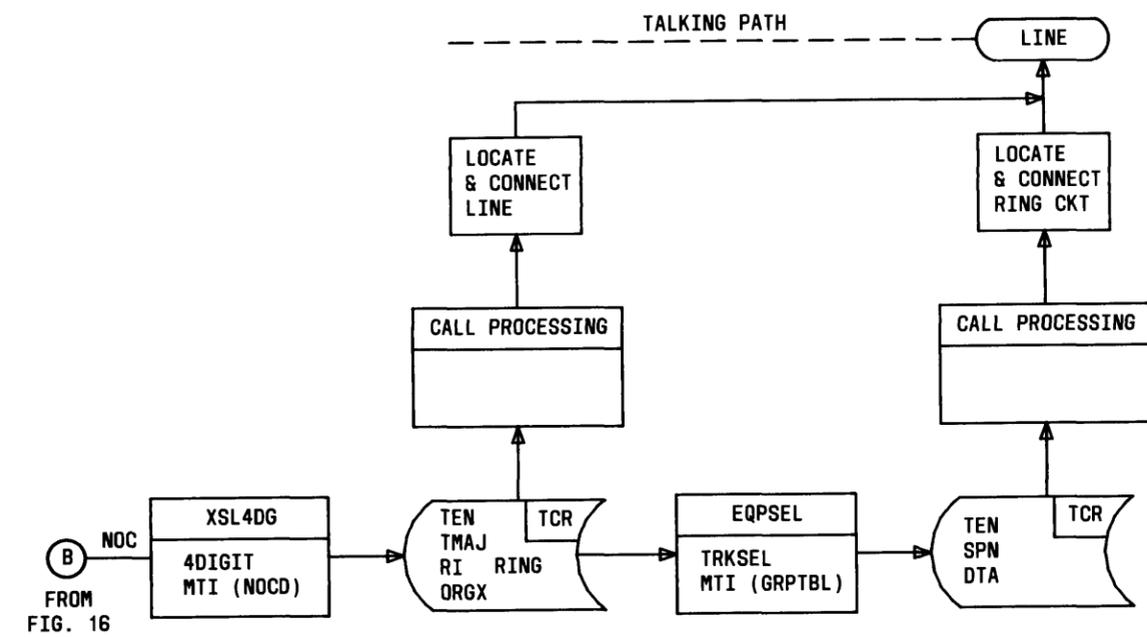


Fig. 25—4-Digit Processing (Intraoffice)

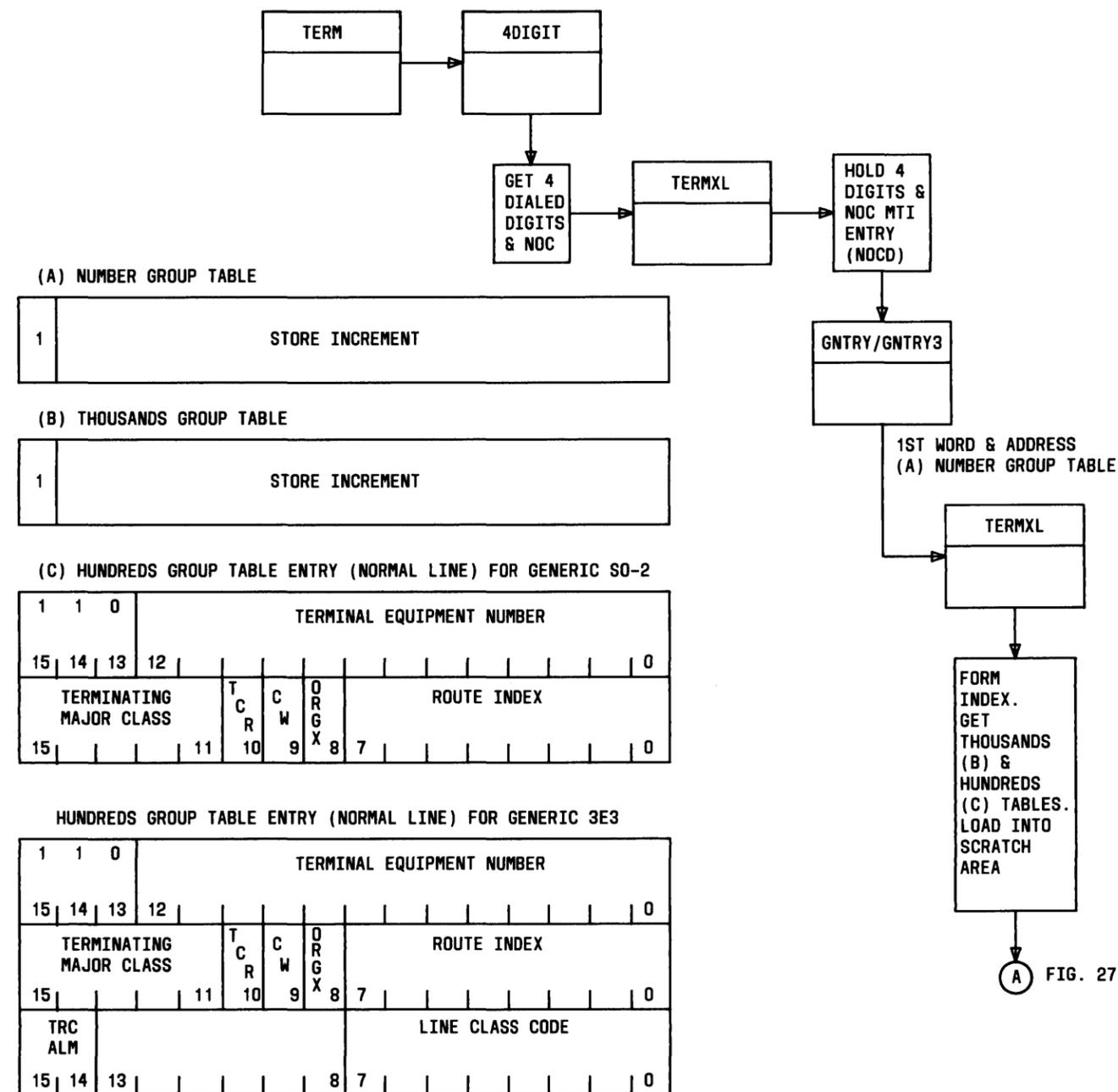
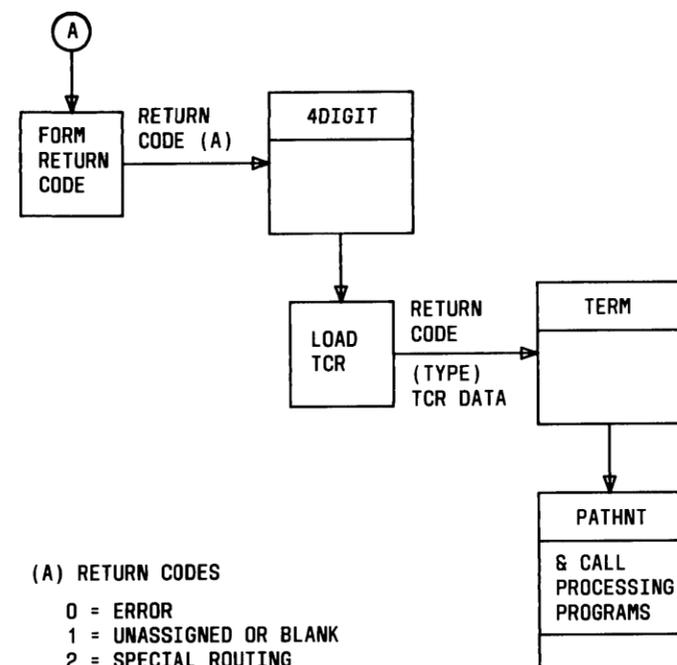


