QUEUE AND GENERAL PURPOSE SOFTWARE SUBSYSTEM DESCRIPTION (SSD)
2-WIRE NO. 1 AND NO. 1A ELECTRONIC SWITCHING SYSTEMS

CONTENTS PAGE

CONTENTS PAGE

Figures

Tables

Printed in U.S.A.

CONTENTS PAGE

1. GENERAL

INTRODUCTION

1.01 This section provides an introduction to the queue and general purpose software subsystem operating in a No. 1/1A Electronic Switching System (ESS) central office. Information unique to the No. 1A ESS is so noted; applications unique to No. 1 ESS are not described in this document.

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

1.03 Part 4 of this section provides a defined list of abbreviations and acronyms used herein.

PURPOSE OF QUEUE AND GENERAL PURPOSE SOFTWARE

1.04 Queueing programs act as administrators for various queues or lists containing items waiting to be handled by the system. When processing a call, the system may encounter various busy conditions (eg, customer dial pulse receivers, outgoing trunks, etc). When these conditions occur, the call is delayed by having it wait in a queue or list until the serving facility required to continue the processing of the call becomes available.

1.05 General purpose programs perform preliminary work and particular functions for client programs. Some of the general purpose programs provide service routines which return control to the client programs when execution is complete. Others perform jobs of considerable magnitude and remain in control of a call during real-time breaks.

2. PIDENTS DESCRIBED IN SECTION

2.01 Program identifications (pidents) categorized as queue and general purpose software are described in this section. Table A provides a pident-program listing cross-reference.

2.02 Pident descriptions are based on the 1AE5 version of the generic program.

3. FUNCTIONAL DESCRIPTION OF QUEUE AND GENERAL PURPOSE PIDENTS

GENERAL PURPOSE QUEUEING

WQUE—Queue Administration Program

3.01 The system may encounter various busy conditions while processing a call. When these conditions occur, the call is delayed by having it wait in a queue or list until the register or hardware necessary to continue the processing of the call becomes available. The queue administration program (WQUE) loads the various queues when requested by client programs due to busy conditions, and unloads the same queues when the equipment or registers are available. If a call is abandoned or a time-out occurs, an item can be removed from a queue at the request of the client program. Refer to Table B for a listing of WQUE global subroutines which perform queue loading and unloading functions for various client programs (referencing pidents).

Types of Queues Administered

3.02 This program administers two types of queues: fixed length queues and variable length queues. A ***fixed length queue*** (FLQ) is a call store (CS) block of memory engineered to a length dependent upon the office size and traffic requirements. This block contains information on

TABLE A

PIDENT—PROGRAM LISTING CROSS-REFERENCE

PIDENT NAME		PROGRAM LISTING (PR)	
		NO. 1 ESS	NO. 1A ESS
GENERAL PURPOSE QUEUEING			
WQUE	Queue Administration	1A125	6A125
QUEUEING FOR TRUNKS AND LINES (QTL)			
QAPR	QTL—Queue Administration and Processing	1A173	6A173
QCDL	QTL—Queueing Data Link Automatic Call Distribution Interface	1A173	6A173
QCIA	QTL—Customer Interface and Special Auditing Routines	1A173	6A173
QEDA	QTL—Queue Entry and Destination Assignment Routines	1A173	6A173
QSIF	QTL—Queue State Information Features	1A173	6A173
QTAL	QTL—Give Audible, Disconnect, and Line Termination Routines	1A173	6A173
QTRK	QTL—Terminate to Trunk Facility Subroutines	1A173	6A173
QWAT	QTL—Queueing for WATS	1A204	6A204
GENERAL PURPOSE PROGRAMS			
CHGD	Scan Point Change Director	1A126	6A126
COPR	Report and Miscellaneous Subroutines	1A126	6A126
CXYH	Seize and Release Routines and L, J, and T Bit Administration for Centrex	1A126	6A126
TRCE	Call Trace	1A086	6A086
YAHA	Seize and Release Routines and L, J, and T Bit Administration	1A126	6A126
YCLK	Register Linking Routine	1A126	6A126
YFDS	Scan of Single Master Scanner Point	1A126	6A126
YFTO	Incoming Trunk to Busy, Overflow, or Special Service Circuit	1A126	6A126
YMRG	Miscellaneous Register Subroutines and Tables	1A126	6A126
YTTO	Originating Line to Busy, Overflow, or Special Service Circuit	1A126	6A126
ZERO	Call Store Zeroing	1A126	6A126

customers unable to be served due to the unavailability of equipment.

3.03 Each fixed length queue has a four-word head cell (HC) in CS. Words comprising the HC are the load pointer, the load bottom address, the unload pointer, and the unload bottom address. The load pointer contains the address where the next item [line equipment number (LEN), trunk scanner number (TSN), or client register address (CRA)] to be entered on the queue is loaded. The unload pointer contains the address where the next item to be removed is found. The queue bottom pointers are used to denote when the bottom of the queue is reached. In this event, loading or unloading will continue at the top of the memory block. When a fixed length queue is empty, the addresses in the load and unload pointer locations are equal. When the program attempts to load an item on a fixed length queue and the queue is currently full, this indication is reported to the client program. A list of FLQs administered by WQUE is given in Table C.

3.04 There are two categories of ***variable length queues*** (VLQ): the linked call register (CR) queue and the linked path memory for trunks (PMT) word queue. The linked call register queue is a one-way list of call registers which have been linked together (via the CR linking mechanism) to await availibility of equipment, availability of CS memory, a time-out, etc. This type of queue uses

TABLE B

WQUE GLOBAL SUBROUTINES

GLOBAL	PRIMARY FUNCTION	REFERENCING PIDENTS
SUPVON	Tests for a hit when an on-hook report is received while call is on queue	CHGD
WQAIOD	Loads AIOD queue	AIOD
WQBMSG	Sets up control word to indicate if blocked LENs are to be printed [based on parameter (ON or OFF) entered via message BDT—LENPRT—aa.]	TTIA
WQINDD	Links new trunk onto incoming overload control queue	TNLS
WQINOC	Links new trunk onto incoming overload control queue	DCNT
WQLAMA	Loads AMA register on AIOD queue	CFUP, AMAC
WQLCXA	Loads centrex attendant on variable length queue	CXKY
WQLDRR	Loads ringing register queue	RVRT, QAPR, OFTR, ICAL, ADLR
WQLFXC	Loads an item on a fixed length queue	ADUP
WQLFXE	Derives trunk idle list head cell (HC), sets HC sign bit to inhibit equipment seizure, and loads item on fixed length queue	DCNL, ADUP
WQLLKC	Obtains associated idle list head cell, sets bit in first word of HC to inhibit equipment seizure, and loads item on either a trunk digit receiver queue or an AIOD queue	NCIN, COIN, CAMA
WQMDRR	Sets up main program return and performs same function as WQLDRR	ORDL
WQMFXC	Sets up main program return and performs same function as WQLFXC	DCNL
WQMLKC	Sets up main program return and performs same functions as WQLLKC	OFTR, NCIN, COIN, CAMA
WQMLKE	Sets up main program return, derives trunk idle list HC, sets HC sign bit to inhibit equipment seizure, and loads item on either a trunk digit receiver queue or an AIOD queue	OFGT, DCNT

TABLE B (Contd)

WQUE GLOBAL SUBROUTINES

GLOBAL	PRIMARY FUNCTION	REFERENCING PIDENTS
WQRCXA	Removes centrex attendant from queue	ADPT
WQRVFE	Removes item from fixed length queue	ADPT
WQRVLK	Removes register from POB queue	TLTD, TAND, SARG, RRSP, RING, RATA, QWAT, QTRK, QTAL, ORDL, MIRV, ICAL, FANS, DCNT, CXOR, COIN, CAMA, ADPT
WQRVRR	Removes register from ringing register queue	QAPR, ORDL, ICAL ADPT
WQUAMA	Unloads AMA queue	AMAC
WQUBDT	Unloads blocked dial tone queue	ECMP
WQUCL1	Unloads class of service tone 1 queue	ECMP
WQUCL2	Unloads class of service tone 2 queue	ECMP
WQUCNR	Unloads coin control circuit queue	ECMP
WQUCN8	Unloads coin zone operator queue	ECMP
WQUCN9	Unloads local coin operator queue	ECMP
WQUCXA	Unloads a centrex attendant from queue	CXKY
WQUDR0	Unloads TOUCH-TONE® receiver queue	ECMP
WQUDR1	Unloads customer dial pulse receiver queue	ECMP
WQUDR2	Unloads MF receiver queue	ECMP
WQUDR3	Unloads revertive pulse receiver queue	ECMP
WQUDR4	Unloads trunk dial pulse receiver queue	ECMP
WQUDR5	Unloads TOUCH-TONE® receiver queue	ECMP

TABLE B (Contd)

WQUE GLOBAL SUBROUTINES

GLOBAL	PRIMARY FUNCTION	REFERENCING PIDENTS
WQUIOC	Unloads incoming overload control queue when overload is in process	AOVD
WQUIQS	Prepares to unload incoming overload control queue when queue is short (no incoming overload in process)—followed by WQUIOC	AOVD
WQUPOB	Unloads POB queue	ECMP
WQURRR	Unloads regular ringing register queue	ECMP
WQURRS	Unloads special ringing register queue	ECMP
WQUTTM	Unloads trunk-to-trunk path memory queue	ECMP
WQUTTT	Unloads trunk TOUCH-TONE® receiver queue	ECMP

the queue word (second word) of the CR as the queueing medium and generally requires a two-word HC. When more than one client or CR is on the queue, the first word of the queue HC contains the client register address (CRA) of the first client on the queue, and the second word of the HC points to the last client which has the zero end indication in its queue word. The second client on the queue has its CRA stored in the queue word of the first CR. The remaining clients on the linked CR type VLQ are ordered in the same manner with the prior client's queue word containing the next client's CRA. The queue indicator (QI) bit in the state word of the CR contains a 1, indicating that the CR is on a queue. A list of VLQ-CRA queues administered by WQUE is given in Table C.

3.05 The linked PMT word queue is a list of PMT words which have been linked together (via an algorithm giving the address of the PMT word) to await availibility of resources. The queueing medium for this type of queue is the PMT word associated with the trunk and a special code in its upper bits to identify the queue type. A three-word HC is generally required for the linked PMT queue. The first word contains the trunk network number (TNN) of the first entry on the queue. The second word is a limit counter which is incremented when a TNN is put on the queue and decremented when a TNN is removed from the queue. When the counter reaches its limit, the client requesting loading receives a queue full indication. (***Note:*** For trunk receiver queues, which are VLQ-PMT queues, the limit counter is included in the HC, resulting in a three-word HC. For ringing register queues, which are VLQ-CRA queues having two-word HCs, the limit counter is not a part of the HC.) The third word of the queue HC contains the TNN of the last entry on the queue. The PMT of the first entry contains the TNN of the second entry, and the remaining entries on the linked PMT type VLQ are ordered in the same manner with the prior PMT word containing the TNN of the next item on the queue. The last PMT word contains zeros in its lower bits and the queue code in the upper bits. A list of VLQ-PMT queues administered by WQUE is given in Table C.

Loading an Item on a Queue

3.06 Generally, when a call processing client program enters WQUE to load an item on either a fixed length or variable length queue, the associated executive control main program (ECMP)

TABLE C

QUEUES ADMINISTERED BY WQUE

NAME AND TYPE
FIXED LENGTH QUEUES
Blocked Dial Tone
Customer Dial Pulse Receiver
Line TOUCH-TONE® Receiver
VARIABLE LENGTH QUEUES—CRA TYPE
AIOD
AMA
Centrex Attendant
Class of Service Tone Trunk 1
Class of Service Tone Trunk 2
Coin Control Register
Coin Zone Operator
Local Coin Operator
POB
Regular Ringing Register
Special Ringing Register
Trunk-to-Trunk Path Memory
VARIABLE LENGTH QUEUES—PMT TYPE
Incoming Overload Control
Multifrequency Receiver
Revertive Pulse Receiver
Trunk Dial Pulse Receiver
Trunk TOUCH-TONE Receiver

flag bit for that queue is set to 1, indicating to ECMP that an item is now on a specific queue. With this flag bit set, ECMP returns on a regular basis to the unload entry for the queue until the queue is emptied, at which time the flag bit is set to zero. In addition, the sign bit of the idle list HC for either the equipment or register is set to inhibit seizures and reserve the equipment or register for items on the queue when they are unloaded.

3.07 The incoming overload control queue is one of the queues loaded and unloaded by WQUE for which no ECMP flag is set. This queue is administered via pident AOVD in conjunction with the trunk seizure and answer hopper (see Section 231-045-100 for a detailed description).

