PERIPHERAL UNIT CONTROLLER/DATA LINK SOFTWARE DESCRIPTION NO. 1 AND NO. 1A ELECTRONIC SWITCHING SYSTEMS

CONTENTS PAGE

Figures

CONTENTS PAGE

1. GENERAL

1.01 This section describes the functional operation of the Peripheral Unit Controller/Data Link (PUC/DL) software for the No. 1 and No. 1A Electronic Switching Systems (ESSs).

Note: This section applies to the 1E7 (No. 1 ESS) and 1AE7 (No. 1A ESS) versions of the generic program.

1.02 Whenever this section is reissued, the reason for reissue will be stated in this paragraph.

1.03 The PUC/DL program identifications (pidents) provide the software for processing the information received from and sent to the remote terminals via data links. These pidents are unique to the operation of the remote data link facilities, Remote Switching System (RSS), and electronic tandem switching (ETS).

1.04 The PUC pidents that perform functions common to the remote data link facilities and the digital carrier trunk facilities are described in Section 231-045-430.

Note: For common channel interface signaling (CCIS), PUC/DL pidents will recognize CCIS information, and this information is passed to the CCIS link security pidents.

1.05 Part 3 describes the functional operation of the PUC/DL pidents listed below. The program listings (PRs) may be referred to for additional information.

- PUDR—PUC/DL Fault Recovery (PR-1A455)
- PUID—PUC/DL Initialization (PR-1A461)
- PUTY—PUC/DL TTY Interface (PR-1A448)
- PUDT—PUC/DL Tables (PR-1A455).

1.06 This section includes structural flow diagrams for some of the pident routines covered in Part 3. These routines allow program control to be returned to the calling routine when the called subroutine task has been completed.

1.07 Part 4 contains the abbreviations and acronyms used in this section.

2. PUC/DL FUNCTIONAL INTERFACE AND MAINTENANCE

HARDWARE INTERFACE

A. PUC/DL Frame

2.01 The PUC is located in a 1-bay frame (Fig. 1) for the data link application.

B. PUC Block Diagram

2.02 The PUC (Fig. 2) contains two identical self-checking microprocessor controllers. Each controller consists of a central processing unit, a memory, an ESS-PUC interface, and a PUC peripheral interface which connects the data links to the remote data link terminals. The controllers execute orders received from central control. In normal operation, both controllers operate in a matched mode state.

2.03 The PUC microprocessors use firmware to control a particular PUC/DL application. The PUC/DL application includes the RSS and ETS.

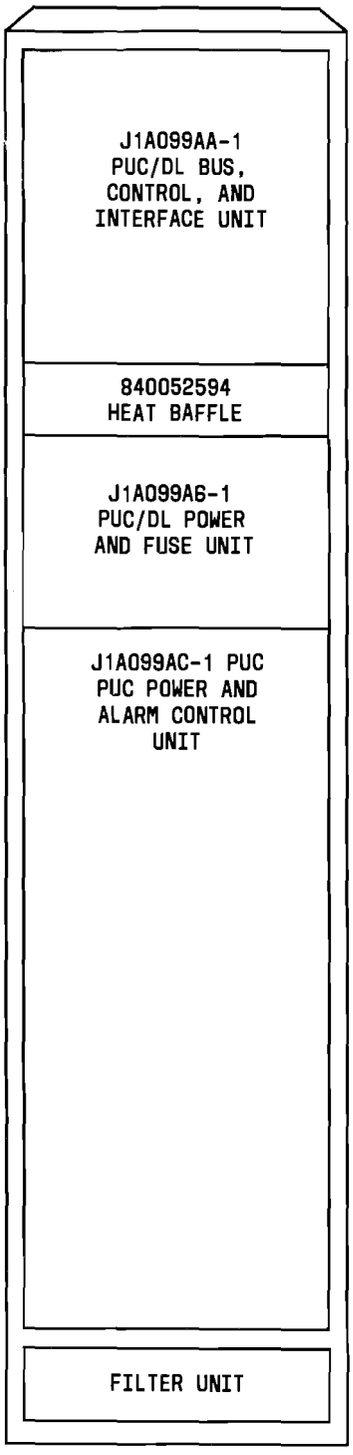


Fig. 1—PUC/DL Frame

RSS Application

2.04 The PUC facility performs the following actions for the PUC/DL RSS (Fig. 3) application:

(a) Receives data from the host ESS, buffers and formats the data into the appropriate data link message protocol, and then transmits the data serially on the data link

(b) Responds to data link reconfiguration request from host central control

(c) Receives serial data from the RSS, buffers it, and signals the host central control so that the information received can be read by the host system

(d) Detects its own PUC faults and reports to the host central control

(e) Performs diagnostic tests when requested by the host central control or on a timed basis

(f) Performs audits of its own memory when requested by the host

(g) Reinitializes itself when requested by the host central control.

ETS Application

2.05 The PUC/DL ETS (Fig. 4) application provides the telephone customer limited access to stored data. The data link interface to the ESS is provided by the Customer Administration Center System (CACS) or a Local Customer Administration System Terminal (LCAT).

C. PUC/DL Operation

2.06 For all remote data link applications, the PUC operates autonomously using its own timing control mechanism. It receives information from the ESS and executes the received orders at a designated time determined by its timing cycle. Data is returned to the ESS via normal scanning orders.

2.07 A block diagram of the PUC/DL processing units is shown in Fig. 5.

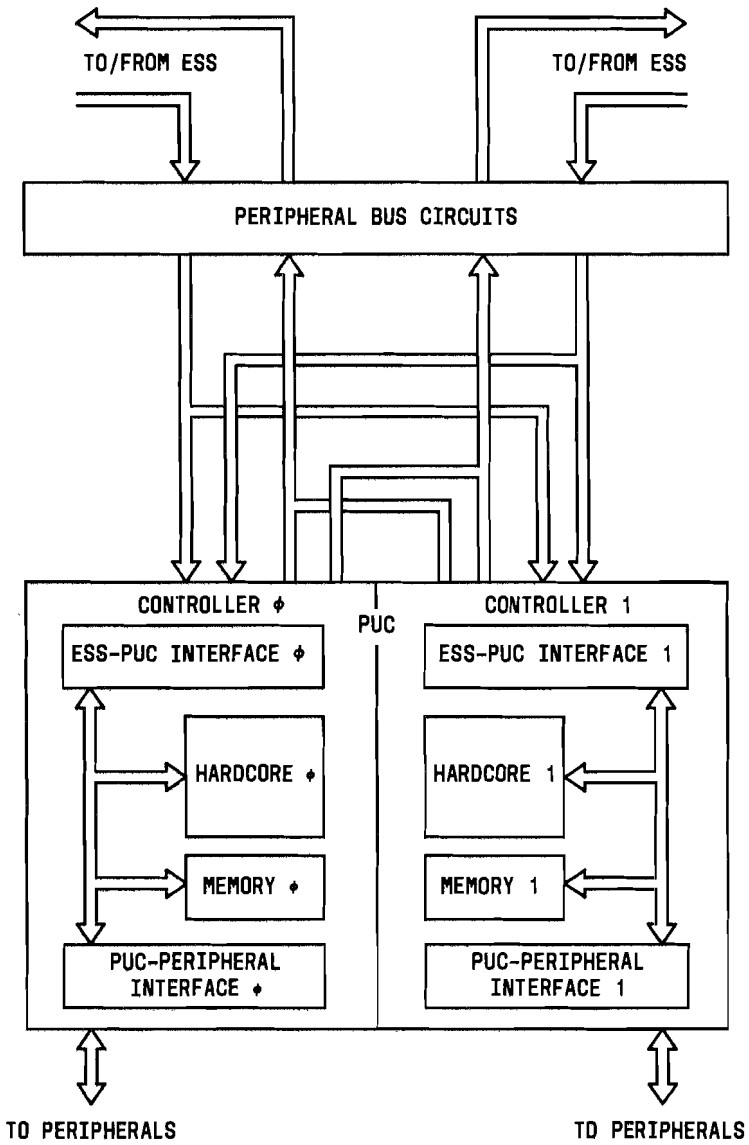


Fig. 2—PUC Block Diagram

SOFTWARE INTERFACE

A. PUC/DL Input Data Flow

2.08 The incoming data to the PUC/DL from central control is transmitted over the peripheral unit address bus (PUAB) to the first-in/first-out (FIFO) buffer. This buffer contains 256 words with each word containing 16 bits of data.