Fig. 26—4-Digit Translation



(A) RETURN CODES

- 0 = ERROR
- 1 = UNASSIGNED OR BLANK
- 2 = SPECIAL ROUTING
- 3 = PBX OR MULTILINE HUNT GROUP
- 4 = LINE HAS A SERIES COMPLETION NUMBER
- 5 = LINE HAS KEY SPN
- 6 = LINE HAS BOTH A KEY SPN & SERIES COMPLETION NO.
- 7 = NORMAL LINE DATA

Fig. 27—Termination of 4-Digit Translation,

PFX DIG0		PFX DIG1		PFX DIG2		TYPE 2-5	
DLT		ARP	FREE	TRUNK GROUP NO.			
15	13			7			0

- ARP = 1 ALTERNATE ROUTE INDEX PROVIDED
- DLT = NUMBER OF DIGITS TO DELETE
- TYPE = 0 UNASSIGNED
- = 1 INTRAOFFICE
- = 2 10-DIGIT INTEROFFICE (NO OVERLAP OUTPUTSING)
- = 3 10-DIGIT INTEROFFICE (OVERLAP OUTPUTSING)
- = 4 7-DIGIT INTEROFFICE (NO OVERLAP OUTPUTSING)
- = 5 7-DIGIT INTEROFFICE (OVERLAP OUTPUTSING)
- = 6 DESTINATION DETERMINED BY DEST CODE IN RI
- = 7 INTEROFFICE (OUTPUTSING-RECEIVED DIGITS)

Fig. 28—Route Index Expansion Words

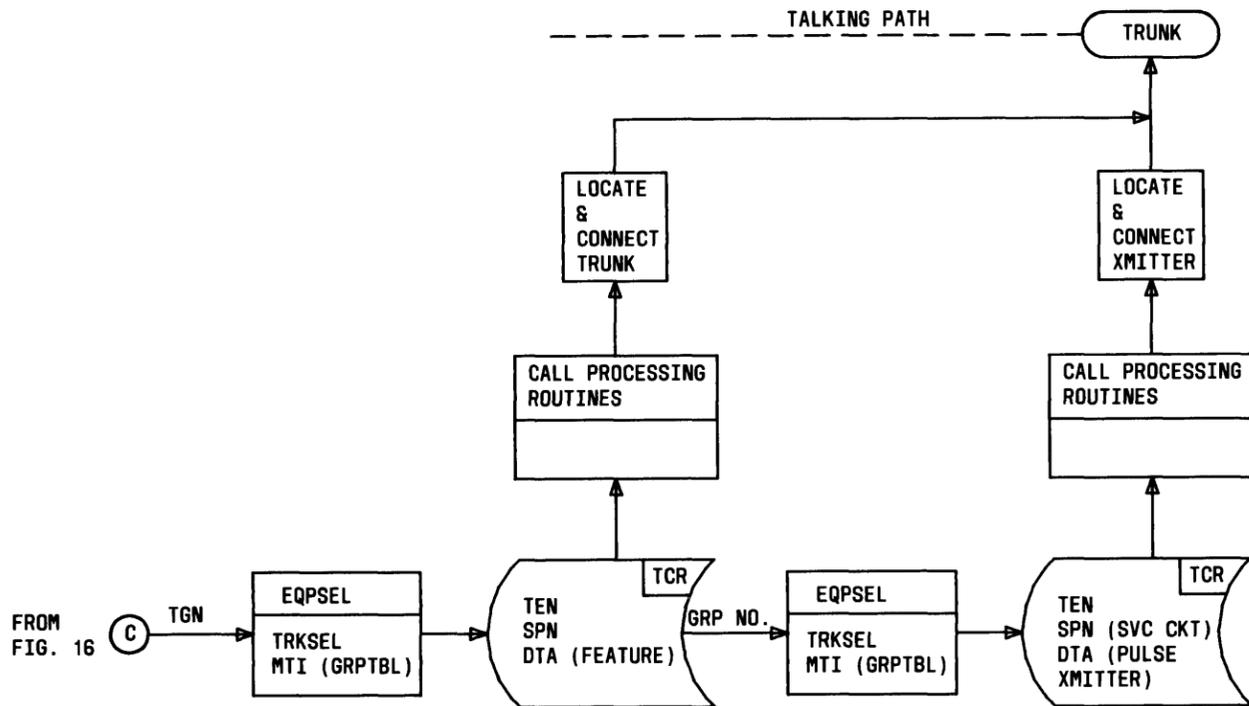
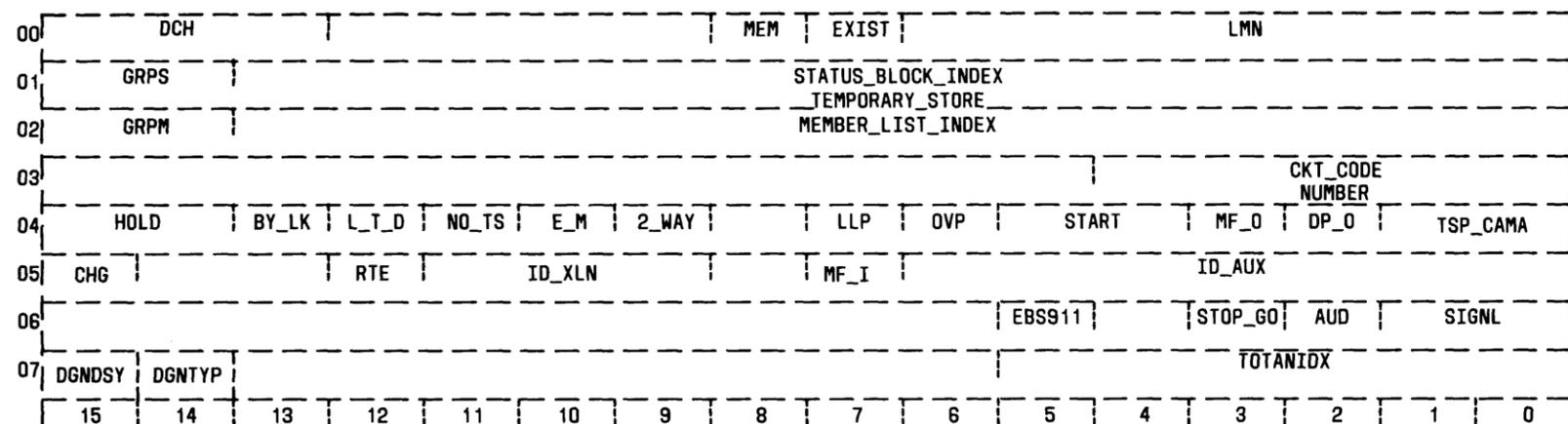