3.08 The first entry made on a receiver queue causes a shortening of permanent signal-partial dial (PSPD) timing as well as a reduction in maximum receiver holding time for incoming trunks. A transfer is made from WQUE to AOVD global AOFERQ where conditional print and shortened PSPD timing flags are activated and the appropriate bits are set in a receiver queue overload register.

3.09 When a receiver queue is establishd, a 100-second timer is started. If the queue overflows, a check is made every 10 seconds via an ECMP entry into AOVD at global AOARQ0 to determine if the queue has existed for at least 100 seconds. If so, a receiver queue overload exists and the 100-second flag is set in the associated receiver queue overload register. Return is made to ECMP. When WQUE attempts to load another entry on the full receiver queue whose overload register has timed out, transfer is made to AOVD global AOVFIQ to initiate overload actions.

3.10 WQUE routines for loading queues are listed in Table B.

Unloading an Item From a Queue

3.11 After an item has been loaded on a queue, ECMP returns periodically from the associated base-level frequency class to the unload entry for the queue. For the blocked dial tone queue, the queue entry remains on the queue for a 2-second period before the item is unloaded for an attempted retrial. For receiver queues, WQUE unloads an entry from the queue only if both a peripheral order buffer (POB) and a receiver are available. For all other queues, an entry is unloaded if the appropriate resource is available. After an entry is unloaded from a queue, transfer is made to the appropriate client program to process the item. If no entry can be unloaded, the sign bit of the idle list HC associated with that queue is set. Removal is attempted on the next unload entry by ECMP. When a specific queue is emptied, the associated ECMP flag bit is zeroed to eliminate further ECMP entries. For receiver queues, WQUE informs AOVD that the queue is empty via transfer to AOVD global AOEMTQ.

3.12 WQUE routines for unloading queues are listed in Table B.

Removing an Item From a Queue

3.13 As a result of an abandon, a time-out, etc., a client can be removed from a queue before

ECMP unloads the entry. In such cases, an entry on an FLQ is removed by zeroing the location in the queue. Linkage to the location is established via the client register's queue word which contains the address of the client register entry on the FLQ. An entry on a VLQ-CRA queue is removed from the queue by searching from the beginning of the list until the CRA to be removed matches the contents of the queue word of one of the registers on the list. At this point, the list is updated by substituting the next CRA on the list for the one just removed from the queue word. Removal of an entry from a VLQ-PMT queue is administered in a manner similar to that of the VLQ-CRA queue. The items to be removed are TNNs in PMT words rather than CRAs in queue words.

3.14 WQUE routines for removing an item from a queue are listed in Table B.

QTL QUEUEING

A. General

3.15 The Queueing for Trunks and Lines (QTL) feature provides a means for automatically queueing calls to a multiline hunt group or a group of trunks. Queueing is accomplished by the use of queueing registers. Each multiline hunt group has one QTL queue associated with it. Each call is placed in queue and waits its turn (first-in, first-out basis). While waiting, the calling party receives audible ringing tone which can be followed by a number of delay announcements.

3.16 With an automatic call distribution (ACD) setup (a type of multiline hunt), a multiline hunt group can be divided into functional subgroups (splits) that perform a particular function (eg, general reservations, mainline, cargo, etc.) Each functional group must be assigned a QTL queue. The queue is accessed by means of one or more listed directory numbers. Calls are placed in queue, provided with audible tone, and unloaded on a first-in, first-out basis (unless priority queueing is available). As a QTL queue for a functional group fills up, the delay time for each additional call increases. When the delay time of the oldest call in queue exceeds a predetermined time, the queue may attempt to alternate route calls to other queues. Alternate routing may be accomplished via alternate server intraflow/interflow (ASI). ASI places calls only on their primary QTL queue but allows them to be serviced by agents associated with other queues belonging to the same alternate server pool (ASP).

3.17 Outgoing WATS simulated facilities may queue via the Outgoing Trunk Queueing (OTQ1) feature. While the call waits in queue, OTQ1 attempts to complete the call via the queued-for WATS band or, if applicable, via an alternate WATS band. If an appropriate WATS facility does not become idle within a preset time limit, the call is either routed via the DDD network or to overflow tone.

3.18 For additional details on queueing for multiline hunt, ACD, and WATS facilities, refer to Section 231-090-167, Section 231-090-339, and Section 231-090-408, respectively.

B. Functional Description

3.19 The process of queueing calls for the facilities described above may be divided into five stages: PRE, TO, ON, OFF, and POST queue. The actions taken during each stage and the QTL pidents involved are listed in Table D.

3.20 Following is a brief functional description of each of the QTL pidents.

QAPR—Queue and Administration Processing

3.21 Pident QAPPR contains basic routines responsible for determining if a call should go on queue and if there is room on the queue for a call. In addition, these routines start the process to give tone to the customer. QAPR also contains general queue unloading routines. Other routines, which are entered from ECMP every flagged class D, perform ON queue processing functions, determining if there is an idle agent and, if so, taking a call off the queue and terminating the call to the idle agent. Additional routines detect time-out and disconnects.

QCDL—Queueing Data Link Automatic Call Distribution Interface

3.22 This is a general purpose ACD pident which performs the following functions:

- Processes data link orders sent to the ACD agent console

TABLE D

QTL QUEUEING

STAGE	ACTIONS TAKEN	QTL PIDENTS INVOLVED
PRE QUEUE	• Checks to see if incoming facility is allowed to queue • Checks for availability of queue • Seizes QTL queueing register • Performs audits	QAPR, QWAT, QCIA
TO QUEUE	• Physically loads call on QTL queue, making pointer and counter updates, etc. • Places call loading time in queueing register • Informs customer of status of call via delay announcements [music, overflow tone, audible ring, silence, confirmation tone (WATS), etc.] • Performs audits	QTAL, QAPR, QSIF, QWAT, QCIA
ON QUEUE	• Performs timing on call • Determines if call can be served • Decides, on basis of give-help/need-help state of queues, whether to interflow • Determines if length of time on queue warrants different delay announcement • Detects availability of facilities [eg, idle agent, idle OGT, idle simulated facilities register (WATS)] • Updates call waiting lamps • Performs audits	QTAL, QTRK, QEDA, QSIF, QCDL, QCIA

TABLE D (Contd)

QTL QUEUEING

STAGE	ACTIONS TAKEN	QTL PIDENTS INVOLVED
FROM QUEUE	• Physically takes call off the queue (interfaces with ECMP flagged class D work) • Performs necessary initialization for attendant line or OGT (eg, seizes OR) to set the call up to appear as normal call again (as if call had never queued) • Removes queueing register for attendant terminating call (eg, ACD or multiline hunt) • Makes final connections (eg, initiates ringing of attendant line or outpulsing of outgoing call) • Performs audits	QAPR, QTRK, QTAL, QWAT, QCIA
POST QUEUE	• Removes queueing register for OGT call (eg, WATS) • Enters outpulsing routines for normal call (ORDL for L-T connection and TAND for T-T connection) • Performs audits and customer updates	QTRK, QCIA

- Administers call waiting lamps for ACD
- Sends ACD status updates to CRT
- Performs some audit functions.

QCIA—Customer Interface and Special Auditing Routines

3.23 Pident QCIA performs two basic functions for ACD:

- Detects and processes state changes in ACD-QTL local night service key and inhibit intraflow key
- Processes customer changeable intraflow triggers.

In addition, QCIA determines the oldest call on the queue and performs special auditing functions.

QEDA—Queue Entry and Destination Assignment Routines

3.24 Pident QEDA contains the basic logic for intraflow decisions (eg, determining which agents need help and which agents can give help in an alternate server arrangement). In addition, QEDA administers ACD priority calling and local night transfer features.

QSIF—Queue State Information Features

3.25 This QTL program places call loading time into the associated queueing register. Other functions include the following:

- Performs basic timing and switching for delay announcements
- Determines states and sends lamp orders connected with the Call Waiting Lamps feature.

QTAL—Give Audible, Disconnect, and Line Termination Routines

3.26 QTAL contains routines which, depending on the type of origination, call the proper routines to perform switching, determine what type of tone or announcement the customer gets, and finish putting the call on queue. For a call that fails to get the necessary facilities when coming off the queue, this program contains routines to give reorder to the customer given the type of incoming facility. Other routines prepare the call coming off the queue terminating to an intraoffice call to flow into the normal legs as if the call were never queued. There are a number of general routines that perform such functions as unlinking and releasing ringing registers and selecting audible route.

QTRK—Terminate to Trunk Facility Subroutines

3.27 This QTL pident deals primarily with calls terminating to an OGT facility. The basic functions performed by QTRK are as follows:

- Detects available OGT facilities when interflowing
- Performs interflow timing
- Initializes outgoing trunks
- Calls outpulsing routines.

QWAT—Queueing for WATS

3.28 Pident QWAT performs all PRE, TO, and FROM queueing functions (Table D) for wide area telephone service (WATS) that pidents QAPR and QTAL do for ACD and basic multiline hunt groups.

GENERAL PURPOSE PROGRAMS

A. CHGD—Scan Point Change Director Program

3.29 This program processes hopper entries resulting from changes in scan points. The hoppers administered by CHGD are as follows:

(a) Hit scan result hopper—loaded by the timed scan junior register scan program (DITS)

(b) Trunk seizure and answer hopper—loaded by the supervisory scan program (CCLT) and unloaded by the automatic overload control program (AOVD)

(c) Line ferrod scan hopper—loaded by the line ferrod scan program.

3.30 CHGC determines which program is to process the state change represented by the scan point change and transfers control to that program.

Hit Scan Result Hopper Unloading

3.31 Every class B execution, the executive control main program (ECMP) enters CHGD via global DIHSRH to unload the hit scan result hopper. Each hopper entry contains a timed scan junior register (TSJR) address, scanned number, and scan result (a disconnect or a hit). The scanner number is examined to determine the type of scanner entry being unloaded from the hopper and, consequently, the program that is to receive control.

3.32 If the scan point is on either a line junctor or an intraoffice trunk in a line-to-line path with no associated call register, control is transferred to the disconnect program (DISC). If the scan point is on a trunk circuit and the trunk path memory indicates that a call register is associated with the trunk circuit, control is transferred to the proper program via the register identification-program tag (RI-PT) disconnect method (global YRIPTD in YMRG).

3.33 If the scan point is not on a line junctor or on an intraoffice trunk in a line-to-line path and is not associated with a call register, control is directed by indexing into one of three transfer order tables. The index [special program index (SPI), trunk program index (TPI), or nontrunk program index (NTPI)] is obtained from a translation based on the scanner number [global TRTSNU in translation routines-basic trunk (TRTB) for a universal trunk scanner number and global TRMSNA in TRTB for a master scanner number]. Transfer table CHSP01 is used if the scan point is on a universal trunk frame circuit; table CHTP01, if on a master scanner trunk circuit; and table CHNT01, if on a master scanner nontrunk circuit.

3.34 If no path is associated with an intraoffice trunk and the trunk is on the trunk high and wet (THAW) list, transfer is made to the

trunk list maintenance program (TNLS) to remove the trunk from THAW. If the trunk is not on THAW, transfer is made to the disconnect program (DISC) to release the TSJR.

3.35 If no path is associated with an interoffice trunk, a check is made to determine if the trunk is on a queue with supervision on. If so, transfer is made to the queue administration program (WQUE); otherwise, the type of trunk (eg, 1A252, 1A203, traffic service position) being processed determines which program receives control to either remove the trunk from THAW or release the TSJR.

3.36 When a hit scan result hopper overflow occurs, the automatic overload control program (AOVD) enters CHGD via global DIHSRR to unload the hopper in interject. Unloading is accomplished as explained earlier for ECMP class B entry processing.

Trunk Seizure and Answer Hopper Unloading

3.37 Every class C execution, AOVD enters CHGD via global DCUTK1 to process the trunk seizure and answer hopper (TSAH) entries unloaded by AOVD. Each hopper entry contains the scanner number of a ferrod whose state changed. The scanner number is examined to determine which program is to receive control.

3.38 Control is transferred by indexing into one of three transfer order tables. The index (SPI, TPI, or NTPI) is obtained from a translation based on the scanner number. Transfer table CHSP10 is used if the scan point is on a universal trunk frame circuit; table CHTP10, if on a master scanner trunk circuit; and table CHNT10, if on a master scanner nontrunk circuit.

3.39 When a TSAH oveflow occurs, AOVD unloads the TSAH entries in interject, transferring to CHGD global DCUTK1 to process each entry.

Line Ferrod Disconnect Hopper Unloading

3.40 Upon every class D execution, ECMP enters CHGD via global OGUNLH to unload the line ferrod disconnect hopper. This hopper is used by multibit scan registers to make reports back to base level. Each hopper entry contains the address of the junior register word that contains the address of the call register associated with the line being scanned. As each hopper entry is unloaded, control is transferred to the proper program via the RI-PT standard method (global YRIPTS in YMRG).