2.09 The transferring of information between the PUC/DL and the ESS is accomplished by the PUC/DL application firmware.

2.10 The input data messages are loaded into buffers in the PUC scanner answer memory (SCAM). A total of four out of eight SCAMs having 64 words each are provided for storing the input data. (See Fig. 6.)

2.11 The PUC/DL firmware separates each input word into three data fields (high order control plus two data fields). The high order control field is checked for parity. If the parity is incorrect, an error report is returned to the ESS. The control field also contains indicators for priority, application, maintenance, and the message type that identifies the order.

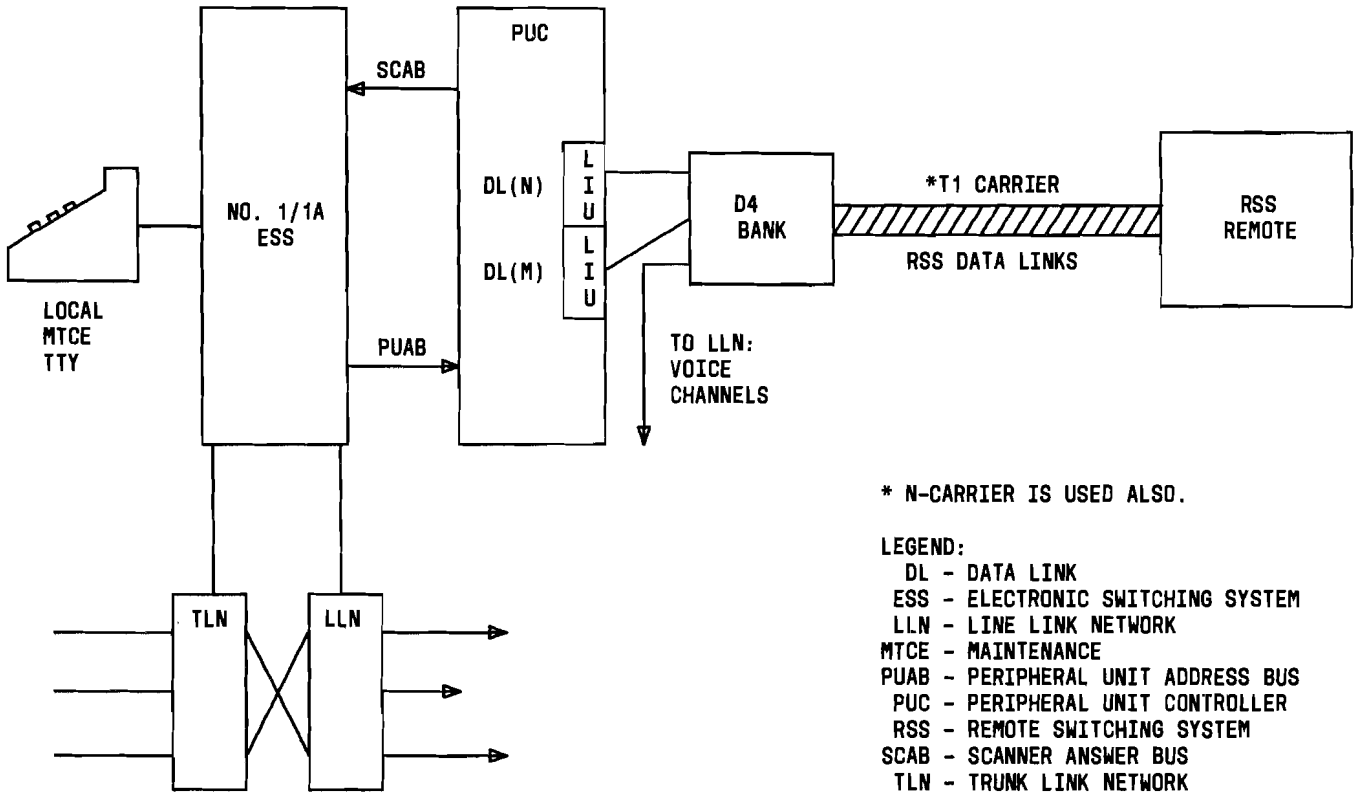


Fig. 3—PUC/DL RSS T1 Carrier Office Configuration

2.12 The incoming message words from the ESS are separated into three categories:

- Data
- Parameter
- Maintenance.

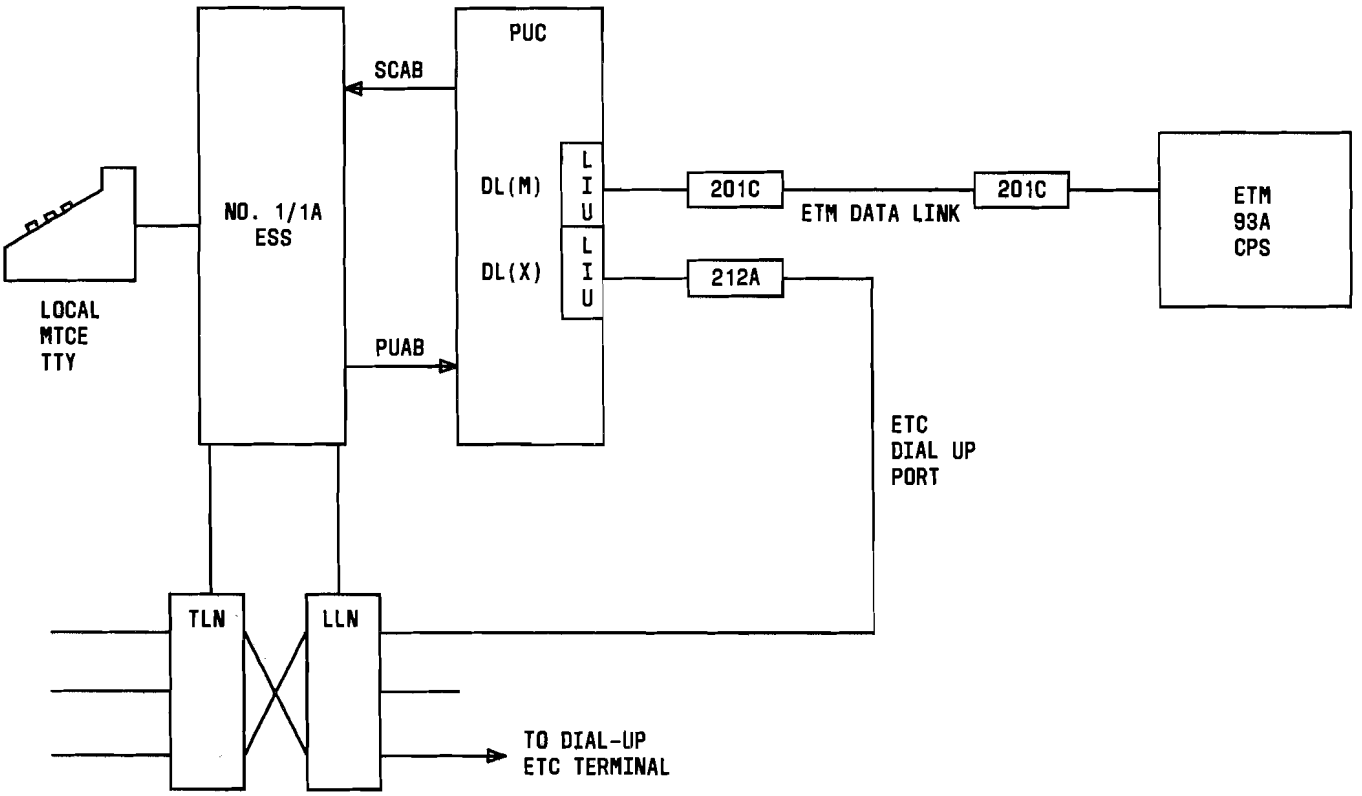
Data Messages

2.13 The data message (Fig. 7) is marked with a destination buffer number specifying a buffer in the PUC/DL read/write memory. The data messages begin with a heading word. The middle field of the heading word contains the destination number. This specifies the transmit buffer where the words which follow the heading word will be stored. The heading word, after being interpreted by the PUC/DL, is discarded. The low-order message field of the heading word contains a word count indicating the number of words to be written in the FIFO buffer. The data words following the heading word are identified by all data bits zero in the high order field.