Fig. 29—Interoffice



LMN	LARGEST MEMBER NUMBER	LEGEND	ID_AUX	AUXILIARY INFORMATION REQUIRED FOR INITIAL DIGIT TRANSLATION
EXIST	THE GROUP EXISTS		MF_1	1 = MF INPULSING EXPECTED FROM FAR OFFICE INCOMING DIGIT
MEM	GROUP HAS AT LEAST ONE WORKING MEMBER		ID_XLN	INCOMING DIGIT TRANSLATION CODE WHICH DIRECTS THE INITIAL TRANSLATION AS FOLLOWS:
DCH	TRAFFIC SCHEDULE (TEMPORARY STORE)			000 = TRUNK IS 1 WAY OUTGOING
STATUS_BLOCK_INDEX	HIGH 2 BITS OF GROUP NUMBER			001 = USE 4-DIGIT TRANSLATION ON 1ST FOUR DIGITS RECEIVED
GRPS	HIGH 2 BITS OF GROUP NUMBER			010 = IGNORE FIRST DIGIT AND USE 4-DIGIT TRANSLATION ON NEXT FOUR DIGITS
MEMBER_LIST_INDEX	HIGH 2 BITS OF GROUP NUMBER			011 = USE 1-DIGIT TRANSLATION ON 1ST DIGIT RECEIVED
GRPM	HIGH 2 BITS OF GROUP NUMBER			100 = USE 3-DIGIT TRANSLATION ON 1ST THREE DIGITS RECEIVED
CKT_CODE	NUMBER			101 = FORM 4-DIGIT NUMBER FROM MOST SIGNIFICANT (1000s) DIGIT PROVIDED BY ID_AUX AND 3 RECEIVED DIGITS. THEN USE 4-DIGIT TRANSLATION
TSP_CAMA	00 = REGULAR TRUNK GROUP 01 = CAMA TRUNK GROUP 10 = TSP TRUNK GROUP 11 = TSPS TRUNK GROUP		RTE	REMOTE TEST EQUIPMENT
DP_0	1 = DIAL PULSE OUTPULSING			0 = DEDICATED FACILITIES ARE PROVIDED FOR LOCAL TEST DESK
MF_0	1 = MF OUTPULSING		CHG	1 = NONDEDICATED FACILITIES ARE PROVIDED. REMOTE TEST EQUIPMENT FACILITIES ARE PROVIDED. AUTOCONNECT PROCEDURES ARE USED.
START	00 = IMMEDIATE START 01 = DELAYED DIAL 10 = WINK START - WAIT 350 MS FOR END OF WINK 11 = WINK START - WAIT 1000 MS FOR END OF WINK		SIGNL	1 = CALLS TO GROUP ARE TO BE CHARGED (INCOMING CALLS ONLY)
OVP	1 = OVERLAP OUTPULSING PERMITTED			00 = NO SIGNAL
LLP	1 = LONG LOOP PULSING REQUIRED		AUD	01 = MF INBAND SIGNALING
2_WAY	1 = 2-WAY TRUNK 0 = 1-WAY TRUNK		STOP_GO	10 = MULTIWINK SIGNALING
E_M	1 = E & M TRUNK 0 = LOOP TRUNK		EBS911	11 = MF EXPAND INBAND SIGNALING
NO_TS	1 = NO-TEST TRUNK			1 = RETURN AUDIBLE (OUTGOING TO 23 INTERCEPT SYSTEM)
L_T_D	1 = TRUNK FROM OR TO LOCAL TEST DESK		TOTANIDX	1 = INTERRUPTION OF PULSING IS PERMITTED ON OUTGOING TRUNKS
BY_LK	1 = TRUNK FROM STEP-BY-STEP OFFICE			0 = NOT A 911 TRUNK USE INB FOR FLTBLK OF SIGNALING
HOLD	00 = REGULAR TRUNK GROUP 01 = JOINT HOLD (RECORDING COMPL. OPERATOR) 10 = SERVICE HOLD [TOLL SWITCH TSP(S), OR NO TEST] 11 = CUSTOMER HOLD (NONOPERATOR INTERCEPT TRUNKS)			1 = SINGLE WINK (PRESENTLY USED ONLY BY 911 SERVICE FOR RING BACK)
			DGNTYP	TERMINAL OFFICE TEST ACCESS NUMBER INDEX
			DGNDYSY	0 = AUTO PROGRAM TEST NOT AVAILABLE
				1-63 = TEST ACCESS INDEXES
				0 = RUN OFFICE-TO-OFFICE TEST
				1 = RUN CONTINUITY TEST
				0 = RUN DIAGNOSTICS VIA DAISY CHAIN
				1 = DO NOT RUN DIAGNOSTICS VIA DAISY CHAIN