3.41 When a line ferrod disconnect hopper overflow occurs, AOVD enters CHGD via global OGUNL1 to unload the hopper in interject.

B. COPR—Report and Miscellaneous Subroutines

3.42 Pident COPR consists of numerous subroutines associated with call register processing and answer timing and processing. Functions performed when client programs enter COPR at the appropriate global entries include the following:

(a) Linking, unlinking, and releasing call registers and reporting changes of state

(b) Administering client register disconnect

(c) Granting or denying clearance to requesting nonmaster register

(d) Setting supervisory scan points following a successful POB execution

(e) Establishing disconnect supervision on L-L, L-T, or T-T connection

(f) Removing registers from timing lists

(g) Answer timing for call register associated calls.

Linking, Unlinking, and Releasing Call Registers and Reporting Changes of State

3.43 COLINK—This routine links, in a circular linked list of call registers, two registers specified herein as A and B. The client register denoted by A is to be linked to register B. Register A may or may not be already linked; B cannot be already linked. COLINK sets the link and the master bits in the registers as required. The linked list is then updated by writing the address of the register specified by B into the link word of the register specified by A and the address of the register that previously followed A (or A itself if it was not already linked) into the link word of B. A is made the master register if it was not previously linked.

3.44 YHUNLK—This routine removes the client register from a circular linked list of call registers and updates the list as required. The link bit of the register to be unlinked is reset. If this register was the master, the master bit of the next register in the linked list is set. The linked list is then updated by writing the address of the register which was in the link word of the unlinked register into the link word of the preceding register.

3.45 YHRPRT—This routine is entered when a client register initiates a change-of-state report. If the client or reporting register is the master register on the circular linked list, the report is passed to the next register on the list through the proper slot in that register's PT table. The RI-PT routines in pident YMRG are used to determine the proper PT entry. Transfer to the appropriate RI-PT routine from YHRPRT is made on the basis of the change-of-state report code provided upon entry to YHRPRT. If the reporting register is not the master, the master register is located and the change of state is reported to the master, using the appropriate RI-PT routine. If the reporting register is not linked, the appropriate RI-PT routine is accessed and the report comes back through the proper slot in the reporting register's PT table.

3.46 YHRLRX—This routine releases the client register from a circular linked list of call registers and restores the register to its idle linked list via a transfer to global YCRONE in pident YMRG. YCRONE is discussed elsewhere in this document.

3.47 YHRPMS—This routine is entered when a client register initiates a change-of-state report that is to be started around a linked list of call registers via the master register on the list. If the register reporting to the master register is not linked to the list, the report comes back through the proper slot in the reporting register's PT table. Otherwise, the master register is located, and the report is given to the master register via the proper slot in the master register's PT table. In either case, the RI-PT routines in pident YMRG are used to determine the proper PT entry. Transfer to the appropriate RI-PT routine from YHRPMS is made on the basis of the change-of-state report code provided upon entry to YHRPMS.

3.48 YHPASS—This routine is entered when a client register initiates a change-of-state report that is to be passed to the next register on a linked list of call registers. YHPASS uses part of subroutine YHRPRT (local YHCC) to determine what change of state is being reported and to what register the report must go. The change is then reported through the proper slot in that register's PT table via the appropriate RI-PT routine. If the client register is not linked to the list, the report comes back through the proper slot in the client register's PT table via the appropriate RI-PT routine.

3.49 COPOUT—This routine releases the client register from a circular linked list of call registers and passes on a report of a change of state to the next register on the linked list. COPOUT uses three of the COPR global routines described earlier to accomplish its purpose: (1) YHUNLK to unlink the client register, (2) YHRLRX to release the register, and (3) YHPASS to pass on the change-of-state report.

3.50 COPSRX—This routine unlinks the client register from a circular linked list of call registers and passes a change-of-state report to the next register on the linked list. COPSRX uses two of the COPR global routines described earlier to accomplish its purpose: (1) YHUNLK to unlink the client register and (2) YHPASS to pass on the change-of-state report.

3.51 COUNRE—This routine unlinks the client register from a circular linked list of call registers and restores the client to its idle linked list. COUNRE transfers to COPR global YHUNLK to unlink the client register and to YMRG global YCRONE to restore the register to its idle linked list.

Administering Client Register Disconnect

3.52 COPR global CORLSN checks the status of the client register and initiates the required actions to disassociate the register from the call. If the given register is linked but is not the master register on a linked list of call registers, it is assumed that the given register has received a change-of-state report and is to be removed from the call. In this case, transfer is made to COPR global COPOUT to release the given register and pass on the report that it received. If the register is linked and is the master, it is unlinked via

transfer to COPR global YHUNLK. The associated path memory is copied into the next register on the linked list via transfer to the appropriate change-in-network (CIN) routine in pident NCIN.

3.53 If the client register is not linked, the associated path memory is copied into a path-memory-for-lines (PML) or path-memory-for-trunks (PMT) table via transfer to the appropriate CIN routine.

Granting or Denying Clearance to Requesting Nonmaster Register

3.54 The following routines are accessed by client programs when a nonmaster register wants control and path memory. The requesting nonmaster register is either granted or denied clearance based on whether supervision on the call is stable or unstable.

(a) COERCE—This routine sets supervision (J bits or T bits) of a line-to-line or trunk-to-trunk call to ignore if supervision is stable. Transfer is made to COPR global COSA (explained below) to make the requesting register the master register. Return is made to the clearance return address (contents of Z register). If supervision of the call is unstable, supervision is left in the accept state, and the clearance request is denied via transfer to local routine DENY from which control passes to COPR globals CODENZ and CODENY (explained below).

(b) COSA—Client programs enter COPR global COSA to make the requesting register the master register on a linked list of call registers and to reset the linkage. Basically, this routine accomplishes the above functions via transfers to other COPR globals:

(1) YHUNLK—Unlinks requesting nonmaster register

(2) COLINK—Links requesting nonmaster register to right of master register

(3) CORLSN—Unlinks the master register, makes requesting nonmaster register (register to the right) the new master, and gives it path memory

(4) COLINK—Links old master register to right of new master.

The request waiting (RW) bit of the old master register is zeroed. If the prior state of the RW bit was 0, immediate clearance is granted to the new master; otherwise, delayed clearance is given after a real-time break.

(c) CODENY—Client programs enter COPR global CODENY when the requesting nonmaster register is to be denied clearance because supervision on the call is unstable. The RW bit in the master register is set. The requesting nonmaster register is first unlinked and then linked to the right of the master register via transfers to COPR globals YHUNLK and COLINK. Return is made to CODENY which transfers control to the clearance return address minus 1 (set up to handle denials).

(d) CODENZ—This routine is the same as CODENY with one exception. The sign bit of Z (clearance return address) is set to inform the centrex trunk preemption program to rerequest clearance.

Setting Supervisory Scan Points Following Successful POB Execution

3.55 Upon return from a successful POB execution with a line-to-line, line-to-trunk, or trunk-to-trunk connection, the appropriate COPR global routine (see Table E) is entered to set the applicable supervisory scan points. Basically, each of these COPR routines consists of a series of transfers to some or all of the following types of external routines:

(a) Translation routines—provide the junctor or trunk scanner number information required for setting the scan points

(b) L-, J-, and T-bit administration routines—set the states of line, junctor, and trunk supervisory bits

(c) Network transition routine—provide the TNN of trunk 2 in T-T connection

(d) POB execution routine—idles the POB.

Following idling of the POB, each of the referenced globals (Table E) transfers control to the main program via a transfer to the top of the stack with a check for interject. If a hardware failure is encountered during execution, control is immediately

transferred to COPR local HF1. If there is a call register associated with the call, HF1 initiates a hardware failure report via COPR global YHRMHF. If there is no call register, return is made to the main program.

Establishing Disconnect Supervision on an L-L, L-T, or T-T Connection

3.56 Client programs enter COPR at global COSUPJ to set up disconnect supervision on a line-to-line, line-to-trunk, or trunk-to-trunk connection. Checks are made on the path memory format indicator (PMFI) to determine the type of connection or call prior to setting scan points. PMFI values of 19, 7, and 1 indicate L-L, L-T, and T-T paths, respectively. Based on the PMFI value, the appropriate COPR local routine (Table F) is entered to set up disconnect supervision. Basically, these routines consist of transfer to the same types of external routines explained in paragraph 3.55, with the exception of the POB execution routine. Control is returned to the client program.

Removing Registers from Timing Lists

3.57 COEND1—Client programs enter global COEND1 to remove a register from a one-way timing linked list of call registers. The queue indicator bit of the client register (register to be removed) is zeroed. A search is made through the linked list to locate the client register. If the register is not found, return is made to the client program. If the register is found, checks are made to determine if call store word Y6NEXT (the address of the next register to be processed) points to the client register. If so, the address of the register to which the client register is linked (forward link only in a one-way linked list) is stored in Y6NEXT. Linkage to the processing head cell is updated, thus removing the client register from the timing linked list. Return is made to the client program.

3.58 COEND2—Client programs enter global COEND2 to remove a register from a two-way timing linked list of call registers. The queue indicator bit of the client register (register to be removed) is zeroed. Linkage of the register to the right (register whose address is in the client queue word) and linkage of the register to the left (register whose address is in the client scan word) of the client register are verified. If either linkage is bad, the requesting register is not removed and return is made to the client program. If the linkage is good, checks are made to determine if call store word Y6NEXT (the address of the next register to be processed) points to the client register. If so, the address of the register to the right of the client register is stored in Y6NEXT. Linkage of registers to the right and left of the client register is updated, thus removing the client register from the timing linked list. Return is made to the client program.

Answer Timing for Call Register Associated Calls

General Answer Processing

3.59 Answer processing begins when a call has been placed in a stable configuration: the originating side has disconnect (on-hook) supervision turned on, the terminating side has answer (off-hook) supervision turned on, and the associated call register has its program tag (PT) set to a value that conforms to the required PT setup for answer timing (Table G). An off-hook report from the terminating side of the call initiates the answer processing sequence. The off-hook time is stored in the scan word of the call register, the PT is incremented by 4, and the call register is put on a 2-second one-way timing linked list, waiting for a possible on-hook signal from the terminating side. For a T-T path, the off-hook signal is sent on the incoming trunk before putting the call register on timing.

3.60 Following is an example of possible actions occurring at this point:

(a) The call register times out before any reports are received from the supervisory scan programs. A true answer report is made to the client. Disconnect supervision is on for both sides of the call.

(b) A disconnect report is received from the originating side. (If an originating side disconnect is received prior to the terminating side off-hook, the call is passed to COPR global COABN1.) The call register is removed from timing, a begin-answer-timing (BAT) report is made if an automatic message accounting (AMA) register (linked or master) is associated with the call, and control is passed to the disconnect program (DISC) via DISC globals DICLDR or DITIDM.

TABLE E

SETTING SUPERVISORY SCAN POINTS FOLLOWING SUCCESSFUL POB EXECUTION

GLOBAL ENTRY	TYPE OF CONNECTION	BITS SET	STATE	INTERFACE ROUTINES
COLLSF	LINE-TO-LINE	J0 = 0	BUSY	NEJNNS (Junctor Translation)
		J1 = 0	BUSY	YACJ00 (J-Bit Administration)
				YACJM0
				PQIDWL (POB Execution)
COLT01	LINE-TO-TRUNK	SCAN POINT 0		TRTNNP (Trunk Translation)
		T1 = 0	BUSY	YAASN0 (T-Bit Administration)
		T2 = 1	ACCEPT	PQIDWL (POB Execution)
COLTB0	LINE-TO-TRUNK	SCAN POINT 0		TRTNNP (Trunk Translation)
		T1 = 0	BUSY	YASD2A (T-Bit Administration)
		T2 = 1	ACCEPT	POB execution takes place in QEPR before entry to COLTB0
		SCAN POINT 1		
		T1 = 0	BUSY	
		T2 = 1	ACCEPT	
COLTBD	LINE-TO-TRUNK	SCAN POINT 0		TRTNNP (Trunk Translation)
		T1 = 0	BUSY	YASDOA (T-Bit Administration)
		T2 = 1	ACCEPT	PQIDWL (POB Execution)
		SCAN POINT 1		
		T1 = 1	IDLE	
		T2 = 1	ACCEPT	
COTT01	TRUNK-TO-TRUNK	SCAN POINT 1		YASDIT (T-Bit Administration)
		T1 = 0	BUSY	PQIDWL (POB Execution)
		T2 = 1	ACCEPT	
COTTBD	TRUNK-TO-TRUNK	TRUNK 1		YASDIT (T-Bit Administration)
		SCAN POINT 1		NETAT2 (Network Transition)
		T1 = 0	BUSY	YASORT (T-Bit Administration)
		T2 = 1	ACCEPT	PQIDWL (POB Execution)
		TRUNK 2		
		SCAN POINT 1		
		T1 = 1	IDLE	
		T2 = 1	ACCEPT	