2.14 When each new data word is stored in the destination buffer (Fig. 8), it is added to the end of the list. If the buffer is full, the present message is discarded and an error is reported. Cumulative counts are kept of all words unloaded from the FIFO buffer and the destination buffer. These counts are recorded in the SCAM where they are read and used by the ESS to control the input rates and prevent overflow during normal operation.

Parameter Messages

2.15 The parameter messages are used to define the operation of the PUC/DL. Most parameters are sent when the unit is brought on-line. The receipt of a parameter causes a table entry to be initialized or changed and an audit to be generated. Its purpose is to change the internal data structures to agree with the new parameter.



LEGEND:

CPS - CUSTOMER PREMISES SYSTEM
DL - DATA LINK
ESS - ELECTRONIC SWITCHING SYSTEM
ETC - ETS FOR CUSTOMER ADMINISTRATION CENTER SYSTEM
ETM - ETS FOR MESSAGE DETAIL RECORDING
LIU - LINE INTERFACE UNIT
LLN - LINE LINK NETWORK
MTCE - MAINTENANCE
PUAB - PERIPHERAL UNIT ADDRESS BUS
SCAB - SCANNER ANSWER BUS
TLN - TRUNK LINK NETWORK

Fig. 4—PUC/DL ETS Office Configuration

2.16 The parameter messages are also used to specify the number and length of destination buffers; the number, length, and location of data output buffers in the SCAM; and the destination and protocol type of each data link.

Maintenance Messages

2.17 The maintenance messages are processed by the controller maintenance programs. They are used for bringing a data link on-line, clearing an error count, and switching data links.

B. PUC/DL Output Data Flow

2.18 All output data and control information from the PUC/DL to the ESS passes via the PUC SCAM. The data is divided into three types:

- Incoming data link messages
- Error and status reports
- Special indicators.