Fig. 30—Trunk and Service Circuit Group Data for Generic SO-2

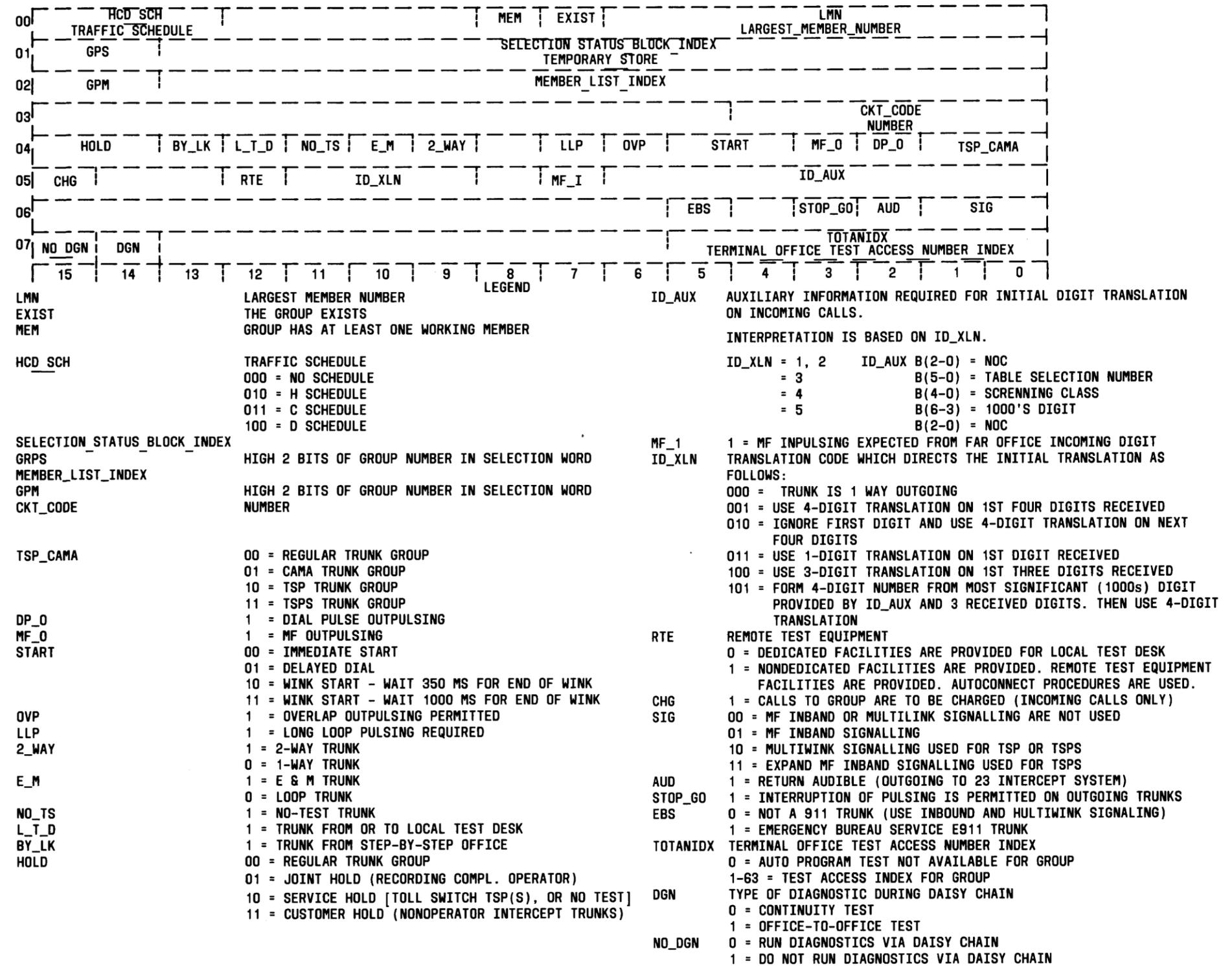


Fig. 31—Trunk and Service Circuit Group Data for Generic 3E3

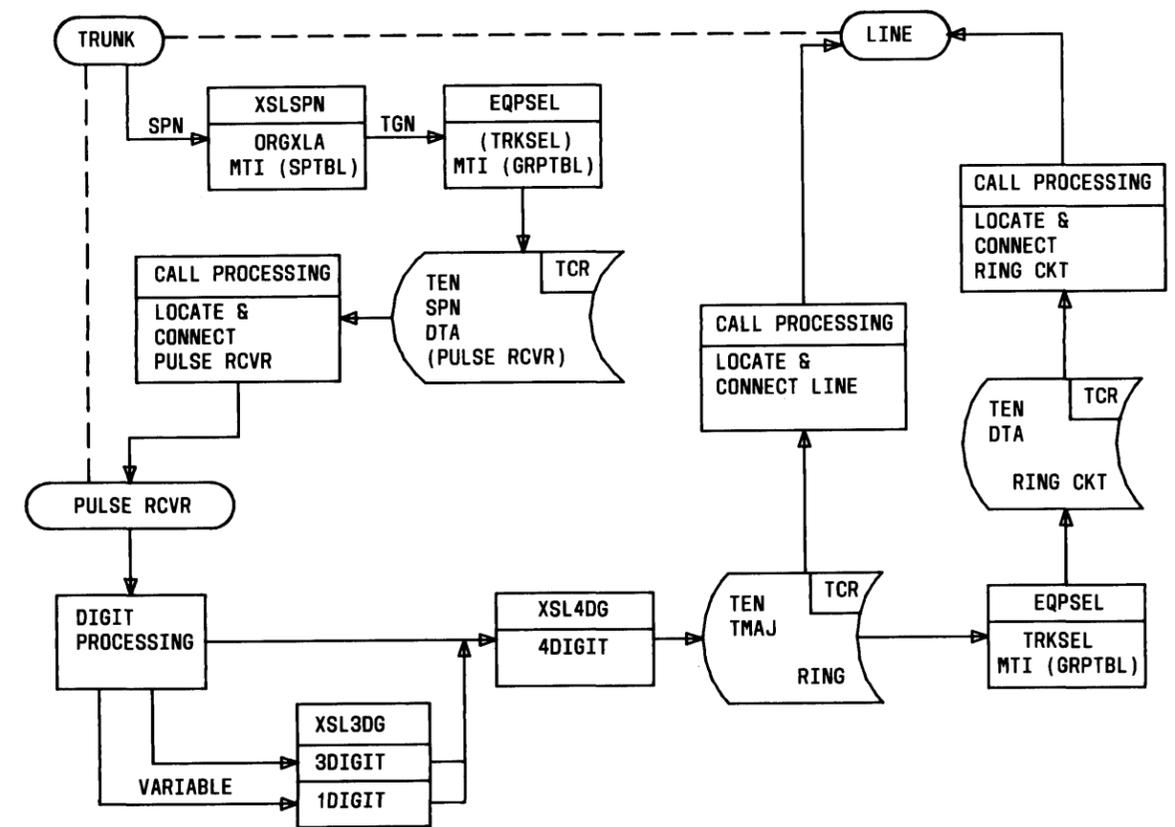


Fig. 32—Trunk Origination

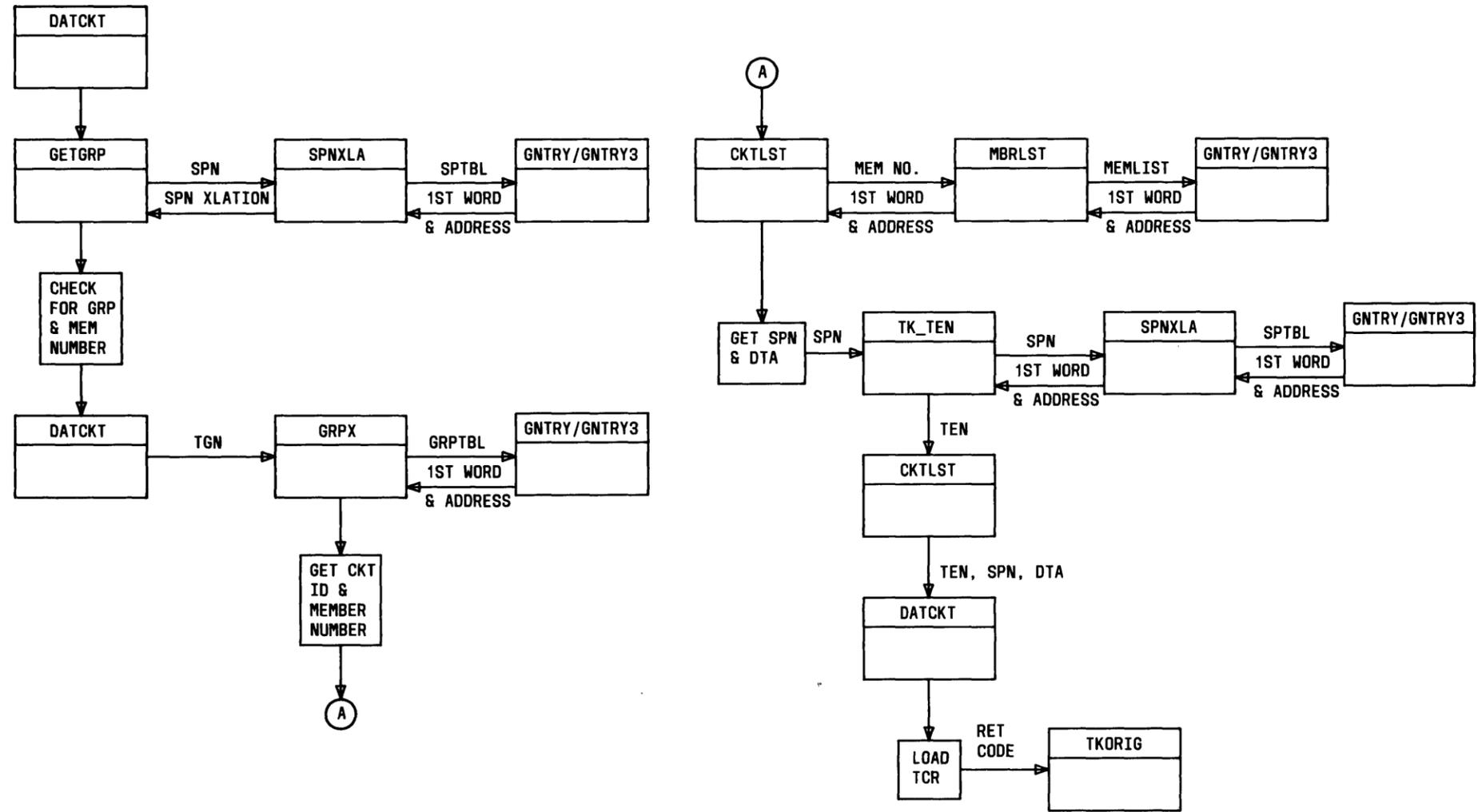


Fig. 33—Trunk Origination Translations

	TYPE		NDE				AUXINFO							
	13	12	11			8	7							0

NDE - NUMBER OF DIGITS EXPECTED OVER THIS TRUNK

TYPE 0 = ERROR, INVALID DIGIT

AUXINFO = ROUTE INDEX (TO A REORDER TONE OR OPERATOR)

1 = INCONCLUSIVE

AUXINFO = TABLE SELECTOR NUMBER TO BE USED IN TRANSLATING THE NEXT DIGIT

2 = LOCALLY TERMINATING CALL

AUXINFO = NORMALIZED OFFICE CODE (NOC)