TABLE E (Contd)

SETTING SUPERVISORY SCAN POINTS FOLLOWING SUCCESSFUL POB EXECUTION

GLOBAL ENTRY	TYPE OF CONNECTION	BITS SET	STATE	INTERFACE ROUTINES
COTTB0	TRUNK-TO-TRUNK	TRUNK 1 SCAN POINT 1 T1 = 0 T2 = 1 TRUNK 2 SCAN POINT 1 T1 = 0 T2 = 1	 BUSY ACCEPT BUSY ACCEPT	YASDIT (T-Bit Administration) NETAT2 (Network Transition) PQIDWL (POB Execution)
CORSVL	LINE RESTORE-VERIFIED	L = 0	IDLE	YALENL (L-Bit Administration) PQIDWL (POB Execution)

TABLE F

SETTING UP DISCONNECT SUPERVISION ON A CONNECTION

GLOBAL ENTRY	TYPE OF CONNECTION	BITS SET	STATE	INTERFACE ROUTINES
COSUPJ (SULL)	LINE-TO-LINE	J0 = 0 J1 = 0	BUSY BUSY	NEJNNS (Junctor Translation) YACJ00 (J-Bit Administration) YACJM0
(SULT)	LINE-TO-TRUNK	SCAN POINT 0 T1 = 0 T2 = 1 SCAN POINT 1 T1 = 0 T2 = 1	 BUSY ACCEPT BUSY ACCEPT	TRTNNP (Trunk Translation) YASD2A (T-Bit Administration)
(SUTT)	TRUNK-TO-TRUNK	TRUNK 1 SCAN POINT 1 T1 = 0 T2 = 1 TRUNK 2 SCAN POINT 1 T1 = 0 T2 = 1	 BUSY ACCEPT BUSY ACCEPT	NETAT2 (Network Transition) YASDIT (T-Bit Administration)

(c) An on-hook (not a bit) report is received from the terminating side and the call register is taken off timing. A BAT report is made if an AMA register (linked or master) is associated with the call. The on-hook report is repeated on the incoming trunk if this is a T-T path. If the call is L-L or incoming trunk-to-line (ICT-L), control is passed to DISC via global DICLDR or DITIDM. If the call is line-to-outgoing trunk (L-OGT) or T-T, answer supervision is restored on the call and no report is made to the client.

3.61 ***Input Requirements:*** Answer processing for call register associated calls has the following input requirements:

(a) The call register PT table must contain the entries or routines listed in Table G. These routines provide common actions for call registers prior to answer and are entered via transfers to the applicable slot in the PT table of the associated call register.

(b) The call register (if it is master) must have valid path memory for a L-L, L-T, or T-T path.

(c) The following items must be initialized in the call register:

- Y4MR, Y4LI, Y4PMFI, Y4RI, Y4PT, and Y4LINK must be set to correct values.
- Y4PMAD must equal 0.
- Y4QUE and Y4SCAN must be available for use.
- Y4TS must equal state of ICT if path is T-T.

3.62 ***Output Requirements:*** Answer processing for call register associated calls has the following output requirements:

(a) Answer report when associated register is not master (AAAAAA entry in call register PT table)—Output conditions are provided by the master register program making the answer report.

(b) Abandon before answer (BBBBBB entry in call register PT table).

(1) If the associated register is not master, output conditions are set by the master register making the abandon report.

(2) If the associated register is master, all scans are off and there are no hopper entries. The TSJR contains the inactive code and is linked to the master register (the address of the second word of the TSJR is stored in Y4SCAN). The abandon has been reported to any linked registers.

(c) Answer report when register is master (CCCCCC entry in call register PT table).

(1) Disconnect supervision is active on both the originating and terminating terminals.

(2) Answer has been reported to any linked registers.

(3) Charge timing (2 to 2.1 seconds) has been done.

(4) If the path is T-T, answer has been signaled on the incoming trunk and Y4TS has been set to 1 to indicate this off-hook state.

(5) If the terminating party on an L-L or L-OGT call goes on-hook then back off-hook after charge timing is initiated, answer will not have been reported. When a report of flash or talking-resumed is made by the disconnect program [see (f) and (g) below], the call register must be able to determine whether or not answer is to be reported.

(6) When answer is reported, the answer time is present in location E4ITOD. Answer time is system time at the initial off-hook.

(d) Off-hook hit processing during answer timing.

(1) L-L or ICT-L control will be transferred to DISC global DITIDM or DICLDR if AMA is associated with the call.

(2) L-OGT answer supervision is restored on the OGT. No report is given to the client, but AMA is notified of BAT if AMA is associated with the call.

TABLE G

PROGRAM TAG TABLE SETUP FOR ANSWER TIMING

BLOCK NO.	ENTRY	TRANSFER TO	STATE OF CALL
1			Initial state when ringing or waiting for answer
	V.Y4PT-2	COABN1	Abandon before answer
	V.Y4PT-1	—	Vacant slot
	V.Y4PT+0	ANSWER	Answer before abandon
	V.Y4PT+1	AAAAAA	Answer report—register not master
2			Register in charge timing or on POB queue
	V.Y4PT+2	BBBBBB	Abandon report
	V.Y4PT+3	ANSWTD	End of timing or hit scan result
	V.Y4PT+4	CCCCCC	Answer report—register is master
	V.Y4PT+5	ANSWPQ	Off POB queue to signal answer on ICT
3, 4, 5, 6, 7			Disconnect timing provided
	V.Y4PT+6 through V.Y4PT+25	Routines in pident DISC (PR-6A134)	(Described in pident DISC at global entry DITIDM)

(3) T-T answer supervision is restored on the OGT. Answer will be removed from the ICT if applicable. No report is made to the client, but AMA is notified of BAT if AMA is associated with the call.

(e) Disconnect after answer.

(f) Talking resumed when terminating party on-hook less than 10 seconds after answer.

(g) Flash by originating party.

3.63 When an on-hook (not a hit) is detected during charge timing, 8 is added to the initial value of Y4PT, charge timing is terminated, and control is transferred to the disconnect program (global DITIDM). Control is returned to one of three exits [(e), (f), or (g) above].

Special Answer Processing

3.64 Whereas general answer processing requires the use of the 2-second off-hook hit protection interval and restoral of answer supervision if an off-hook hit should occur on the OGT of a L-OGT or T-T call, some clients still employ the COANS1 special answer timing code to administer calls involving charge timing. This code provides for 600 to 800 ms timing, specifies the answer time as the time after timing has occurred, and does not restore answer supervision on T-T and L-OGT calls if BAT occurs but passes the call onto DISC to begin time release disconnect timing.

3.65 ***Input Requirements:*** Input requirements for special answer processing are the same as those listed earlier for general answer processing with one exception. Routines COANS1, CODCTO, and COPOBQ replace entries ANSWER, ANSWTD, and ANSWPQ, respectively, in blocks 1 and 2 of the PT table (Table G).

3.66 ***Output Requirements:*** Output requirements for special answer processing are the same as those listed earlier for general answer processing, with the following exceptions:

(c)(3)	Timing is 600 to 800 ms.
(c)(6)	Answer time is system time at repeat of answer.
(d)(1)	Not applicable.
(d)(3)	Not applicable.

C. CXYH—Seize and Release Routines and L, J, and T Bit Administration for Centrex

3.67 CXYH performs the same functions for centrex service that YAHA (described elsewhere in this document) performs for noncentrex service. Certain features provided to centrex groups use particular call registers with special word layouts. CXYH is responsible for seizing and releasing four types of registers when they are needed on a call. These registers, the global entry points for seizing and releasing them, and the pidents referencing these entry points are listed in Table H.

3.68 Line, junctor, and trunk bit administration for centrex is administered by pident YAHA.

D. TRCE—Call Trace

3.69 The primary purpose of pident TRCE is to find the terminal (line, trunk, or service circuit), if any, which is connected to a known terminal. On failure to find the sought terminal in a connected path, TRCE gives an indication of the state of the known terminal in the system.

3.70 TRCE is requested by the following types of programs:

(a) Operator No-Test-Program—to permit the operator to gain access to the subscriber line (even if it is busy) and to perform either talking or monitoring functions.

(b) Trunk and Line Test Panel Programs—to monitor a line or a trunk.

(c) Maintenance Control Program—(1) to perform routine checking of lines and/or trunks when traffic in the office is low and (2) to find the junctor network number used when a restore-verify failure occurs.

(d) Call Waiting Program—to determine the path of a busy phone.

TABLE H

CXYH SEIZE AND RELEASE REGISTER ROUTINES

REGISTER TYPE	GLOBAL ENTRY		REFERENCING PIDENTS
	SEIZE	RELEASE	
CUSTOMER FACILITY GROUP	YASCFG	YARCFG	CFGR YMRG, SARG, CFGR
6 PORT CONFERENCE	YASZC6	YARLC6	CXKY YAHA, SARG, CXKY
CENTREX LOOP	YASZLP	YARRLP	SMAC, RING, CXLO, CXKY, ADUX, ADPB, ADDX YMRG, SARG, CXKY, CNLP, ADPB, ADDX, ADBN
SIMULATED FACILITIES	YASZSF	YARLSF	QTRK, ISIG, CXSF YMRG, SARG, QTRK, ISIG, CXSF

(e) Teletypewriter Program—to process associated input message requests.

Refer to Table I for a listing of TRCE global subroutines, their primary functions, and pidents transferring to TRCE at these global entry points.

Functional Description

3.71 Pident TRCE is functionally divided into two routines: NETRCL and NETRCT. NETRCL is used to find the terminal connected to or reserved for a line and NETRCT, the terminal connected to or reserved for a trunk.

Call Trace for Lines

3.72 This input to subroutine NETRCL is a valid line equipment number (LEN) which is stored in the first word of a block of memory used exclusively for trace. This block of memory contains all the pertinent information gathered by TRCE.

3.73 To determine the state of the given line, TRCE transfers to YAHA subroutine YALENL. If the line is idle, TRCE returns to the client program unless the trace was requested via the TTY. In this case, the appropriate output message (CT01 for L-L connection and CT02 for L-T connection) is printed and transfer is made to the executive control main program (ECMP). If a client register is associated with the path, an NE06 message is printed before transferring to ECMP. See the Output Message Manual (OM-6A001) for additional details on these output messages.

3.74 If the given line is marked busy, TRCE searches for the LEN in the path memory associated with the given LLN. During the search, the line network tag [(LNT)—rightmost 13 bits of the LEN] is compared with the corresponding bits of each path memory for lines (PML) word in the subtables for path memory for lines (SPML). Path memory is a part of temporary memory where enough information about a connection is stored to enable the system to reconstruct the connection. There are 256 junctor terminals on each line junctor switch frame. Corresponding to each of these is a PML word in either of two formats: (1) when a L-L path is reserved or connected and is not

TABLE I

TRCE GLOBAL SUBROUTINES

GLOBAL ENTRY	PRIMARY FUNCTION	REFERENCING PIDENTS
NETRCL	Finds terminal connected to a line	SADT, PSPD, OFNT, NMFL, NMDT, LTDK, ISIG, CXBV
NETRCT	Finds terminal connected to or reserved for a trunk	TLTC, TBTF, ISIG, CXBV
NETPRT	Finds terminal connected to or reserved for a trunk (same as NETRCT except a bit is set indicating output message is required)	TSPS
NETRCX	Zeros the trace block and performs fast line path trace	WAIT
TTYTRL	Processes NET-LINE- input message	TTIA
TTYTRT	Processes NET-TNN- input message	TTIA, CXKY

linked to a call (or client) register, the primary item of path memory information in the two PML words corresponding to the two junctor network numbers on the line link network (JNNLs) is the LNT; (2) when a L-L path is reserved or connected and is linked to a call register, the primary items of information in the two PML words corresponding to the two JNNLs are the linked call register address (CRA) and the path memory index (PMI). The PMI, when added to the path memory annex (PMA), determines the place where the path memory information is displaced. The PMA is a block of temporary memory, which may be part of the call register, that is used to contain path memory information that is displaced from PML and PMT words.

3.75 If necessary, the search is made through all four SPMLs associated with the given LLN. If a matching PML word of the LNT format having a sign bit of 1 is found, the control bits in the word are examined to determine if the path associated with the LNT is in a reserved state (path in use and no current flowing) or a connected state (path in use and current flowing). If no match occurs with PML words of the LNT format, the search continues through PML words of the CRA-PMI format. If a match with the known LNT is found in the displaced path memory, the same checks are made and the same actions taken.

3.76 If the path is reserved, the search is continued with the next PML word in the SPML table. If the LNT is found in a connected path, the index of the address of the PML word which contains it or points to it identifies the junctor network number of the junctor connected to the given line. This JNN is used as an input to a JNN translation via a transfer to global NEJQJ2 in the junctor translations program (NEJR). This translation yields the JNN of the junctor terminal at the other end of the junctor and indicates whether the input line is connected to another line or to a trunk. Both JNNs in the connected path are stored in the trace block.