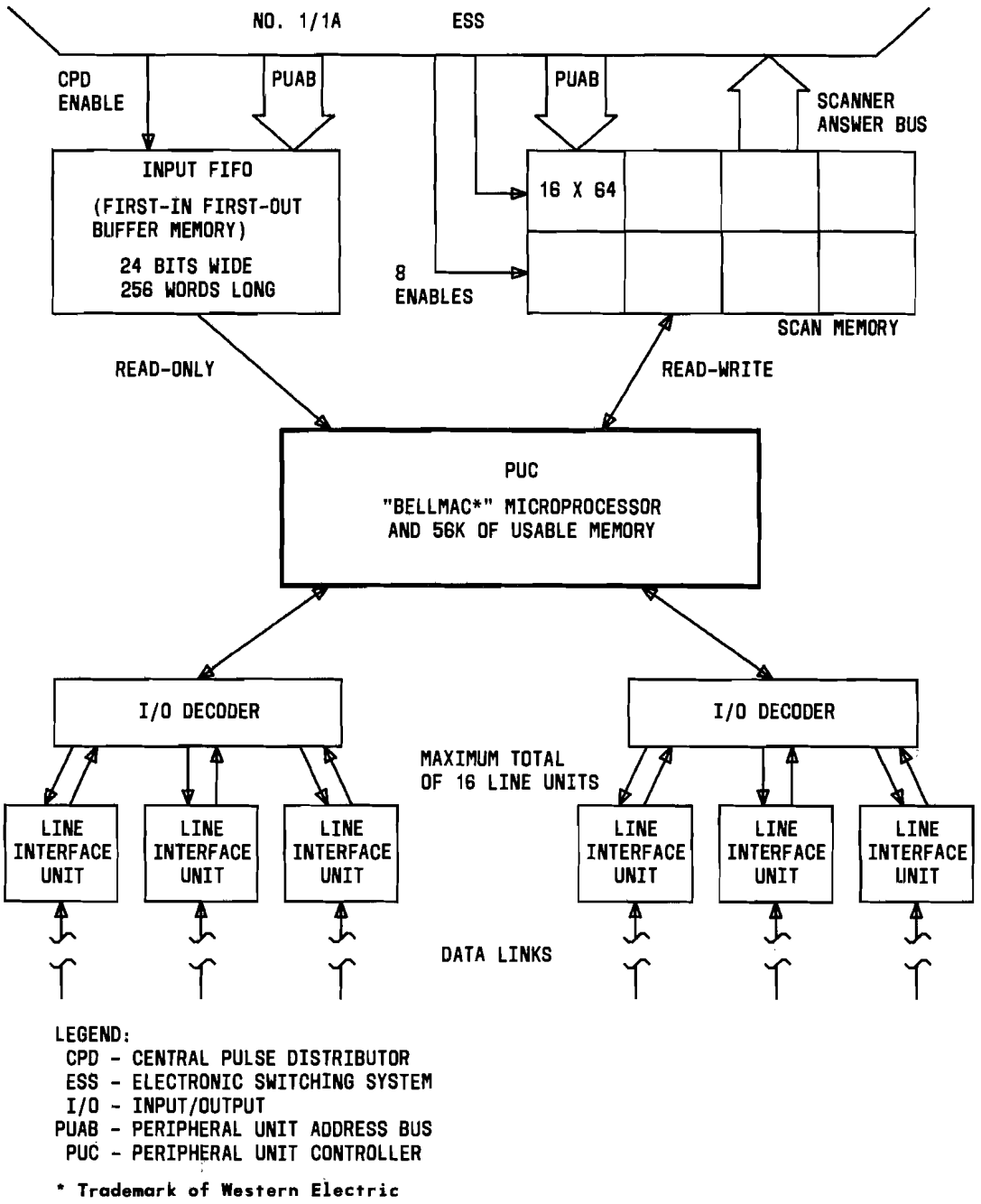


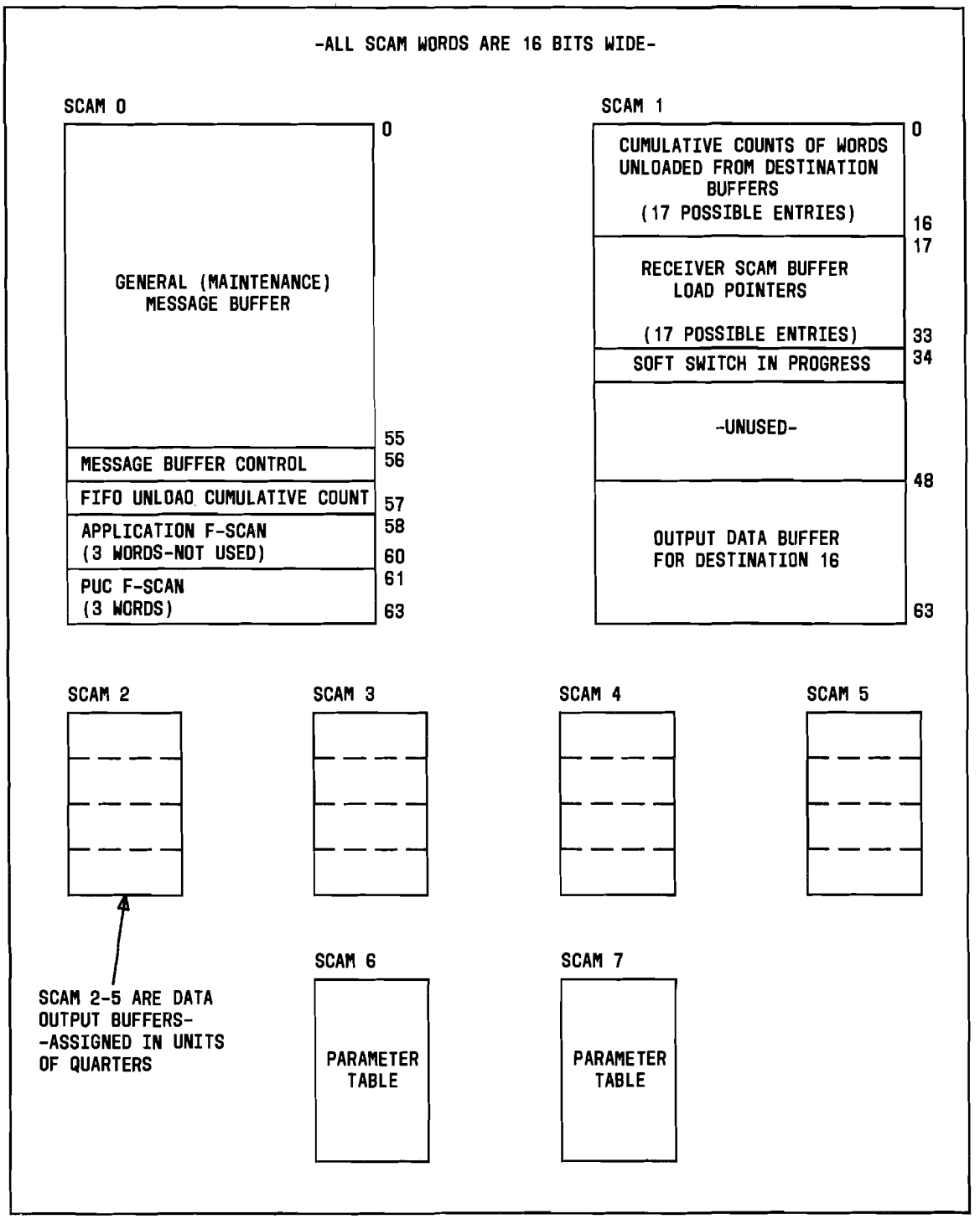
Fig. 5—PUC/DL Processing Units

Incoming Data From Data Links

2.19 The incoming data from the data links is deposited in the circular buffers located in the SCAM. The SCAM load pointer is used for loading the data in the SCAM. The unload pointer used for unloading the buffers is located within the ESS.

Error and Status Reports

2.20 The maintenance messages contain the error and status reports. These messages are stored in the general reply buffer which is provided with a load and an unload pointer. A maintenance message



LEGEND:
FIFO - FIRST IN-FIRST OUT
SCAM - SCANNER ANSWER MEMORY

Fig. 6—PUC SCAM Layout

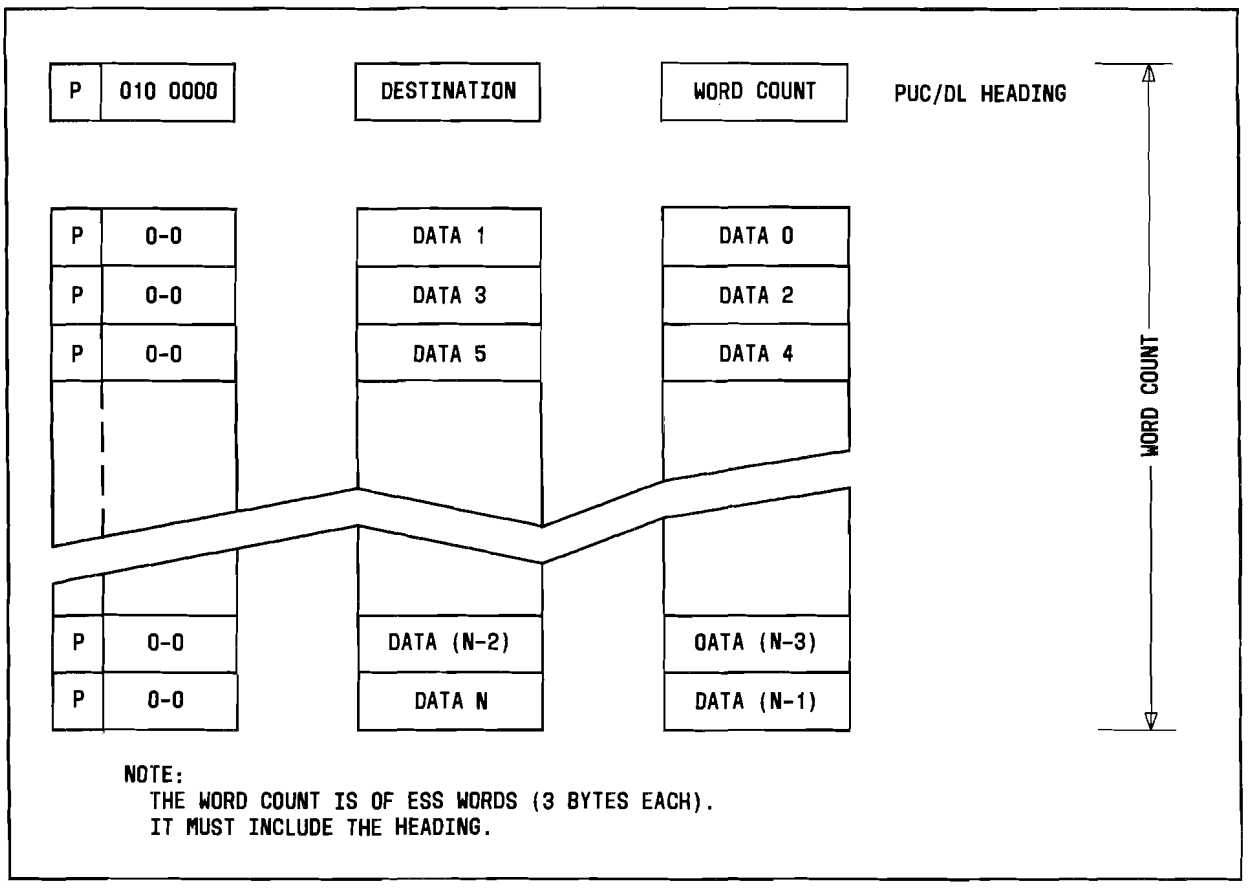


Fig. 7—Input Data Message

is sent only when there is available space in the buffer.

Special Indicators

2.21 The special indicator type includes unload counters and pointers. An indicator is also provided to indicate when a data link switch is in progress.

C. Parameters

2.22 Parameters are single-word application maintenance messages which contain information used for updating and activating data links. They contain one byte of parameter information and two bytes of information that distinguish the parameter message from other types. The four categories of parameters are:

- General
- Destination
- Data link basic
- Data link protocol.

General Parameters

2.23 The general parameters are applicable to the operation of the entire PUC/DL facility.

Destination Parameters

2.24 The destination parameters contain information relating to the input and output buffers. Each set (16 maximum) defines the destination buffer length, scratch block length, and the SCAM buffer length.

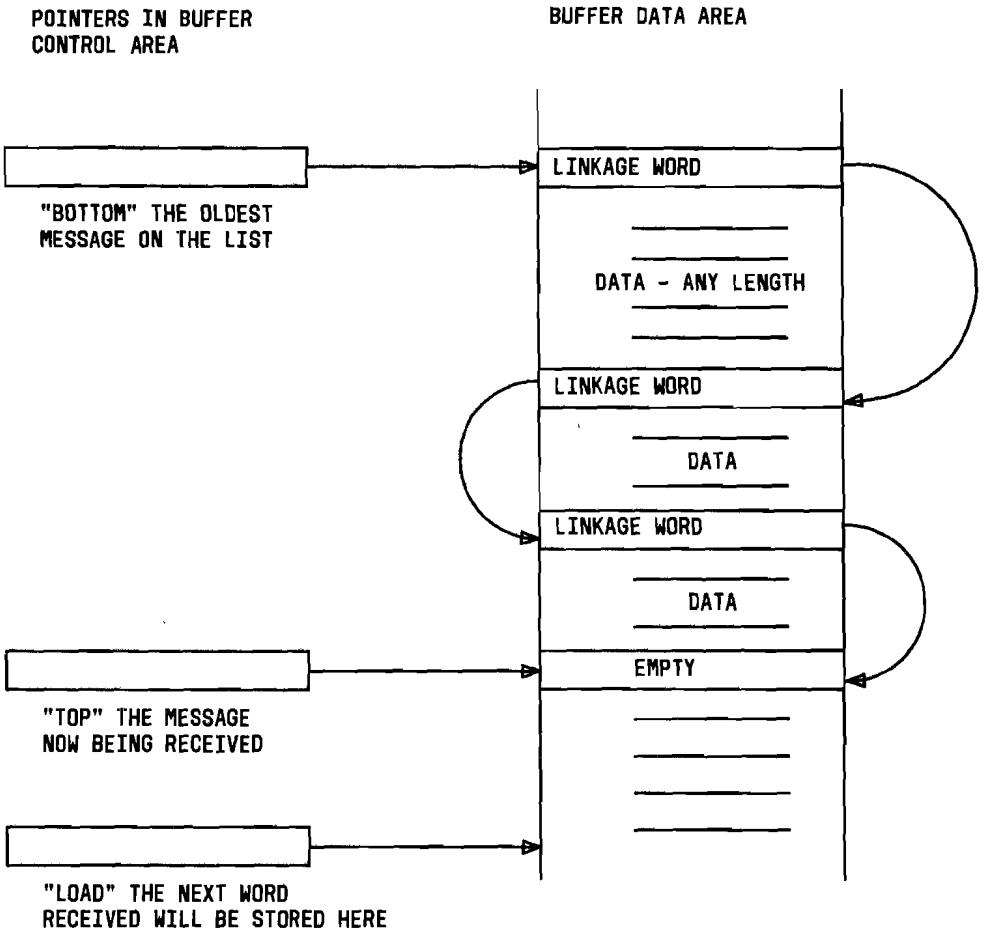


Fig. 8—PUC/DL Destination Buffer Structure

Data Link Basic Parameters

2.25 The data link basic parameters are used to specify the information applicable to individual data links. The two categories of data link basic parameters are protocol number and destination number.