3 = TANDUM CALL

AUXINFO = ROUTE INDEX

Fig. 34—1-Digit Translations

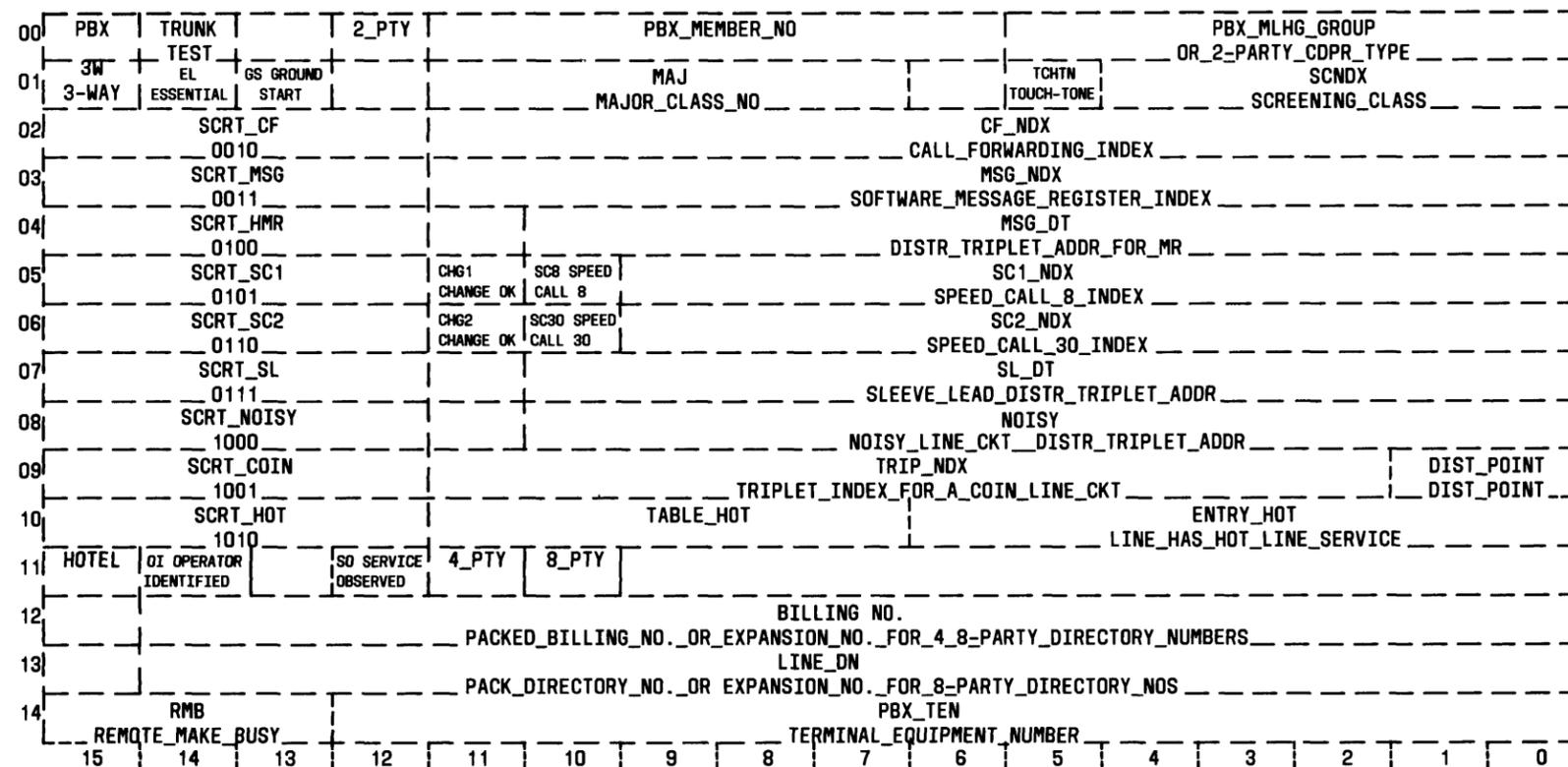


Fig. 35—SCRATCH Data Layout Containing the Line Data for Generic SO-2

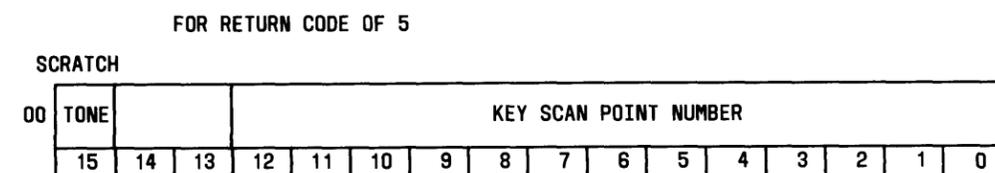
SCRATCH

00	PBX		T_PTY	2_PTY	PBX_MEMBER_NO.				PBX_MLHG_GROUP OR 2-PARTY CDPR FIXBLK							
01	3W	EL	GS		MAJ MAJOR CLASS NO.			TCHTN	SCNDX SCREENING CLASS							
02	SCRT_CF 0010				CF_NDX CALL FORWARDING INDEX											
03	SCRT_MSG 0011				MSG_NDX SOFTWARE MESSAGE REGISTER INDEX											
04	SCRT_HMR 0100				MSG_DT DISTR TRIPLET ADDR FOR MR											
05	SCRT_SC1 0101				CHG1	SC1_NDX SPEED CALL 8 INDEX										
06	SCRT_SC2 0110				CHG2	SC2_NDX SPEED CALL 30 INDEX										
07	SCRT_SL 0111				SL_DT SLEEVE LEAD-DISTR TRIPLET ADDR											
08	SCRT_NOISY 1000				NOISY NOISY LINE CKT - DISTR TRIPLET ADDR											
09	SCRT_COIN 1001				TRIP_NDX TRIPLET INDEX FOR A COIN LINE CKT							DIST_POINT DIST POINT				
10	SCRT_HOT 1010				TABLE_HOT				ENTRY_HOT LINE HAS HOT LINE SERVICE							
11	HOTEL	OI	SPS	SO	4_PTY	8_PTY					INHROH	INHALIT				
12	BILLING_NO. PACKED BILLING NO. OR EXPANSION NO. FOR 4/8 PARTY DIRECTORY NUMBERS OR PACKED SPECIAL BILLING NO. FOR WATS															
13	LINE_DN PACK DIRECTORY NO. OR EXPANSION NO. FOR 8-PARTY DIRECTORY NO.S															
14	RMB RANDOM MAKE BUSY				PBX_TEN TERMINAL EQUIPMENT NUMBER											
15	UNUSED															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

LEGEND

2_PTY	TWO-PARTY LINE	8_PTY	8-PARTY LINE
T_PTY	TIP PARTY OF A 2-PARTY LINE	4_PTY	4-PARTY LINE
TCHTN	TOUCH-TONE	SO	SERVICE OBSERVING
GS	GROUND START LINE	OI	OPERATOR IDENTIFIED
EL	ESSENTIAL LINE	HOTEL	HOTEL-MOTEL PBX
3W	3-WAY SERVICE		
CHG1	LINE PERMITTED TO CHANGE 8-CODE LIST		
CHG2	LINE PERMITTED TO CHANGE 30-CODE LIST		
INHALIT	INHABIT ALIT TESTING ON-LINE		
INHROH	INHABIT ROH TONE RETURNED ON-LINE		

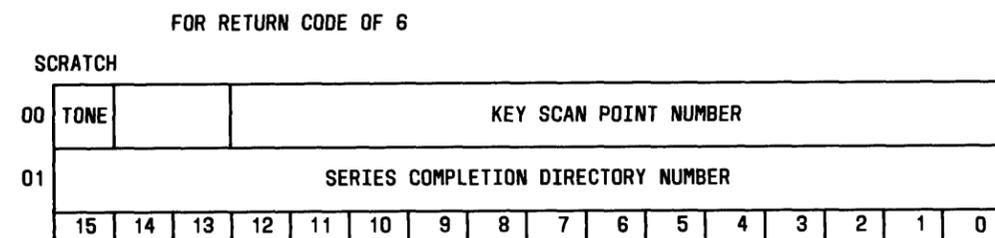
Fig. 36—SCRATCH Data Layout Containing the Line Data for Generic 3E3



LEGEND

TONE 0 = REORDER TYPE
 1 = BUSY TONE

Fig. 37—Base Level SCRATCH Area Upon Exit From Subroutine 4DIGIT or 4DSCX (Return Code of 5)



LEGEND

TONE 0 = REORDER TYPE
 1 = BUSY TONE

Fig. 38—Base Level SCRATCH Area Upon Exit From Subroutine 4DIGIT or 4DSCX (Return Code of 6)

FOR RETURN CODE OF 2 (ROUTE TO ANOTHER OFFICE)

SCRATCH

02		TABLE NUMBER	ENTRY NUMBER													
03	01		CODE INDEX													
04			HG_HI													
05	HG_LO															
06			CD_HI													
07	CD_LO															
08	01		CD_IDX													
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

LEGEND

- HG_HI HUNDRED GROUP TABLE ADDRESS HIGH BITS
- HG_LO HUNDRED GROUP TABLE ADDRESS LOW BITS
- CD_HI HIGH BITS OF THE ADDRESS OF THE CODE INDEX NUMBER GROUP ENTRY
- CD_LO LOW BITS OF THE ADDRESS OF THE CODE INDEX NUMBER GROUP ENTRY
- CD_IDX CODE INDEX

Fig. 39—Base Level SCRATCH Area Upon Exit From Subroutine TERMXL (Return Code of 2) For Generic SO-2

FOR RETURN CODE OF 2 (ROUTE TO ANOTHER OFFICE)

SCRATCH

02					TABLE NUMBER				ENTRY NUMBER							
03	01					CODE INDEX										
04																
05											HG_HI					
06	HG_LO															
07											CD_HI					
08	CD_LO															
09	01					CD_IDX										
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