Line-to-Line Path

3.77 If the sought terminal is a line, transfer is made to global NEJLCB in the network transition control program (NTWK) to find the sought LEN. Upon return to TRCE, the LEN is stored in the trace block and a CT01 message is printed giving the results of the trace operation. If a call register is associated with the path, an NE06 message is printed. Transfer is made to ECMP.

Line-to-Trunk Path

3.78 If the sought terminal is a trunk, the junctor network number (JNN) of the given line (JNNL) is compared with the contents of each path memory for trunks (PMT) word in the subtables for path memory trunks (SPMT). Each PMT word has the necessary path memory information either in the PMT word or in diplaced path memory (when the path is linked to a call register). When a match occurs, the trunk network number (TNN) corresponding to the address of the PMT word is stored in the trace block and a CT02 message is printed giving the results of the trace operation. If a call register is associated with the path, an NE06 message is printed. Transfer is made to ECMP.

Line on List or Queue

3.79 If TRCE routine NETRCL completes the search through the SPML tables and fails to find the line in a connected path in memory, even though the line bit is marked busy, various queues and lists are examined via a transfer to global NMLIST in the line bit audit program (NMDT). If the input LEN is found on one of these lists or queues (eg, the high and wet list or the blocked dial tone queue), a code identifying the list or queue and the address of where the LEN is found are stored in the trace block. TRCE returns to the client program unless the trace was requested via the TTY. In this case, the appropriate output messages are printed and transfer is made to ECMP.

3.80 If the LEN cannot be found, flags are set for two audits (line bit audit 50 and system audit 46) to investigate and perhaps correct the state of the line bit and path memory. For detailed information on the functions performed by these audits, refer to Section 231-045-215.

Call Trace for Trunks

3.81 The input to subroutine NETRCT is a trunk network number (TNN) which is stored in the first word of a block of memory used exclusively for trace. To determine the status of the given trunk, transfer is made to TNLS subroutine

NMTRST. Upon return from TNLS, TRCE follows much the same procedures as that previously explained under call trace for lines. If the trunk is in a stable and valid path, the program hunts for the other terminal. If the status is other than stable and valid, the code word (containing trunk state options) returned from NMTRST is stored in the trace block, the appropriate status bits are set to indicate a nonstable state, and TRCE returns to the client program.

Trunk-to-Line (Trunk)

3.82 A trunk is said to be in a stable and valid state when it is in a connected or reserved path. If this is a trunk-to-line (T-L) path, the JNN for the line is stored in the path memory for trunks (PMT) word. Using this JNNL, the sought LEN is derived from the PML word associated with it, and the JNN of the given trunk (JNNT) is obtained by translations. The path memory for a trunk-to-trunk (T-T) path is always displaced, and from it both JNNs and the sought TNN are determined and stored in the trace block.

3.83 Failure to find the given TNN on a queue, or list, or in a busy state implies that the PMT is not updated. Before a return is made to the client program indicating this, a transfer is made to SADT audit routine SADMPT which restores the trunk to its idle linked list and updates path memory.

TTY Input

3.84 Two entry points are provided for call trace to be requested via the TTY:

(1) TTYRL—for the directory number

(2) TTYRT—for the TNN.

The input and output messages involved are discussed in Section 231-110-301.

E. YAHA—Seize and Release Routines and L, J, and T-Bit Administration

3.85 This pident is a collection of routines that perform two general services for client programs:

(a) Seize a register from its idle linked list or release a register and restore it to its idle linked list. A listing of registers that YAHA seizes and releases for client programs and the associated global entry points are given in Table J. The types of registers processed include senior registers, junior registers, and auxiliary memory blocks. Senior registers control switching of the call and charging. The more common senior registers are originating, disconnect, ringing, and AMA. Junior registers are used to buffer I/O data for service circuits (eg, a junior register may contain digits to be outpulsed by a multifrequency transmitter). Common junior registers include transmission, reception, and coin control. Auxiliary memory blocks are used by senior call registers for storing path memory data for complex network connections. Common auxiliary memory blocks are 5-, 6-, 9-, and 10-word path memory annex (PMA) registers.

(b) Read and set the states of line, junctor, and trunk supervisory bits (L, J, T1, and T2 bits).

Administration of a Typical Idle Register Linked List

3.86 To reduce decisions and thus minimize execution time, separate routines are used to administer the different idle register linked lists. A typical one-way linked list of idle registers is shown in Fig. 1. The head cell contains the addresses of the most idle register (AA) and the least idle register (XX) on the list. The most idle register is determined by the first-on, first-off principle and the least idle, by the last-on, last-off principle. Each register on the list contains in its second word the address of the next most idle register. The last register contains all zeros in its second word.

Seizing and Idle Register

3.87 When YAHA is entered via a seize register global (Table J), the address of the most idle register on the idle linked list for the type of register requested is transferred from the associated head cell to a central control (CC) register. For example, the address of register AA (most idle) in Fig. 2 is transferred from the head cell to CC register Y (Fig. 2). The idle linked list record for AA is erased by writing all zeroes into the second word of register AA. To reset the idle linked list, the address of the next register (BB) to be used on a call is entered in the head cell. BB then becomes the new most idle register.

TABLE J

YAHA SEIZE AND RELEASE REGISTER ROUTINES

REGISTER TYPE	GLOBAL ENTRIES	
	SEIZE	RELEASE
AMA (9-word)	YASAM9	YARLA9
AMA (13-word)	YASZAM	YARLAM
AMA Special Services (18-word)	YASAMS	YARLSS
Conference	YASZCF	YARLCF
Coin Charge	YASCNC	YARCNC
Coin Control Junior	YASCNK	YARCNK
Dial Pulse Outpulsing Junior	YASDPJ	YARDPJ
Disconnect	YASZDR	YARLDR
Hotel-Motel	YASHMO	YARHMO
Incoming	YASZIR	YAROIR
Incoming Step by Step	YASISR	YARISR
Incoming Trunk Test	YASITT	YARITT
Line Ferrod Scan	YASLFD	YARLFD
Multifrequency Outpulsing Junior	YASMFJ	YARMFJ
Operator	YASZOP	YARLOP
Originating	YASZOR	YAROIR
Outpulsing	YASZOR	YAROIR
Outpulsing Annex	YASANX	YARANX
PCI Outpulsing Junior	YASPCJ	YARPCJ
PMA (5-word)	YASPM5	YARPM5
PMA (6-word)	YASPM6	YARPM6
PMA (9-word)	YASPM9	YARPM9
PMA (10-word)	YASP10	YARP10
PSPD	YASZPS	YARLPS
Regular Ringing	YASRR1	YARRR1
Reverting Call	YASRVC	YARRVC
Special Ringing	YASRR2	YARRR2
Scan Junior (6-word)	YASSC6	YARSC6
Temporary Transfer	YASTPT	YARTPT

Releasing a Busy Register

3.88 When YAHA is entered via a release register global (Table J), a register no longer needed on a call is placed in the least-idle position of the idle linked list for that particular register. For example, the address of register YY (new least idle) in Fig. 3 is recorded in the second word of register XX (old least idle). For this operation, the address of YY is held in a CC register, and the address of XX is held in the idle list head cell. Zeroes are stored in the second word of YY, and the address of the new least idle register (YY) is recorded in the head cell (Fig. 3). The idle list is now stable in its new state.

Administration of Line (L), Junctor (J), and Trunk (T1 and T2) Supervisory Bits

General Description

3.89 Each line in an office is assigned a line (L) bit in the line busy/idle bit table. The L bit keeps the busy/idle status of the line.

3.90 Similarly, each line junctor in an office is assigned two junctor (J) bits in the line junctor supervisory bit table, one for each of the two sides (side 0 and side 1) of the junctor. The J bits maintain the off-hook/on-hook status of their respective sides of the junctor.

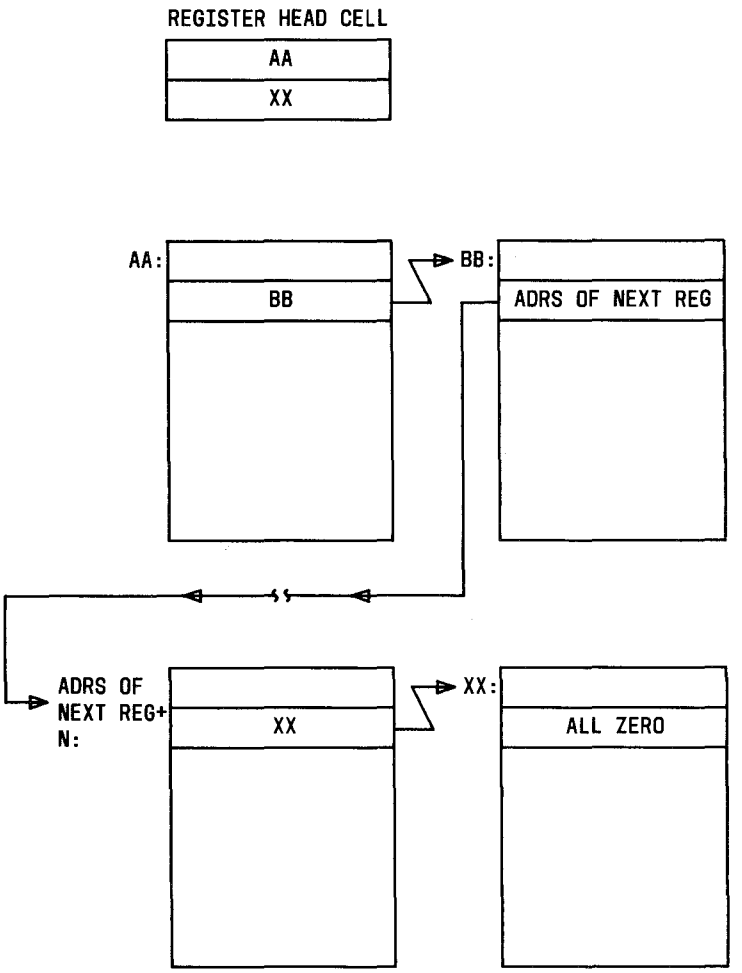


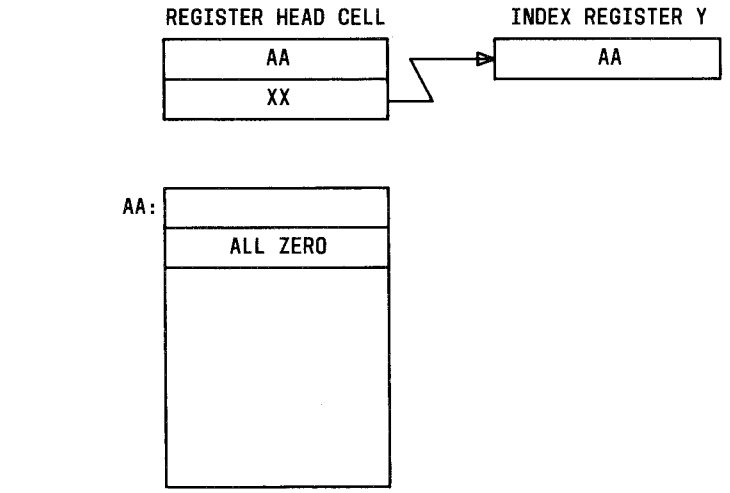
Fig. 1—Typical One-Way Linked List of Idle Registers

3.91 Each trunk in an office is assigned two T1 bits and two T2 bits in the T1 and T2 bit tables, respectively. Like line junctors, each trunk has two sides. Each T1 bit is used to maintain the off-hook/on-hook status of its respective side of the trunk. Every T1 bit has a corresponding T2 bit. T2 bits are set or reset by call processing programs to control reports of state changes of their corresponding T1 bits. If a call processing program wants a report when a T1 bit changes state (ie, the respective side of the trunk changes state from off-hook to on-hook or vice versa), that program sets the corresponding T2 bit to accept (1). If no T1 status change report is wanted (eg, during the switching of a line to a trunk), the corresponding T2 bit is set to ignore (0).