Data Link Protocol Parameters

2.26 The data link protocol parameters are associated with the protocol routines and are applicable only to the protocol that is assigned to the data link.

MAINTENANCE

A. PUC/DL Faults

2.27 The PUC/DL faults are reported to maintenance personnel via the input/output terminals. Both the maintenance and the diagnostics on the PUC/DL facility are controlled by the ESS software pidents described in Part 3.

B. Data Link Recovery

2.28 The operational capability of the data links is checked by the PUC. When a trouble (carrier loss, protocol response failure from the remote end, excessive error rates) is detected, the PUC sends the report to the ESS. The recovery from a data link fault is handled automatically by the recovery pident PUDR. The recovery action involves establishing a working configuration with the data link, diagnosing the faulty link, and reporting the fault to the maintenance personnel via an output message.

C. Diagnostics

2.29 The diagnostic function includes the testing of various sections of the data link by looping

signals at different interfaces. The returned signal is compared to the transmitted signal with mismatches indicating a failure.

D. Maintenance Orders

2.30 The PUC maintenance orders are processed by the controller maintenance programs. They are normally used for bringing a data link on-line, clearing an error count, or switching between redundant data links to the same destination.

E. Audits

2.31 Audits are performed to build and initialize data structures in the read/write memory. The parameter tables located in SCAM 6 and 7 provide the input data for the audit routines. The audits provide space for the destination buffers and scratch blocks when these areas are requested by the parameter inputs. Whenever parameters are changed, the entire sequence of audits are performed.

3. PUC/DL PIDENT FUNCTIONAL DESCRIPTION

PUC/DL FAULT RECOVERY—PUDR

A. General

3.01 The PUDR pident performs actions such as switching data links, sending orders to the PUC, scheduling diagnostics on the data links, and updating the call store that is associated with the data links in an effort to effect system recovery from a faulty data link.

B. Functional Routines

3.02 The external routines that interface with the PUDR pident are described in the following paragraphs.

Initialization Request (PUDLII)

3.03 Subroutine PUDLII is an entry point for the PROCESS_INIT_REQUEST subroutine. Whenever a link is initialized by pident PUID, PUDLII is entered. This routine validates the initialization request. It also checks the application status block associated with the link to see if another reconfiguration job is in progress. If both checks pass, the application and application member number of the data link are stored in the status block. Subroutine PUDLRC is then called to perform the initialization function.

Diagnostic Request (PUDDGI)

3.04 Diagnostic requests initiated either by pident PUDD (PUC data link diagnostic) or SAPU (PUC data link audit) causes entry into PUDR at subroutine PUDDGI. This entry point serves as the interface for subroutine PROCESS_DIAG_REQUEST. This subroutine checks the application status block of the data link to see if another reconfiguration job is in progress. If so, the request is denied; if not, the request is passed to PUDLRC.

PUC Report Handler (PUDLPF)

3.05 Subroutine PUDLPF serves as the interface entry for processing the data link report. The report that is received from the PUC is actually processed by subroutine PROCESS_PUC_REPORT. Validity checks are performed on the report before it is allowed to pass to the reconfiguration routines. If the report type is invalid or the report-ignore flag is set, the report is discarded. If the report is a leaky-bucket-overflow type, the threshold is checked against the actual threshold data link current state. The report is also discarded if thresholds do not match. If the validation checks pass, the stimulus is stored and the main program flag 384 is set (see below).

PUC/DL Scheduler (PUDLFR)

3.06 Whenever the main program flag 384 is set indicating there is reconfiguration work, an entry is made at subroutine PUDLFR. This subroutine scans the application status blocks of the data link in search for work. When the link needing work is found, a transfer is made to PUDLRC.

PUC/DL Timer (PUDLTM)

3.07 Whenever a time-out condition occurs on a data link indicated by the main program flag 430, PUDR is entered at subroutine PUDLTM. This subroutine searches the links for time-out. The associated time-out counter is decremented. If the counter reaches a zero value, the main program flag 430 is turned off and flag 384 is set.

Hourly Entry (PUDLHR)

3.08 Subroutine PUDLHR is the interface entry point for the HOURLY_REX subroutine. This

latter routine clears the trouble analysis counters and schedules a diagnostic on all out-of-service designate and out-of-service remote links. If the designate link is active and there is no reconfiguration work in progress, the performance parameters are restored to their original values.

Midnight Entry (PUDMID)

3.09 Subroutine PUDMID is the interface entry point for the MIDNIGHT_REX subroutine. For every PUC/DL, the MIDNIGHT_REX subroutine performs the following actions:

(1) Clears the trouble analysis counters

(2) Reestablish nominal performance parameters

(3) Schedules a diagnostic on certain links.

Data Link Reconfiguration (PUDLRC)

3.10 Subroutine PUDLRC is entered from subroutine PUDLII. It manages the entire data link reconfiguration process that is performed on a data link in response to an input stimulus. It calls various subroutines to perform the requested recovery action. These subroutines are presented in a modularly structured arrangement (Fig. 9) and are further described in the following paragraphs.

Initialization of T-Scratch Area (INITIALIZE)

3.11 The first called procedure by PUDLRC is subroutine INITIALIZE. This subroutine checks the validity of the data link application and initializes the T-scratch area in call store that is associated with the data link.

Initialization of Mate Link (GET_MATE_FOR_INIT)

3.12 If the data link application validity check passes, subroutine GET_MATE_FOR_INIT is called to derive the member number and associated state block of the mate link for the duplex link application.

Initialization of Status Block (PUDLAI)

3.13 The PUDLRC calls subroutine PUDLAI in pident PUID to search the PUC hopper parameter table in order to obtain the application and member number of the data link. The PUDLAI calls either PSEUDO_INIT_RSS or PSEUDO_INIT_ETS (application dependent) to initialize the associated pseudo status block.

Delayed Work Schedule (DELAYED_WORK_FOR_INIT)

3.14 After a link has been initialized, subroutine DELAYED_WORK_FOR_INIT is called by PUDLRC to determine which deferred jobs should be scheduled. If the initialization action was blocked because either there was not enough space in the FIFO or the FIFO buffer is locked out, this subroutine increments the FIFO counter that is located in the application status block. However, if the initialization is not blocked, the initialization action is executed.

Duplex Link Reconfiguration (DUPLEX_LINK_RECONFIG)

3.15 Whenever a reconfiguration of duplex links is to be performed, PUDLRC calls DUPLEX_LINK_RECONFIG. This subroutine calls the following subroutines to derive the reconfiguration states and actions:

(a) GET_MATE—Initializes the mate link scratch block

(b) TBL_ANAL—Modifies stimulus if state oscillates

(c) FIND_DUPLEX_STATE—Determines next states and actions via an access table

(d) HANDLE_DUPLEX_FAILURE—Attempts to bring the failed link on-line.

3.16 Subroutine GET_MATE is called to derive the member number of the mate link. In the case of an RSS data link application, subroutine GET_MATE_RSS is called to find the member numbers of both links via the RSS common block translator.

3.17 If the input stimulus pertains to the PUC failure reports, subroutine TBL_ANAL is called to examine the trouble analysis counters in the application status block of the designate link. Depending upon the state of the link, the stimulus is changed for a trouble analysis if the appropriate counter has overflowed.