LEGEND

- HG_HI HUNDRED GROUP TABLE ADDRESS HIGH BITS
- HG_LO HUNDRED GROUP TABLE ADDRESS LOW BITS
- CD_HI HIGH BITS OF THE ADDRESS OF THE CODE INDEX NUMBER GROUP ENTRY
- CD_LO LOW BITS OF THE ADDRESS OF THE CODE INDEX NUMBER GROUP ENTRY
- CD_IDX CODE INDEX

Fig. 40—Base Level SCRATCH Area Upon Exit From Subroutine TERMXL (Return Code of 2) For Generic 3E3

FOR RETURN CODE OF 3

SCRATCH

02	TABLE NUMBER				ENTRY NUMBER											
03	TERMINATING MAJOR CLASS	TRC	0	ORGX							LAST HUNT NUMBER					
04												HG_HI				
05	HG_LO															
06												NG_HI				
07	NG_LO															
08	NG_SCTH 1	IT_SCTH														
09												TG_HI				
10	TG_LO															
11	TG_SCTH 1	PC_SCTH	IH_SCTH													
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

LEGEND

ORGX	ORIGINATING TRANSLATION DATA REQUIRED	TG_HI	HIGH BITS OF THE THOUSANDS GROUP TABLE ENTRY ADDRESS
TRC	CALL TRACE FEATURE	TG_LO	LOW BITS OF THE THOUSANDS GROUP TABLE ENTRY ADDRESS
HG_HI	HUNDRED GROUP TABLE ENTRY ADDRESS HIGH BITS	IH_SCTH	STORE INCREMENT FOR HUNDRED GROUP TABLE
HG_LO	HUNDRED GROUP TABLE ENTRY ADDRESS LOW BITS	PC_SCTH	PRECUT BIT
NG_HI	HIGH BITS OF NUMBER GROUP TABLE ENTRY	TG_SCTH	STORE INCREMENT EXISTS FOR HUNDREDS GROUP TABLE
NG_LO	LOW BITS OF NUMBER GROUP TABLE ENTRY		
IT_SCTH	STORE INCREMENT FOR THOUSAND GROUP TABLE		
NG_SCTH	STORE INCREMENT EXISTS FOR THOUSANDS GROUP TABLE		

Fig. 41—Base Level SCRATCH Area Upon Exit From Subroutine TERMXL (Return Code of 3) for Generic SO-2

FOR RETURN CODE OF 3

SCRATCH

02					TABLE NUMBER				ENTRY NUMBER							
03	TERMINATING MAJOR CLASS				TRC	0	ORGX					LAST HUNT MEMBER				
04	TRACE ALARM						LINE CLASS CODE INDEX									
05											HG_HI					
06	HUNDRED GROUP TABLE ENTRY ADDRESS LOW BITS															
07											NG_HI					
08	LOW BITS OF NUMBER GROUP TABLE ENTRY															
09	NG_SCTH 1											STORE INCREMENT FOR THOUSAND GROUP TABLE				
10											TG_HI					
11	LOW BITS OF THE THOUSANDS GROUP TABLE ENTRY ADDRESS															
12	TG_SCTH 1	PC_SCTH											STORE INCREMENT FOR HUNDRED GROUP TABLE			
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

LEGEND

- ORGX ORIGINATING TRANSLATION DATA REQUIRED
- TCR CALL TRACE FEATURE
- HG_HI HUNDRED GROUP TABLE ENTRY ADDRESS HIGH BITS
- NG_SCTH STORE INCREMENT EXISTS FOR THOUSANDS GROUP TABLE
- TG_HI HIGH BITS OF THE THOUSANDS GROUP TABLE ENTRY ADDRESS
- PC_SCTH PRECUT BIT
- TG_SCTH STORE INCREMENT EXISTS FOR HUNDREDS GROUP TABLE

Fig. 42—Base Level SCRATCH Area Upon Exit From Subroutine TERMXL (Return Code of 3) for Generic 3E3

TABLE A
SEQUENTIAL MEMORY LAYOUT OF MASTER TABLE INDEX ENTRIES
(NOTES 1, 2#, 3#, 4*, AND 5*)

NAME	WORDS	NOTES
EQIOCHAN	2#, 3*	5*, 6
TAPEOPTS	1	6
TAPEQUE	1	6
TAPEQUO	1	6
TAPETBL	3	6
LSTTSWRD	2	6
WPTBL	4#, 16*	6
T1CPBS	25#, 33*	
EQMISMOD	1	
ALITWRD	1	
JCT_RATE	1	
SVC_RATE	1	
TRK_RATE	1	
BOSCAN	1	
CHARGETB	16	
CKTSCAN	258	7
TRKSCAN	192	7
CLSRV	8	
CP_FEM	1	
DSYBLK	7	
DADADTA*	1*	5*
EQPT_PD	32	
IBLK	3	
INTCHG	1	
LNPATBL	2#, 4*	
LOG_SCAN	1	
MASTYPE (SPARE)	1	
(SPARE)#	7#	
MSGCL	24#, 48*	

Contains
Data

* Applies to 3E3 Generic Only

Applied to Issue 4A of the SO-2 Generic

See Notes at end of table.

TABLE A (Contd)

SEQUENTIAL MEMORY LAYOUT OF MASTER TABLE INDEX ENTRIES
(NOTES 1, 2#, 3#, 4*, AND 5*)

NAME	WORDS	NOTES	
MT_FEM	1		Contains Data
N_OPDATA#	2#		
NETWEQPT	1		
NETWMT	1		
NOCBCD	8		
NPAFAT*	4*	8*	
OFF_DATA	2#, 3*		
OFFICE	8		
OIEXP	16		
SCHEDINH*	16*	5*	
SCRSIZE	1		
SKEDBLK	15		
TRAFD	27		
TLUID	16		
TLTPL	2		
NOTEST	4		
TUTTL*	1*		
FETL*	2*		
WEBCN	3		
SPDATA	20	9*	
AMA_BUF	3		Contains Temporary Store Addresses
CFTBL	3		
MRTBL	3		
COINSTAT	3		
GRPSTAT	12		
3PCSTAT	3		
SPTMPMTI	3#, 90*	9*, 10#	
TSPARE	3	11	

* Applies to 3E3 Generic Only

Applies to Issue 4A of SO-2 Generic Only

See Notes at end of table.

TABLE A (Contd)

**SEQUENTIAL MEMORY LAYOUT OF MASTER TABLE INDEX ENTRIES
(NOTES 1, 2#, 3#, 4*, AND 5*)**

NAME	WORDS	NOTES
SC1LST	3	
SC2LST	3	
CFALST	3#	
CFLST	3*	
CODLST*	3*	
MRALST#	3#	
SBILTBL	3	
CCTBL*	3*	
LCCTBL*	3*	
CUTSTAT	3	
CALLBACK	3	
TOTANTBL	3	
1DGTBL	3	
FATTBL	12	
CDIEXP	3	
SCTBLS	3	
RIEXP	3	
ARIEXP	3	
COINTRIP	3	
GRPTBL	12	
MEMLST	12	
EXPTBL	3	
SPTBL	3	
SVCTBL	3	12
SV_TBL	(2)	12
NOCD	24	
SPXLAMTI	3#, 90*	9*, 10#
SPARES	3	11
STRLMT	3	13
STRLIM	(2)	13

Contains
Program Store
Addresses

* Applies to 3E3 Generic Only

Applies to Issue 4A of SO-2 Generic Only

See Notes at end of table.