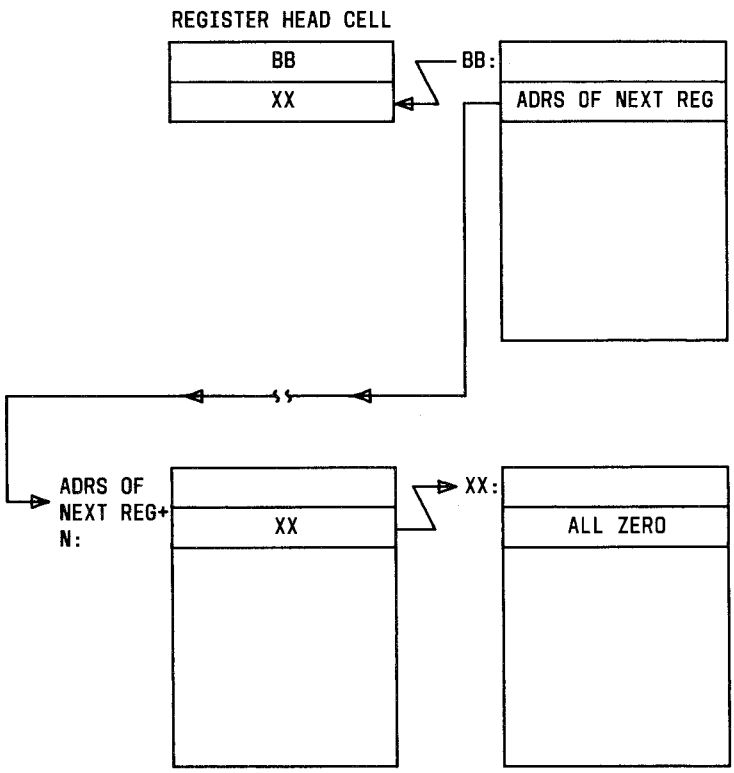
3.92 An I/O program checks the state of both sides (side 0 and side 1) of each trunk every 100 ms (more often for some trunks) and updates the corresponding T1 bits accordingly. If a T1 bit has been changed from 1 (on-hook) to 0 (off-hook) or vice versa, the I/O program making the check reports the change to a call processing program if the associated T2 bit is set to accept (1). The I/O programs do not set or reset T2 bits and line bits; only call processing programs write these bits.

YAHA Functional Interface Description

3.93 Pident YAHA contains subroutines for setting, writing, and reading the line, junctor, and T1 and T2 bits. If a call processing program wants

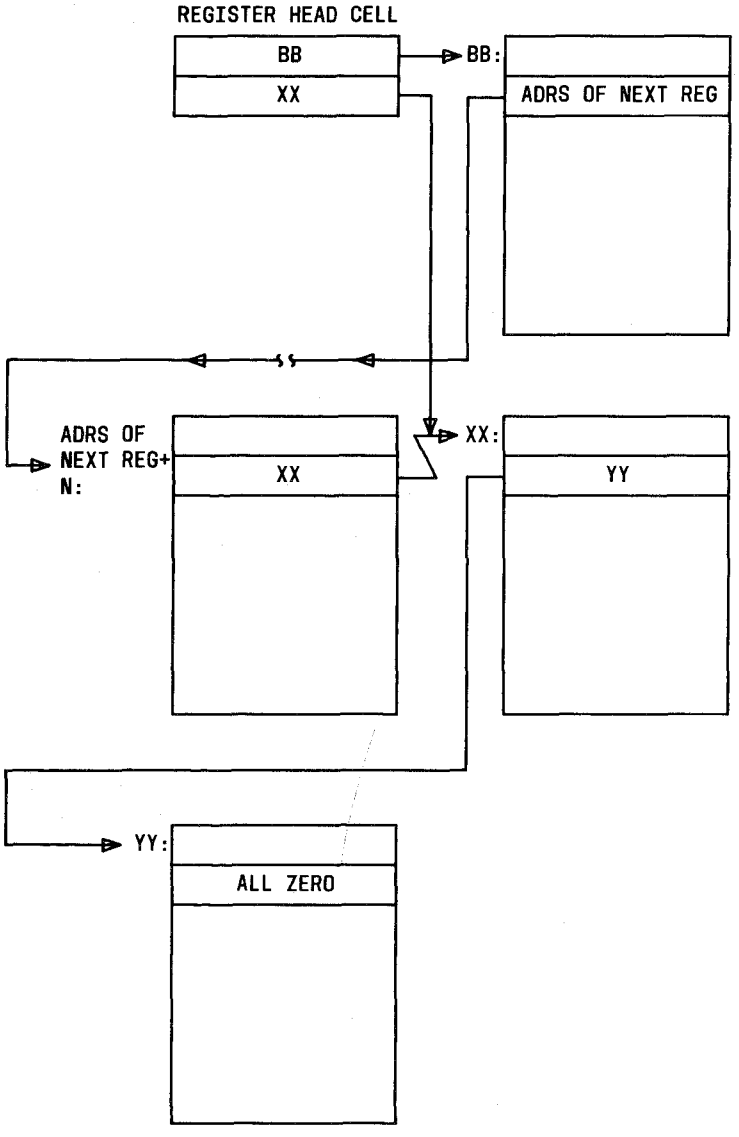


AA = MOST IDLE REGISTER
XX = LEAST IDLE REGISTER

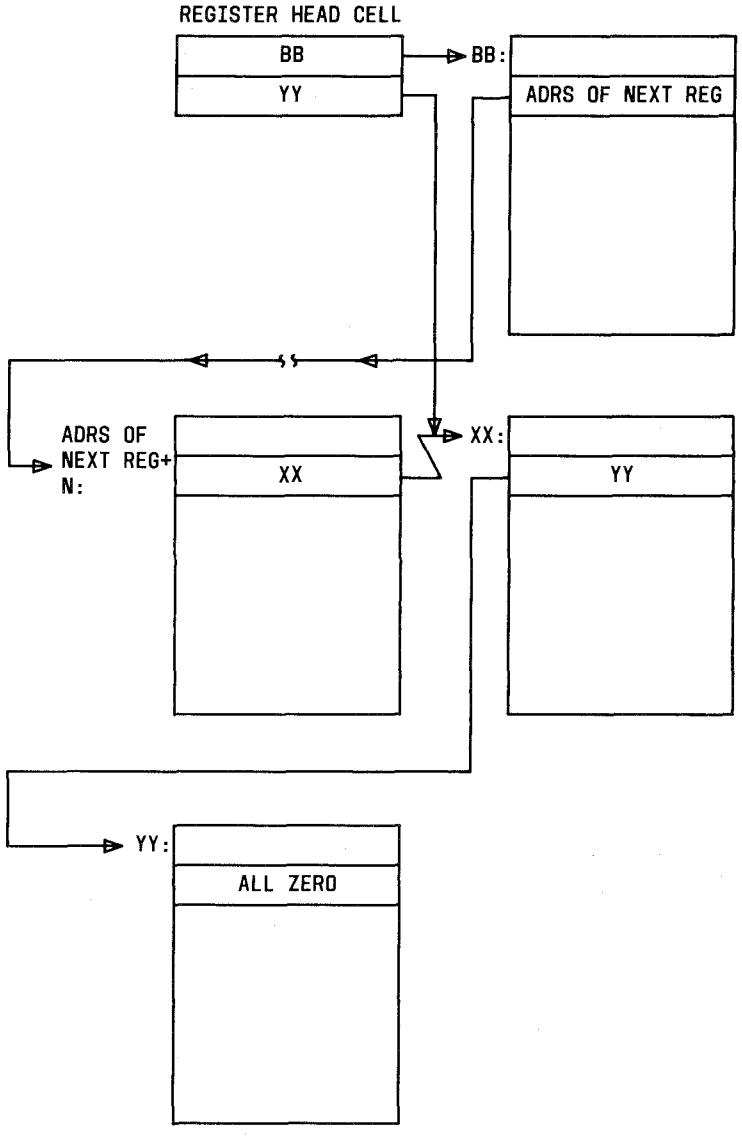


BB = MOST IDLE REGISTER
XX = LEAST IDLE REGISTER

Fig. 2—Seizing an Idle Register



BB = MOST IDLE REGISTER
XX = LEAST IDLE REGISTER (OLD)



BB = MOST IDLE REGISTER
XX = LEAST IDLE REGISTER (NEW)

Fig. 3—Releasing a Busy Register (Restoring to Idle Linked List)

to set a line bit to 0, indicating that the line is busy, the program transfers to the appropriate L-bit administration routine in YAHA. This routine determines which bit in the busy/idle bit table is assigned to that line [using the line equipment number (LEN) supplied by the call processing program], sets the bit to 0, and reports the previous state of the bit to the client program. At the conclusion of a call, a YAHA routine sets the line bit to 1.

3.94 When a call processing program wants to read the status of a T1 bit on side 0 of a given TNN, the program transfers to the appropriate T-bit administration routine in YAHA. This routine which reads side 0 T1 bits, determines which bit in the T1 bit table belongs to side 0 of the given TNN, reads it, and returns the results to the client program. Pident YAHA also contains separate routines for setting and resetting T1 and T2 bits of side 0 and side 1. Each routine requires a TNN as input. The T1 set and reset routines return the previous state of the T1 bit to the client program.

3.95 Pident YAHA includes separate routines for setting and resetting side 0 and side 1 J bits. The J-bit administration routines require a JNN as input and return the previous state of the J bit to the client.

F. YCLK—Register Linking Routine

3.96 This pident links a given register to a single call register or to a circular linked list of call registers on a single call. The address (in X) of the new register to be linked, and the address (in Y) of the register to which the new register is to be linked are provided by the client program.

3.97 Client programs enter YCLK via global YCLINK. If two single registers are to be linked together, the register address in X is stored in the link word of the register whose address is in Y. Then the register address in Y is stored in the link word of register whose address is in X.

3.98 If Y contains the address of a register to be linked to a group of linked registers and X contains the address of any one of the registers in the group, linking is accomplished as follows. The register address in Y is stored in the link word of the register which previously contained the address specified by X. The address in X is stored in the link word of the register specified by Y. This places the added register in a position preceding (ie, with link word pointing to) the register whose address is in X. Return is made to the client program.

G. YFDS—Scan of Single Master Scanner Program

3.99 Client programs enter pident YFDS via global YFMSDS to determine if a scan point on a master scanner is busy or idle. The master scanner number of the scan point to be scanned is provided by the client program. The scanner number is expanded to determine the enable address, the row, and the answer bit position of the scan point. Orders are sent on the peripheral bus to read the scan point. Upon completion of the scan, YFDS transfers control to the client return address if a busy condition (scan point = 0) is indicated. If an idle condition (scan point = 1) is indicated, control is transferred to the client return address incremented by 1.

H. YFTO—Incoming Trunk to Busy, Overflow, or Special Service Circuit Program

General

3.100 Client programs use the YFTO program to connect an incoming trunk to busy tone, regular overflow tone, common overflow tone, or a special service circuit such as high tone, low tone, announcement, or milliwatt test circuit. When entry is made to YFTO, a network path involving the incoming trunk may or may not exist. If a path does exist, the YFTO program will abandon and erase the path(s) involving the incoming trunk and then connect the trunk to a tone or special service circuit. Upon successful completion of the connection, supervisory scans for detecting the disconnect of the incoming trunk are initiated.

3.101 If the requested connection is to a busy tone trunk and a block or busy condition is encountered, the YFTO program attempts a connection to common overflow tone. If the requested connection is not to a busy or an overflow tone trunk, an attempt is made to connect to common overflow tone if a blocked condition is encountered. If a blocked or busy condition is found during an attempt to connect to common overflow tone or regular overflow tone, the incoming trunk is taken out of service until it disconnects.

Functional Description

Entry Requirements

3.102 Client programs enter YFTO at one of four entry points in an attempt to connect an incoming trunk to a tone trunk or to a special service circuit:

(1) YFBUSY—to busy tone

(2) YFOVER—to regular overflow tone

(3) YFOVFC—to common overflow tone

(4) YFSVRI—to special service circuit.

At the entries associated with the three tone trunks, YFTO supplies the corresponding route indexes (RI) of three trunks. The client program supplies the RI of the special service circuit for the fourth entry.

3.103 Before a client program enters the YFTO program, a call register and a POB must have been seized. The call register must be either a master register, linked or not linked, or a linked register with a master register on the link. The address of the first word of the POB which may be used for loading peripheral orders is also an entry requirement.

3.104 The incoming trunk must be in the hold, on-hook state, ie, the tip and ring toward the trunk are open circuited and on-hook supervision is being returned; the TNN of the incoming trunk must have been stored in a YFTO private call store word.

3.105 The path memory format index (PMFI) in the state word of the master register must contain 0, 1, 10, 9, or 31 [9 and 31 indicate that the incoming trunk is a foreign exchange (FX) trunk]. If the index contains 1, 9, 10, or 31, the path memory annex of the master register must then contain the corresponding path memory involving the incoming trunk. In addition, if the index contains 9 or 10, the line associated with the existing path must have been restored and verified before the YFTO program is entered. No path memory information is required when the index contains 0 or 31.