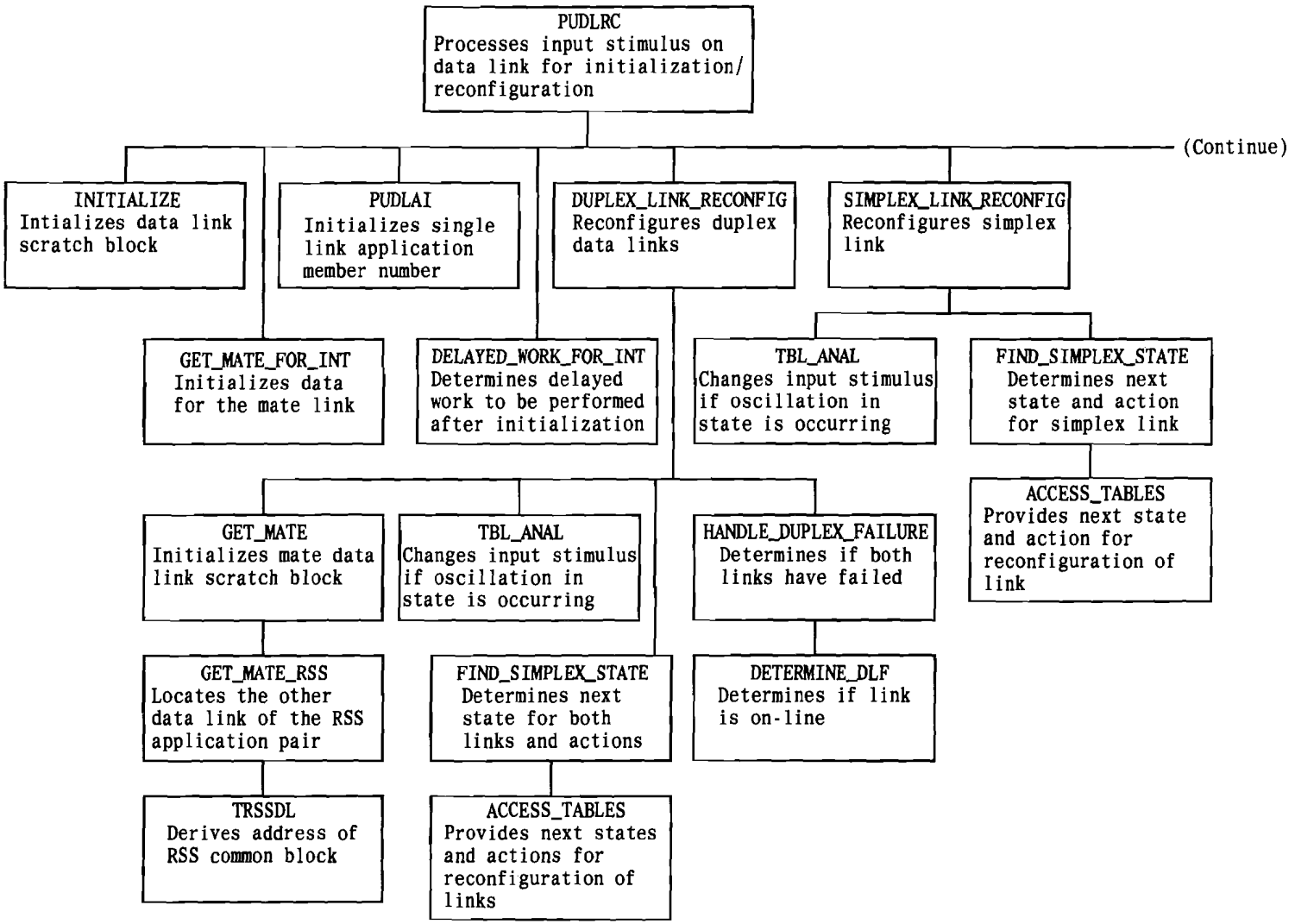


Fig. 9—Subroutine PUDLRC Structural Block Diagram in Pident PUDR (Sheet 1 of 4)

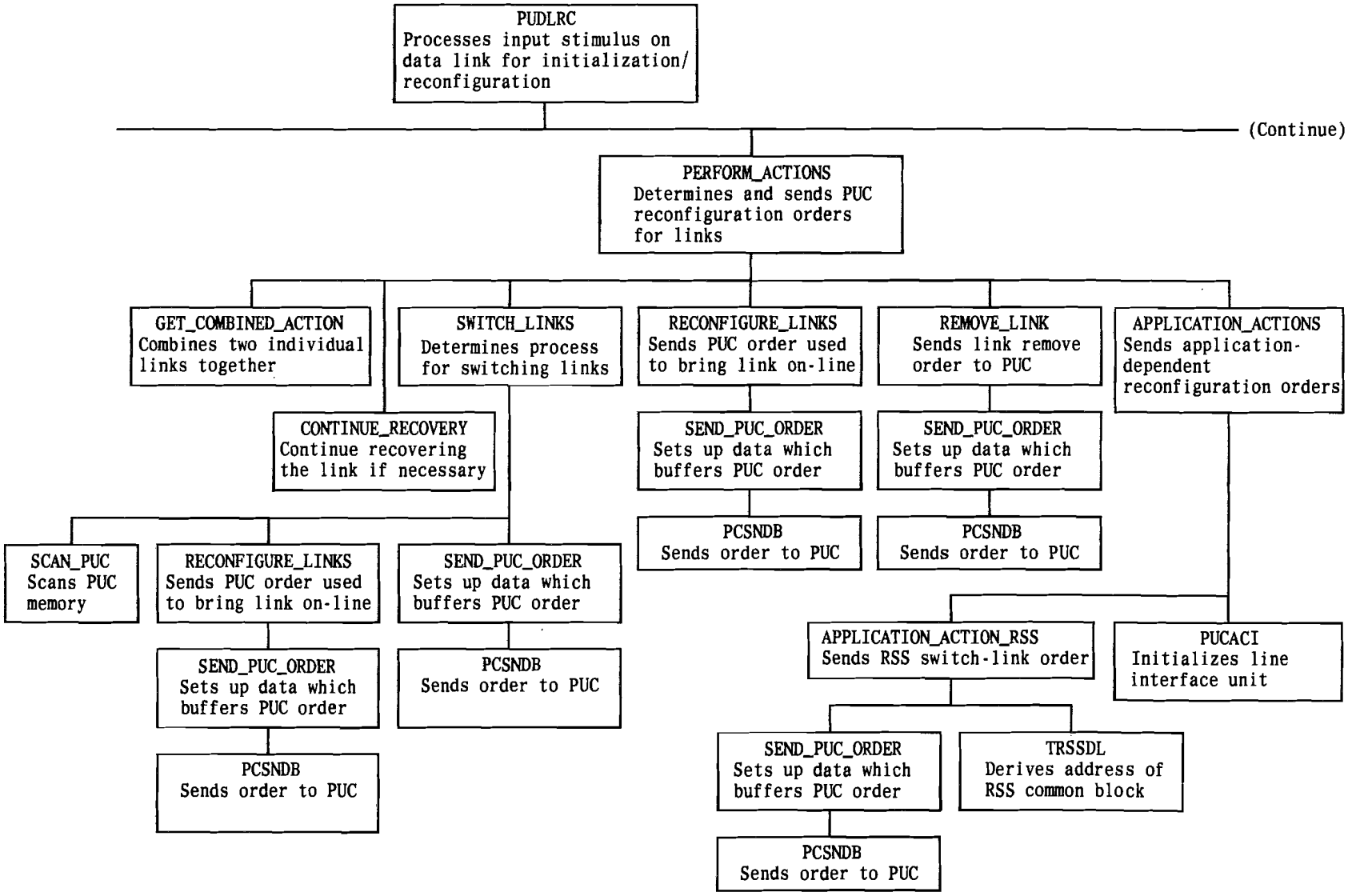


Fig. 9—Subroutine PUDLRC Structural Block Diagram in Pident PUDR (Sheet 2 of 4)