TABLE A (Contd)

**SEQUENTIAL MEMORY LAYOUT OF MASTER TABLE INDEX ENTRIES
(NOTES 1, 2#, 3#, 4*, AND 5*)**

NAME	WORDS	NOTES
LCC_TBL#		14#
TNXLCC#		14#

* Applies to 3E3 Generic Only

Applies to Issue 4A of SO-2 Generic Only.

Notes:

1. The Master Table Index is a fixed area of program store which contains or provides linkage to the various parameters and translation data for the generic program. This area is used for the Master Table Index entries listed or reserved for future Master Table Index entries. This area may not be used for any other purpose.
- 2# To satisfy "reallocation" requirements, the translators pointed to by the Master Table Index entries containing store address are to be in the same sequence as the Master Table Index entries.
- 3#. The Master Table Index entries must start on a 4K boundary. For Generic Issue 4, the Master Table Index entries start at $\emptyset(250000)$. For Generic Issue 4A, the Master Table Index entries start at $\emptyset(260000)$.
- 4*. The Master Table Index must start on a 4K boundary. For the 3E3 generic, the Master Table Index starts at 136K (octal 420000). The last word of TRANSLATION store is the last word of PHYSICAL store of 256K-1 (octal 777777) whichever is less.
- 5*. Data words are not recent changeable for the 3E3 generic. The data is inputted or changed by a translation overwrite.
6. These data words are generic-defined and are not changeable through the recent change programs.
7. CKTSCAN includes the 192 words for TRKSCAN.
- 8*. Data used by office records only. Inputted by ODA or recent change.
- 9*. There are 20 spare words of data, 30 spare 3-word Master Table Index entries for temporary store and 30 spare 3-word Master Table Index entries for translation store.
- 10#. There are 20 spare words of data. A spare 3-word Master Table Index entry for temporary store and a spare 3-word MTI entry for translation store.

TABLE A (Contd)

**SEQUENTIAL MEMORY LAYOUT OF MASTER TABLE INDEX ENTRIES
(NOTES 1, 2#, 3#, 4*, AND 5*)**

Notes (Contd):

11. There are two 3-word Master Table Index entries, one for temporary store and one for translation store, giving the number of spare words in each store in the first word.
12. The 3 words for SVCTBL include the 2 words for SV_TBL.
13. The 3 words for STRLMT include the 2 words for STRLIM.
- 14#. These are nonresident translation tables. They are not referenced by the Master Table Index.

TABLE B
 ORIGINATING AND TERMINATING MAJOR CLASSES
 (SO-2, ISSUES 4 AND 4A, 3E3)

MAJOR CLASS	ASSIGNMENT	ORIG	TERM
0	Unassigned	1	1
1			
2			
3			
4	Two-Party Ring	1	1
5	Two-Party Tip	1	1
6	Individual - Traffic	1	1
7	Individual - Free		1
8	Individual	1	1
9	Hotel - Motel	1	
10	Manual	1	
11	*TSPS Selective Call Screening	1	
12			
13			
14			
15			
16	Multiparty - Party 1	1	1
17	Multiparty - Party 2		1
18	Multiparty - Party 3		1
19	Multiparty - Party 4		1
20	Multiparty - Party 5		1
21	Multiparty - Party 6		1
22	Multiparty - Party 7		1
23	Multiparty - Party 8		1
24	Coin-First Coin	1	1
25	Dial-Tone-First Coin	1	1
26			
27			
28	Intercept		1
29	Autoconnect	1	1
30	Denied Service	1	1
31	Special Routing		1

* Note: This assignment is only found in the 3E3 generic.

TABLE C
CIRCUIT CODES FOR TRUNK AND
SERVICE CIRCUIT GROUPS

CIRCUIT CODE	DESCRIPTION
0	Tone and Announcement Circuits
1	Two-Way E- and M-Lead Trunk
2	Spare
3	Incoming Reverse Battery Trunk (Wink or Immediate)
4	Incoming Reverse Battery Trunk (Delay Dial)
5	Outgoing Reverse Battery High-Low Trunk
6	Spare
7	Office-to-Office Test Trunk
8	Customer Dial Pulse Receiver Circuit
9	TOUCH-TONE Receiver — Includes: Customer Dial Pulse Receiver Circuit and TOUCH-TONE Calling Detector Circuit
10	Multifrequency Receiver Circuit
11	Multifrequency Transmitter Circuit
12	Dial Pulse Transmitter Circuit
13	Regular Ringing Circuit
14	Superimposed Ringing Circuit
15	Coin Control Circuit
16	Conference Circuit
17	Incoming Local Test Desk (No. 14 and No. 16)
18	Station Ringer Test Circuit
19	Other Test Circuits (Note): Continuity and Polarity Test Dial Pulse Receiver Test Transmission Test Termination MW and Transmission Environment Test Loop Environment Test Tone Presence Detector Line Insulation Test TOUCH-TONE Receiver Test

Note: Other test circuits are included in one service circuit group with fixed member number assignments.

TABLE D
ENTRY IDENTIFICATION CODES

ENTRY_ID	DESCRIPTION
0	Containing a vacant entry which is skipped
1	Containing an expansion entry which has pointers to another block of data
2	Containing a call forwarding index which is stored in word 2 of SCRATCH
3	Containing a software message register index which is stored in word 3 of SCRATCH
4	Containing a hardware message register distributor triplet address which is stored in word 4 of SCRATCH
5	Containing a speed call 8 index which is stored in word 5 of SCRATCH
6	Containing a speed call 30 index which is stored in word 6 of SCRATCH
7	Containing a sleeve-lead circuit distributor triplet address which is stored in word 7 of SCRATCH
8	Containing a noisy line circuit distributor triplet address which is stored in word 8 of SCRATCH
9	Containing a coin line circuit triplet index and distributor point number which is stored in word 9 of SCRATCH
10	Containing a hot line service table number and entry number which is stored in word 10 of SCRATCH
11	Containing a PBX individual line data which is the originating major class, the TOUCH-TONE® indicator, and the screening class which is stored in word 1 of SCRATCH
12*	Containing a bit to indicate that the line is not to be ALIT tested, and a bit to indicate that the line is not to have receiver off-hook (ROH) tone returned on the line.

* Applies to 3E3 Generic only.