Completon of Connection

3.106 ***Successful:*** Before relinquishing control of a call to any other program, YFTO stores needed information in private call store words reserved for use by YFTO (Table K). At the entries associated with connections to busy tone, common overflow tone, or regular overflow tone, YFTO initializes R4TON0 to indicate that the circuit program index (CPI) of the tone trunk is known as a constant and R4TON2 to indicate that no signal answer is to be returned to the other office. The route index corresponding to the tone trunk is stored in R4TON1. At the special service circuit entry, if the route index provided is to be used for an incoming trunk test termination, R4TON0, R4TON1, and R4TON2 are set as described for a tone trunk entry. If the route index is not for an incoming trunk test termination, the route index of the circuit trunk is expanded; the "return supervision" item of word one of the route index expansion table specifies whether answer signal is to be returned to the outside office or not, and R4TON2 is set accordingly. Then R4TON0 is set to indicate that the CPI of the special service circuit is not known as a constant and will be derived from the TNN of the circuit trunk. The route index of the circuit trunk which was supplied by the client program is stored in R4TON1. For all entries, the POB loading address is stored in R4TON6 and the YFTO program address to be used at the completion of the tone report is stored in R4TON5.

3.107 A tone report is then initiated, notifying all call registers linked on the call that a tone connection is to be attempted. (The special service circuit trunk is treated as a tone trunk.) The client program associated with the master register uses the address in R4TON5 to return control to YFTO.

3.108 If the route index found in R4TON1 is to be used by the incoming trunk test program, YFTO then transfers control of the call to that program. If this is not the case, YFTO seizes a trunk-to-trunk memory (TTM) block and transfers control to a routine in the peripheral order loading programs for changes in network, the routine being dependent upon the contents of the PMFI item in the master call register. If the PMFI is a 1 or 10, the connection of the incoming trunk and another trunk is abandoned. In addition, if the PMFI is 10, the reserved path of the incoming trunk and a

TABLE K

PRIVATE CALL STORE (CS) FOR YFTO

SYMBOLIC ADDRESSES OF CS WORDS	INFORMATION STORED IN CS WORDS
R4TON0	Information indicating whether or not the CPI of the trunk to be connected to the incoming trunk is a constant
R4TON1	The route index of the trunk to be connected to the incoming trunk
R4TON2	Information indicating whether or not an answer signal is to be returned to the originating office upon successful completion of the desired connection
R4TON3	Information indicating whether or not the master register is to be left on the call and if the incoming trunk is off-hook
R4TON4	The trunk network number of the incoming trunk
R4TON5	The return address in YFTO to be used by a client program at the completion of a report of tone or end of call
R4TON6	The address of the first word of the POB which may be used for loading peripheral orders

line is erased and the connection of the line to a trunk is abandoned. The network actions which will connect the incoming trunk to the tone trunk or special service circuit are then loaded into the POB. The path memory corresponding to the connection is placed into the TTM block.

3.109 The YFTO program regains control of the call. If the PFMI in the master register is 10, the line bit of the line previously associated with the call is then idled. Since the path memory of the new trunk-to-trunk connection is not found in a call register, all call registers on the call are released. YFTO global YFTONT is used to load into the POB the relay and scan actions required to complete the connection. The TTM address is placed in the first word of the POB, and a failure option of one and a success address in the COPR program are loaded into the second word. The POB is then activated and control is returned to the main program.

3.110 Upon successful connection of the incoming trunk and a tone trunk or a special service circuit trunk, COPR receives control to initiate supervisory scans for the disconnect of the incoming trunk.

3.111 ***Unsuccessful:*** Several conditions may prevent the connection of the incoming trunk and requested tone or special service circuit:

(a) ***No TTM Block Available:*** If the YFTO program fails to seize a TTM block, the program address in R4TON5 is updated, and an end-of-call report is initiated. This results in all call registers on the call being notified that the call is to be abandoned. At the completion of the report, the client program associated with the master register returns control to YFTO via the program address in R4TON5. If the PMFI item in the master register contains a 0, all call registers associated with the call are released. YFTO routine YFHIWT then places the TNN of the incoming trunk on the trunk high and wet (THAW) list, sets up the supervisory scans for disconnect of the incoming trunk, and returns control to the main program. When the PMFI item in the master register does not contain a 0, YFTO then transfers control to global TYP1X in the YTTO program which takes type 1 failure actions on the call, releases all call registers associated with the call, and transfers control to the main program. The Type 1 failure actions

result in the TNN of the incoming trunk being placed on the THAW list until a disconnect occurs.

(b) ***Invalid Path Memory:*** After seizing a TTM block, if the PMFI item in the master call register does not contain a 0, 1, or 10, the TTM block is idled and control of the call is transferred to the subroutine in the YTTO program which takes type 1 failure actions on the call. The TNN of the incoming trunk is placed on the THAW list until disconnect occurs.

(c) ***Blocked or Busy:*** If a blocked or busy condition is encountered, the TTM block is idled. If a line was reserved to the incoming trunk in the previous path, the line bit of that line is also idled. YFTO global YFBLKD is then used to complete the required actions on the call.

(d) ***Hardware Failure:*** If a hardware failure is encountered during execution of the POB to connect the incoming trunk to a tone trunk or special service circuit, type I failure actions are taken on the call. The COPR program receives control and is responsible for any further actions on the call since the success address, placed into the POB before activation, was in that program.

(e) ***Special case of FX Trunk:*** Since an incoming FX trunk cannot be connected to tone, whenever a client requests that such a connection be made, the PMFI is set to 0, and the actions taken are those described in (a) above.

I. YMRG—Miscellaneous Register Subroutines and Tables

3.112 YMRG consists primarily of register hunt routines, register identifier-program tag (RI-PT) routines, and general release register routines. Also included in this program are the following tables associated with register processing:

(a) YMJRHC—Table of junior register head cells

(b) YMRIHC—Table of senior register head cells

(c) YMRILN—Table of senior register and junior register lengths

(d) YMRIPT—Table of program tag table addresses

(e) YMRIRL—Table of senior register release routines

(f) YMRISC—Table of junior register linkage audit scan word codes.

Register Hunt Routines

3.113 The register hunt routines refer to the parameter area data in program store to deliver to a client program, one after another on repeated entries, the addresses of all registers of a specified type. These routines are used primarily by audit programs in their checking or rebuilding of register lists but are also used by automatic message accounting and overload programs.

3.114 Client programs enter YMRG via global YMBLLH to begin a register hunt. The routine determines the address of the first register of the type specified by the client (central control register F contains the parameter area address of the primary word for that register type). The length of the register and the number of registers of that type are also calculated in this routine. Return is to the client continue return address (J + 1).

3.115 Subsequent entries to YMRG to continue the register hunt are made via global YMCLLH. All of these entries are made before a real time break; therefore, YMCLLH refers to information previously stored in scratch memory by YMBLLH to obtain the register length, the number of registers remaining in the hunt, the first-word address of the previous register, etc. Return is to the client continue return address (J + 1) until there are no more registers of the specified type to find, at which time transfer is made to the client return address (J).

RI-PT Report Routines

3.116 When the state of a call changes, a report is made to the call register (CR) assigned to the call at that particular phase of call processing. The type of information conveyed to the CR (via transfer to the appropriate RI-PT routine) indicates which of four possible events has occurred:

(1) A disconnect (off-hook to on-hook)

(2) Removal of a call register from a timing linked list upon time-out

(3) Removal of a register from a queue

(4) Any expected event except the three preceding, particularly answer (on-hook to off-hook).

3.117 The RI-PT routines are used to select the appropriate transition routine required to continue call processing. The transition routine selected is based on the type of report conveyed to the CR and on two items (RI and PT) stored in the state word of the CR. The RI item is a unique code assigned to each type of call register and is used to index into the Y4RIT table (Fig. 4). This table contains the addresses of all PT tables, one for each type of call register.

3.118 The PT item, which indicates the state of the call, is then used to index through the transfers in the PT table (Fig. 4) selected via Y4RIT. A PT table is subdivided into blocks of four transfer orders or transition routines. Basically, the four transition routines in each block are associated with the four types or categories of reports (disconnect, timing, standard, and queue) made to the CR. Treatment of the PT item, when indexing, differs with each type of report. Manipulation of the two least significant bits of the contents of the PT, as described below, breaks each PT table into blocks of four transition routines. The index placed in the PT selects one of these blocks, normally by pointing to the third of the four routines. The appropriate one of the four is selected for execution according to the category

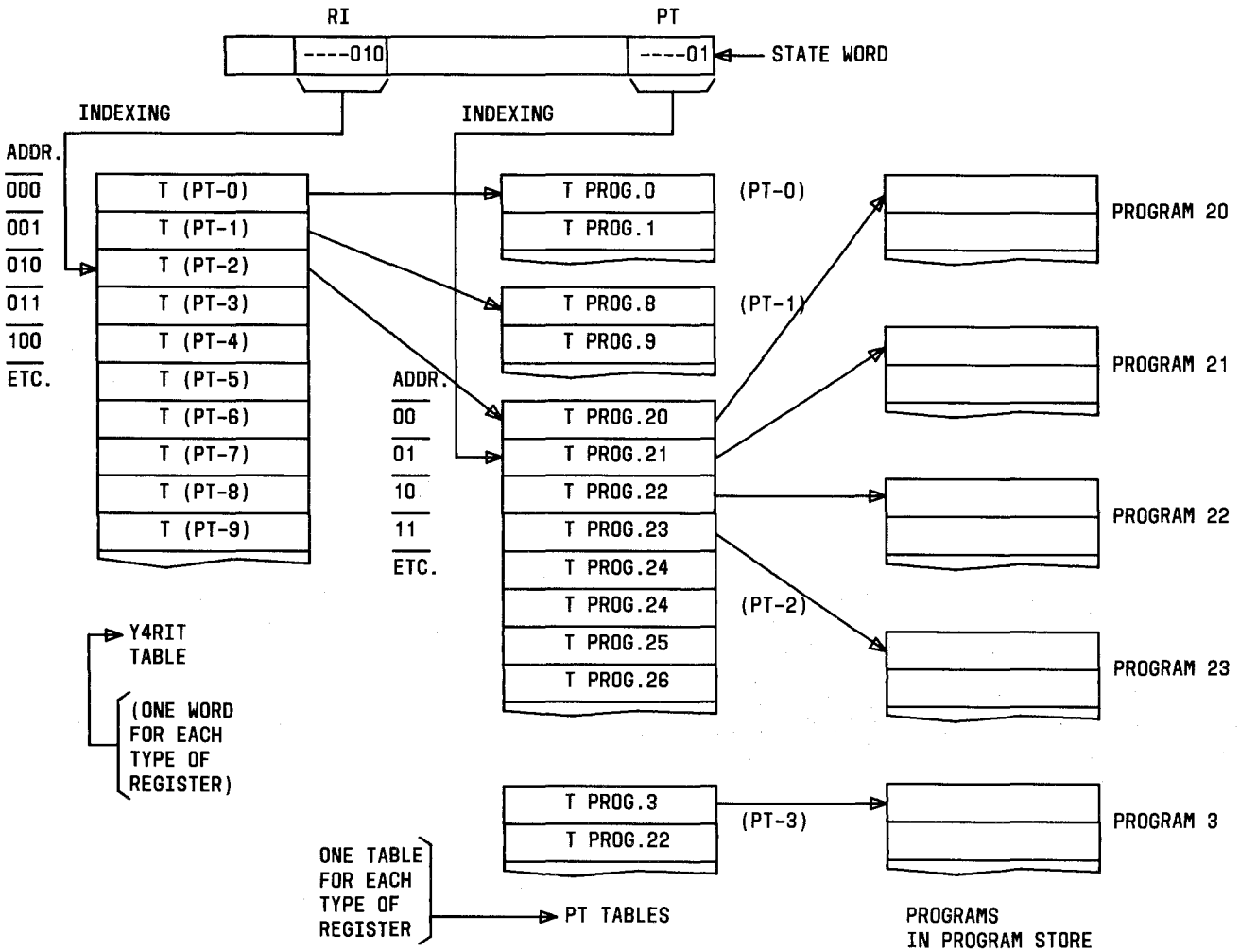


Fig. 4—RI-PT Method of Transferring to a Particular Call Processing Program

of the report by means of the routine associated with the category.

Disconnect Report

3.119 The disconnect routine (global YRIPTD) modifies the PT before indexing by setting the second low-order bit of the PT to 0, but leaving the value of the PT unchanged in the call register. This method is used to report a disconnect, ie, either party going from off-hook to on-hook.

Timing Report

3.120 The timing routine (global YRIPTT) sets the second low-order bit of the PT to 0 before using the PT as an index and then sets the low-order bit of the PT (the QI bit) to 0 in the call register. This method is used to report the removal of a call register from a timing linked list upon time-out.

Standard Report

3.121 The standard routine (global YRIPTS) uses the PT unmodified as the index and leaves the PT unchanged in the call register. This method is used to report answer (a change from on-hook to off-hook) and any other expected event not reported by the disconnect, timing, and queue methods.

Queue Report

3.122 The queue routine (global YRIPTQ) uses the PT unchanged as the index and sets the low order bit of the PT (QI bit) to 0 in the call register. This method is used to report the removal of a register from a queue.