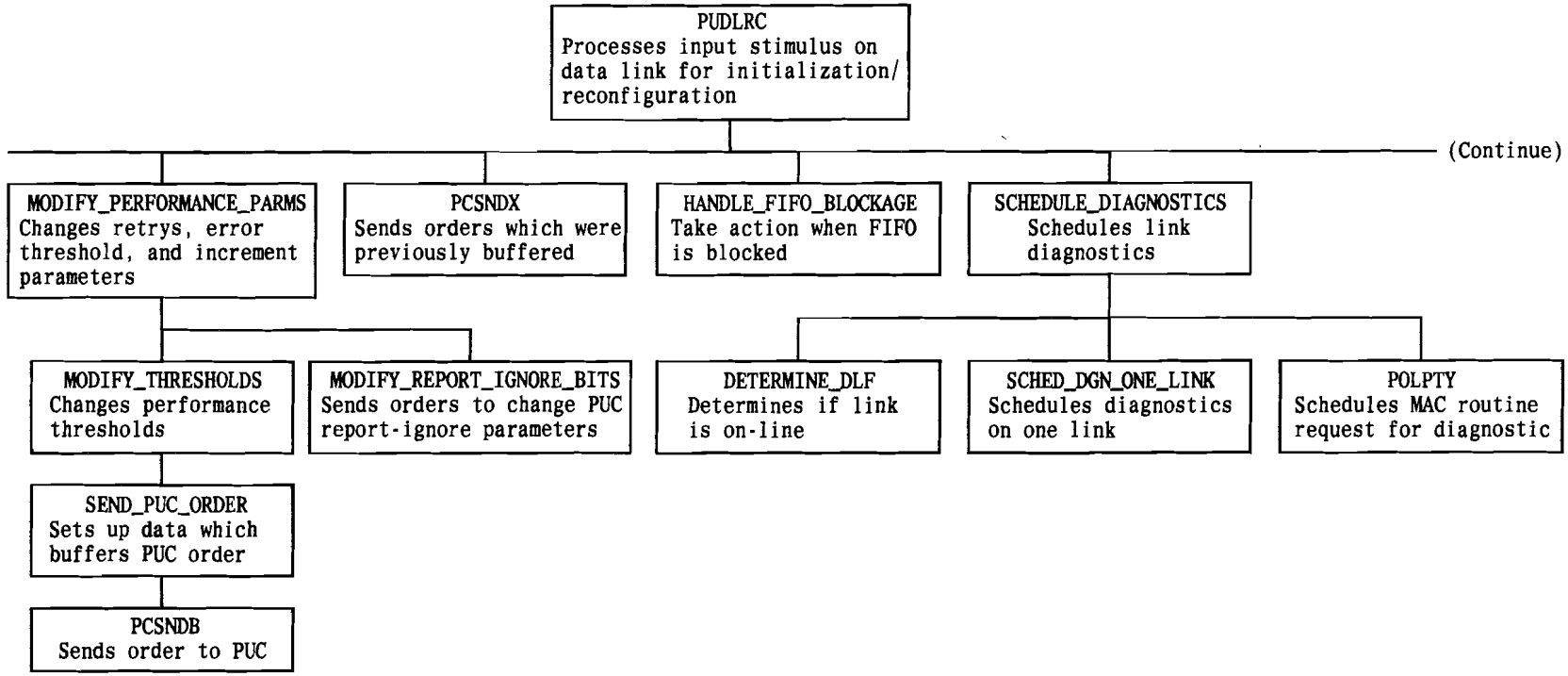


Fig. 9—Subroutine PUDLRC Structural Block Diagram in Pident PUDR (Sheet 3 of 4)

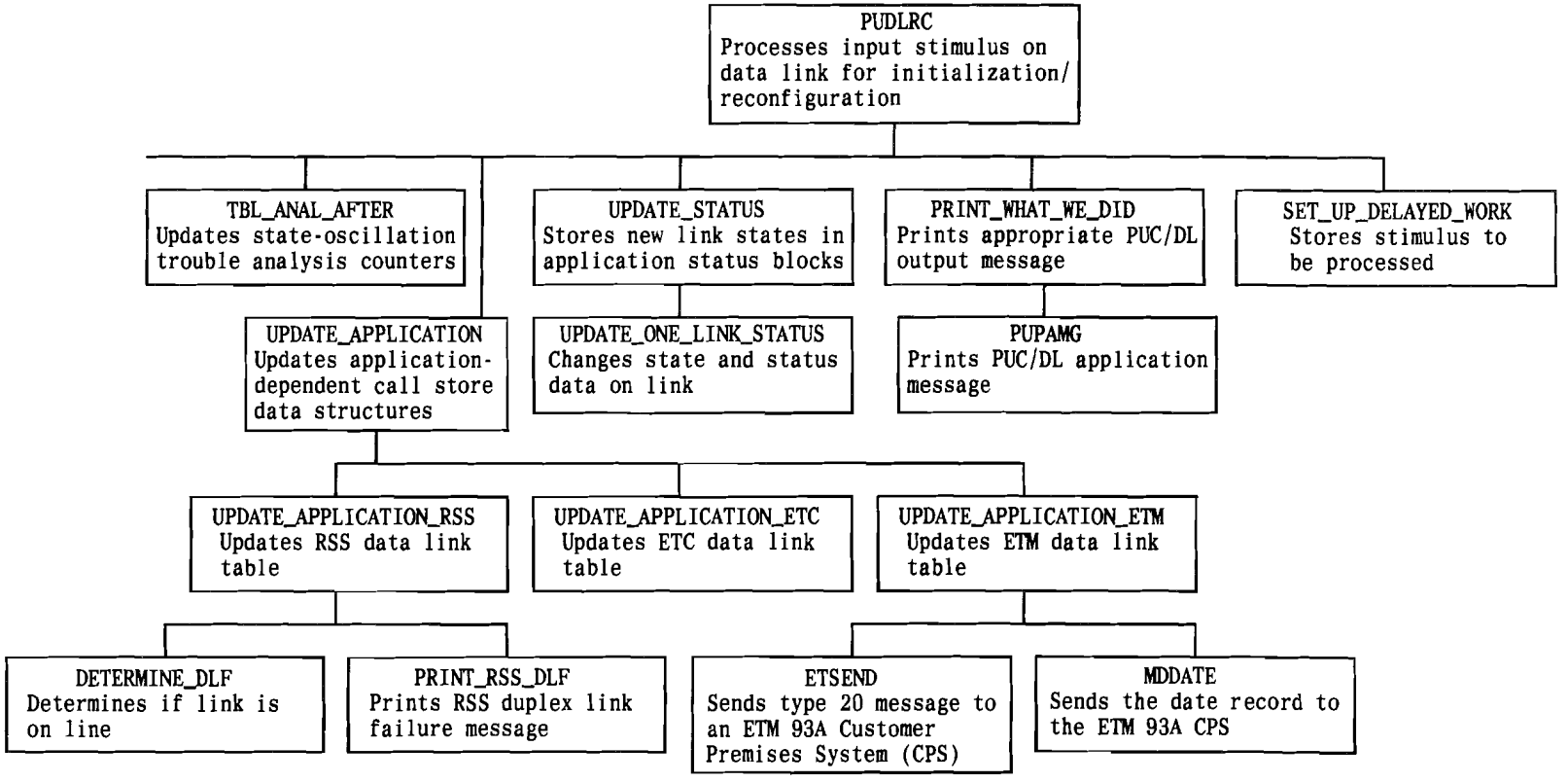


Fig. 9—Subroutine PUDLRC Structural Block Diagram in Pident PUDR (Sheet 4 of 4)

3.18 The reconfiguration next states and actions for both the designate and mate links are derived by subroutine FIND_DUPLEX_STATE. This subroutine performs the following actions:

(1) Determines the proper set of state tables to be used

(2) Accesses the tables via subroutine ACCESS_TABLES.

3.19 Subroutine HANDLE_DUPLEX_FAILURE performs the following actions:

(1) Determines if the designate and mate links have failed

(2) Attempts to bring link on-line in case of a failure via subroutine DETERMINE_DLF.

Simplex Link Reconfiguration (SIMPLEX_LINK_RECONFIG)

3.20 The reconfiguration of a simplex link is handled by subroutine SIMPLEX_LINE_RECONFIG. This subroutine calls the following subroutines:

(a) TBL_ANAL—Accesses the state tables for next states

(b) FIND_SIMPLEX_STATE—Modifies stimulus if a change in state is occurring.