Miscellaneous Scan Result Hopper Unloading

3.123 Every class B execution, ECMP enters YMRG via global DITSJH to unload the miscellaneous scan result hopper. This hopper which is loaded by DITS contains hit scan result one-word entries. As each entry is unloaded, control is transferred to the proper program via the RI-PT standard method (YRIPTS).

3.124 When a miscellaneous scan result hopper overflow occurs, AOVD enters YMRG via global YRDINT to unload the hopper in interject.

General Release Register Routines

3.125 The purpose of this group of routines is to release in a single register, or all registers in a circular linked list of called register on a single call, or all registers but one in a circular linked list.

3.126 To initiate the release of call registers, client programs access YMRG via the following global entries:

(a) YCRALL—Releases all linked call registers except the master register

(b) YCLKRG—Releases all linked call registers

(c) YCRONE—Releases a single call register (address in X)

(d) YCRONY—Releases a single call register (address in Y).

The routines, after reading the RI in the state word of the given register, transfer through the YMRIRL table indexed by RI to the appropriate routine to release the register. Register release routines, the majority of which are in YAHA, restore registers to the appropriate idle linked list and return to the client.

J. YTTO—Originating Line to Busy, Overflow, or Special Service Circuit Program

General

3.127 Pident YTTO is used by other programs to connect an originating line to busy tone, regular overflow tone, or a special service circuit trunk such as high tone, low tone, announcement, or milliwatt test circuit. Upon entry to YTTO, a network path involving the originating line may or may not exist. If a path does exist, the YTTO program will abandon and erase the path involving the originating line and then connect the line to a tone or a special service circuit. Upon successful completion of the connection, supervisory scans for detecting the disconnect of the originating line are initiated.

3.128 If the requested connection is not to an overflow tone trunk and a blocked or busy condition is encountered, the YTTO program attempts a second connection—this time, to common overflow

tone. If a blocked or busy condition is found when an attempt is made to connect to common or regular overflow tone, the call is abandoned and all memory associated with the call is idled. This will result in the originating line receiving dial tone if it remains off-hook.

Functional Description

Entry Requirements

3.129 Client programs enter YTTO at one of six entry points in an attempt to connect the originating line to a tone trunk or to a special service circuit:

(1) YTBSYS—to busy tone

(2) YTOVFR—to regular overflow tone

(3) YTOVFC—to common overflow tone

(4) YTOTHR—to special service circuit

(5) YTNCAR—to no-circuit announcement

(6) YTY218—to special service circuit.

Each entry results in an attempt to connect the originating line to a different trunk: busy tone, common overflow tone, regular overflow tone, a special service circuit whose CPI is from the TNN of the circuit trunk. The route index of the trunk to be connected to the originating line is used to identify the trunk.

3.130 Before a client program enters the YTTO program, a call register and a POB must have been seized. The call register must be a master register, linked or not linked, or a linked register with a master register on the circular linked list. The address of the first word of the POB which may be used for loading peripheral orders is also an entry requirement.

3.131 The PMFI in the state word of the master register must be 0, 7, 9, or 21. If the PMFI is 0, the path memory annex of the master register must contain the LEN and the control bits of the originating line. If the PMFI is 7, 9, or 21, the path memory annex of the master register must contain the corresponding path memory involving the originating line. In addition, if the PMFI is 21, the terminating line associated with line existing path must be restored and verified before the YTTO program is entered.

Completion of Connection

3.132 ***Successful:*** Before relinquishing control of a call to any other program, YTTO stores needed information in private call store words reserved for use by YTTO (Table L).

3.133 A tone report is then initiated, notifying all call registers linked on the call that a tone connection is to be attempted. The client program associated with the master register uses the address in R4TON5 to return control to YTTO.

3.134 Upon receiving control of the call, YTTO sets up information for the peripheral order loading programs for changes in network and transfers control to a routine in the latter program, the routine being dependent upon the PMFI in the master call register. If the PMFI in the master register is 7, 9, or 21, the connection of the originating line and a trunk is abandoned. In addition, if the PMFI is 21, the reserved path of the originating line and a terminating line is erased and the connection of the terminating line to another trunk is abandoned. Then, when the PMFI is 0, 7, or 21, the network actions which will connect the originating line to the tone trunk or special service circuit are loaded into the POB. When the PMFI is equal to 9, the network actions which will connect the existing reserved path of the originating line to a trunk are placed into the POB; by definition of the existing path, a new trunk is not introduced in this connection. If a terminating line was previously associated with the call, the line bit of the terminating line is idled after the network actions have been loaded into the POB.

3.135 If the master call register is not linked, PML and PMT words contain the new path memory and the master register is released. However, if the master register is linked, the new path memory is displaced and is found in the path memory annex of the linked register following the master register. The master register is then unlinked and released, the register containing the displaced new path memory is marked master register, and the PMFI item is updated to reflect the line-to-trunk path.

TABLE L

PRIVATE CALL STORE (CS) FOR YTTO

SYMBOLIC ADDRESSES OF CS WORDS	INFORMATION STORED IN CS WORDS
R4TON0	Information indicating whether or not the CPI of the trunk to be connected to the originating line is a constant
R4TON1	The route index of the trunk to be connected to the originating line
R4TON2	The address of the first word of the POB which may be used for loading peripheral orders
R4TON3	Information indicating options desired by the client program
R4TON5	The return address in YTTO to be used by a client program at the completion of a report of tone or end of call
R4TON6	A flag (NO_SUP_SCAN) used to inform program not to verify network continuity; also POB address stored here by YFTO

3.136 A subroutine of the YTTO program is used to load the relay and scan actions into the POB to complete the connection; the routine utilized depends upon whether the CPI of the trunk in the connection is known (global YTTONL) or must be translated from the TNN of the trunk (global YTTNCP). The TNN of the trunk is then placed into the first word of the POB, a failure option of type 1 and a success address in the COPR program are loaded into the second word, the POB is activated, and control is returned to the main program.

3.137 Upon successful connection of the originating party to a tone trunk, a special service circuit, or a previously reserved trunk, the COPR program receives control. COPR then starts supervisory scans for detecting the disconnect of the originating party, idles the POB, and returns control to the main program.

3.138 ***Unsuccessful:*** Several conditions may prevent the connection of the originating party and the requested trunk:

(a) ***Invalid Path Memory:*** After completion of the tone report and if the PMFI item in the master register contains an invalid value (not 0, 7 or 21), control of the call is transferred to a routine in YTTO (global YTTYP1 or YTYP1X) which takes type 1 failure actions on a call.

(b) ***Blocked or Busy:*** If a blocked or busy condition is encountered, 0 is placed into the PMFI item of the master register, indicating no path exists, and the line equipment number and the control bits of the originating line are placed into the path memory annex of the master register. If a terminating line was associated with the previous path, the line bit of that line is idled.

(1) If the connection which failed was not to common or regular overflow tone, YTTO transfers to its entry to connect the originating party to common overflow tone. The actions taken on this second attempt are those described for a connection to common overflow tone with 0 in the PMFI of the master register.

(2) If the connection which failed was to common overflow tone or to regular overflow tone, the program address in R4TON5 is updated, and an end of call report is initiated, notifying all call registers on the call that the call will be abandoned. At the completion of the report, the client program associated with the master call register returns control to YTTO via the address in R4TON5. The POB is then loaded with the network and relay actions to restore and verify the originating line.

(3) If the originating party is not a coin line, all call registers associated with the call are released, the POB is activated, and control is returned to the main program. Upon successful execution of the restore and verify actions, a routine in the COPR program receives control to idle the POB and the line bit of the originating line and to return control to the main program.

(4) If the originating party is a coin line, the POB is activated without releasing any call registers, and the main program receives control. Upon successful execution of the restore and verify actions, YTTO receives control. The POB is idled, and control is transferred to the coin charge program, which is responsible for releasing the call registers associated with the call and idling the line bit of the originating line.

(c) ***Hardware Failure:*** Type 1 failure actions are taken if a hardware failure is encountered during the execution of a POB which has been activated by YTTO. If the failure occurred during the execution of a POB containing the restore and verify actions of an originating coin line, YTTO releases all call registers and transfers control to the main program. A hardware failure occurring during the execution of any other POB activated by YTTO transfers control to COPR, which is then responsible for further actions on the call, since the success address in the POB upon activation is in the COPR program.

3.139 All actions explained for successful and unsuccessful completion of connection to tones and service circuits cover the cases when R4TON3 (Table L) contains zeroes. Normally client programs do not request options. For actions taken when R4TON3 is nonzero, refer to the YTTO program listing.

K. ZERO—Call Store Zeroing Program

3.140 This program is used to rapidly zero a block of adjacent call store words. Twenty global entries (ZERO01 through ZERO20) allow client programs to enter pident ZERO at the appropriate point for zeroing 20 or fewer words. For example, a client requiring that 16 words be zeroed would enter at global ZERO16. The address at which zeroing begins is provided by the client.

3.141 If more than 20 words are to be zeroed, client programs enter ZERO at global ZROCKF or global ZROMNY. A looping arrangement zeros adjacent call store words in blocks of 20 until the required total has been reached. Return is made to the client program.

4. ABBREVIATIONS AND ACRONYMS

ACD	Automatic Call Distribution
AIOD	Automatic Identified Outward Dialing
AMA	Automatic Message Accounting
AOVD	Automatic Overload Control Program
ASI	Alternate Server Intraflow/Interflow
ASP	Alternate Server Pool
CC	Central Control
CHGD	Scan Point Change Director Program
COPR	Report and Miscellaneous Subroutines
CPI	Circuit Program Index
CR	Call Register
CRA	Client Register Address
CS	Call Store
CXYH	Seize and Release Routines L, J, and T Bit Administration for Centrex
DISC	Disconnect Program
DITS	Timed Scan Junior Register Scan Program
ECMP	Executive Control Main Program
ESS	Electronic Switching System
FLQ	Fixed Length Queue

HC	Head Cell
ICT-L	Incoming Trunk-to-Line
JNN	Junctor Network Number
JNNL	Junctor Network Number on a Line Link Network
JNNT	Junctor Network Number on a Trunk Link Network
LEN	Line Equipment Number
LNT	Line Network Tag
L-OGT	Line-to-Outgoing Trunk
NEJR	Junctor Translations Program
NMDT	Line Bit Audit Programs
NTPI	Nontrunk Program Index
NTWK	Network Transition Control Program
OTQ1	Outgoing Trunk Queueing (Phase 1)
PIDENT	Program Identification
PMA	Path Memory Annex
PMFI	Path Memory Format Indicator
PMI	Path Memory Index
PML	Path Memory for Line
PMT	Path Memory for Trunks
POB	Peripheral Order Buffer
PSPD	Permanent Signal Partial Dial
QAPR	Queue Administration and Processing
QCDL	Queueing Data Link Automatic Call Distribution Interface
QCIA	Customer Interface and Special Auditing Routines
QEDA	Queue Entry and Destination Assignment Routines
QI	Queue Indicator
QSIF	Queue State Information Features
QTAL	Give Audible Disconnect, and Line Termination Routines
QTL	Queueing for Trunks and Lines
QTRK	Terminate to Trunk Facility Subroutines
QWAT	Queueing for WATS
RI	Route Index
RI-PT	Register Identifier-Program Tag
RW	Request Waiting
SPI	Special Program Index
SPML	Subtables for Path Memory for Lines
SPMT	Subtables for Path Memory for Trunks
THAW	Trunk High and Wet List
TNN	Trunk Network Number
TPI	Trunk Program Index
TRCE	Call Trace
TRTB	Translation Routines-Basic Trunk
TSAH	Trunk Seizure and Answer Hopper
TSJR	Timed Scan Junior Register
TSN	Trunk Scanner Number
VLQ	Variable Length Queue
WATS	Wide Area Telephone Service
WQUE	Queue Administration Program

YAHA	Seize and Release Routines and L, J, and T Bit Administration
YCLK	Register Linking Routine
YFDS	Scan of Single Master Scanner Point
YFTO	Incoming Trunk to Busy, Overflow, or Special Service Circuit
YMRG	Miscellaneous Register Subroutines and Tables
YTTO	Originating Line to Busy, Overflow, or Special Service Circuit
ZERO	Call Store Zeroing.

5. REFERENCES

A. OM-6A001—Output Message Manual

B. Section 231-045-100—Operational Software Control Structure

C. Section 231-045-215—Audits

D. Section 231-090-167—Queueing for Trunks and Lines

E. Section 231-090-339—ACD Queueing and Call Distribution to Agents Feature

F. Section 231-090-408—Outgoing Trunk Queueing Feature (Phase 1)

G. Section 231-110-301—Call Tracing and Calling Line Identification Procedures

H. Queue and General Purpose Program Listings (See Table A).