3.21 Subroutine TBL_ANAL is called by SIMPLEX_LINK_RECONFIG to examine the trouble analysis counters in the application status block assigned to the link.

3.22 The SIMPLEX_LINK_RECONFIG subroutine also calls FIND_SIMPLEX_STATE to determine the next state and action required for reconfiguring the simplex link. This module determines the proper state and actions needed to reconfigure the link.

Reconfiguration Action (PERFORM_ACTIONS)

3.23 The PUDLRC calls subroutine PERFORM_ACTIONS to collect the PUC orders to perform the link reconfiguration within the T-scratch buffer. This subroutine calls the following subroutines:

(a) GET_COMBINED_ACTION—Combines links for processing

(b) SWITCH_LINKS—Determines link switching requirements and sends orders to PUC for switching links

(c) RECONFIGURE_LINKS—Sends PUC order to reconfigure links

(d) REMOVE_LINK—Removes link from on-line state

(e) APPLICATION_ACTIONS—Sends application dependent reconfiguration order to PUC.

3.24 Subroutine GET_COMBINED_ACTION utilizes the combined action table PUDXCA for combining the individual actions required for the reconfiguration of the designate and mate links.

3.25 The switching requirements for the links are determined by subroutine SWITCH_LINKS. This subroutine first checks to see if a switching order is currently in progress. If this is the condition, the counter in the application status block of the data link is incremented. Then, if this increment causes the counter to overflow, an attempt is made to hard-switch the links. However, if there is not a switching order in progress, a soft-switch order is initiated which sets up a deferred reconfiguration action. This deferred action expects a link-restore response. Then, if the link-restore response has not been received on the previous link at this time indicating that there might be a remote terminal link-switch message in the transmit buffer, the transmit buffer is cleared and a hard-switch order is attempted.

3.26 If a soft-switch is to be performed, subroutine SWITCH_LINKS calls SEND_PUC_ORDER to set up the necessary input data required for PCSNDB to send the buffered order.

3.27 The hard-switch of the links is performed by subroutine RECONFIGURE_LINKS. This subroutine brings the link on-line.

3.28 In case the link fails to come on-line, subroutine REMOVE_LINK is called by PERFORM_ACTIONS to remove the link order.

3.29 The reconfiguration action for the data link application is performed by subroutine

APPLICATION_ACTIONS. Subroutine APPLICATION_ACTION_RSS is called to send the order for the RSS application and subroutine PUCACI is called to send the order for the ETS application.

Modification of Parameters (MODIFY_PERFORMANCE_PARMS)

3.30 The PUC orders for changing the switching retrys, error thresholds, and error increments parameters are collected by subroutine MODIFY_PERFORMANCE_PARMS. This subroutine calls MODIFY_REPORT_IGNORE to send orders that will change the PUC report-ignore parameters.

Schedule Diagnostics (SCHEDULE_DIAGNOSTICS)

3.31 Whenever there is a blockage of the FIFO because of overflow, PUDLRC calls subroutine SCHEDULE_DIAGNOSTICS to check the diagnostic inhibit flag. If it is not set, subroutine SCHED_DGN_ONE_LINK is called to schedule a diagnostic on the link. If the flag is set, subroutine PDLPTY is called to schedule the MAC routine request.

Data Link Call Store Table Update (UPDATE_APPLICATION)

3.32 The data link tables in call store that are associated with the data link application are updated by subroutine UPDATE_APPLICATION whenever a reconfiguration of the data links is performed. The data link table associated with the RSS application is updated by subroutine UPDATE_APPLICATION_RSS. This subroutine resets the counters in the table to their proper values if the transmit buffer is clear. In the case of a duplex failure, subroutine DETERMINE_DLF is called to determine if the link is on-line. Subroutine PRINT_RSS_DLF is then called to print the failure message on the duplex data link. Whenever the failure occurs, this subroutine causes the output message REPT RSS AA DUPLEX DATA LINK FAILURE to be printed. Whenever the failure ends, output message REPT RSS AA DUPLEX DATA LINK FAILURE CLEARED is printed.

3.33 Subroutine UPDATE_APPLICATION_ETC is called to update the call store table items for the ETC (ETS-CACS) data link application.

Data Link Status Update (UPDATE_STATUS)

3.34 Entry is made from PUDLRC into subroutine UPDATE_STATUS to store the new data link states into the application status block of the data link in call store. It calls module UPDATE_ONE_LINK_STATUS to store the states for the links.

PUC/DL Output Message (PRINT_WHAT_WE_DID)

3.35 The initialization and reconfiguration that was performed is printed out indicating what was actually done. The message is based primarily upon the execution code. Subroutine PUPAMG in pident PUTY is called to format and output the data. The message is printed via the PRINT macro.

PUC/DL INITIALIZATION—PUID

A. General

3.36 The primary purpose of PUID is to initialize all the defined data links on a PUC/DL frame.

B. Functional Routines

3.37 The following subroutines serve as external interface entry points to PUID:

(a) PUDFRI—Pumps up PUC frame when restored

(b) PDLLBP—Loads PUC/DL buffer parameters

(c) PDLLDP—Loads PUC/DL parameters

(d) PDLRPI—Initializes PUC report-ignore parameter

(e) PUDLIN—Initializes all data links on a PUC frame

(f) PUDLAI—Initializes application data links associated with a member number

(g) PUDRPI—Concatenates PUC report-ignore parameter.

PUC Restoration Report (PUDFRI)

3.38 Subroutine PUDFRI is entered from PUPM to report the restoration of the PUC frame to the

data link maintenance routines. It sets up the input data (PUC member number, enable table address) that is required by the initialization routine. After the data is identified, subroutine PUDLIN is called to initialize the data link.

Report-Ignore Parameter Initialization (PDLRPI)

3.39 An external entry is made into subroutine PDLRPI to ensure that the report-ignore information in the PUC and central control match for each data link. This entry calls subroutine PUDRPI to concatenate the report-ignore data in the application status blocks for all the data links.

Load Data Link Parameters (PDLLDP)

3.40 The PDLLDP serves as the external interface entry point for loading the PUC/DL parameters.

Buffer Data Link Parameters (PDLLBP)

3.41 The PDLLBP serves as the external interface entry point for buffering the PUC/DL parameters.

Data Link Initialization (PUDLIN)

3.42 Subroutine PUDLIN (Fig. 10) is entered from subroutine PUDFRI. Its function is to initialize all the defined data links that are assigned to a given PUC frame. The initialization functions consist of:

(a) Setting the report-ignore data in the application status blocks for the data links

(b) Initializing the peripheral unit enable block, application data structures, and status block

(c) Initializing the PUC/DL parameters

(d) Sending the PUC report-ignore parameters to PUC.

Data Structure Initialization (PUDSBI)

3.43 Subroutine PUDSBI is called to initialize the memory data area associated with the PUC/DL frame. This data area is referred to as the sample block in the signal processor office; it is referred to as the peripheral unit enable block in the central office. The PUC hopper parameter table contains the pointer into the memory area.

Application-Dependent Data Structure Initialization (PUDAPI)

3.44 The data link application-dependent memory data is initialized by subroutine PUDAPI. It determines the data link application and calls subroutine PURSSI to initialize the memory for the RSS application. Subroutine PUETSI is called to initialize the area for the ETS application.

Application Status Block Initialization (PUDASI)

3.45 The application status blocks for all the data links associated with the application member number are initialized by subroutine PUDASI. It determines the data link application and calls the particular subroutine to perform the initialization. Subroutine PURASI is called for the RSS application. Subroutine PUCASI is called for the ETS application. The address of the RSS common block is obtained by subroutine TRSSDL. The ETS common block data is translated by subroutine TRSPNC.

Data Link Parameter Initialization (PUCIN)

3.46 The initialization of the PUC/DL parameters for each link is controlled by subroutine PUCIN. This subroutine checks the state of the link stored in the application status block and calls the following subroutines to load and send the appropriate PUC order:

- LOAD_PUC_DL_PARMS
- LOAD_PUC_BUFFER_PARMS
- ACTIVATE_LINK
- DEACTIVATE_LINK.

3.47 Subroutine LOAD_PUC_DL_PARMS is entered either from PUCIN or from PDLLDP. It loads the PUC/DL parameters into the PUC memory and sends orders pertaining to the following:

(a) Buffer linkage

(b) Protocol number

(c) Maximum number of data frames that may be left outstanding

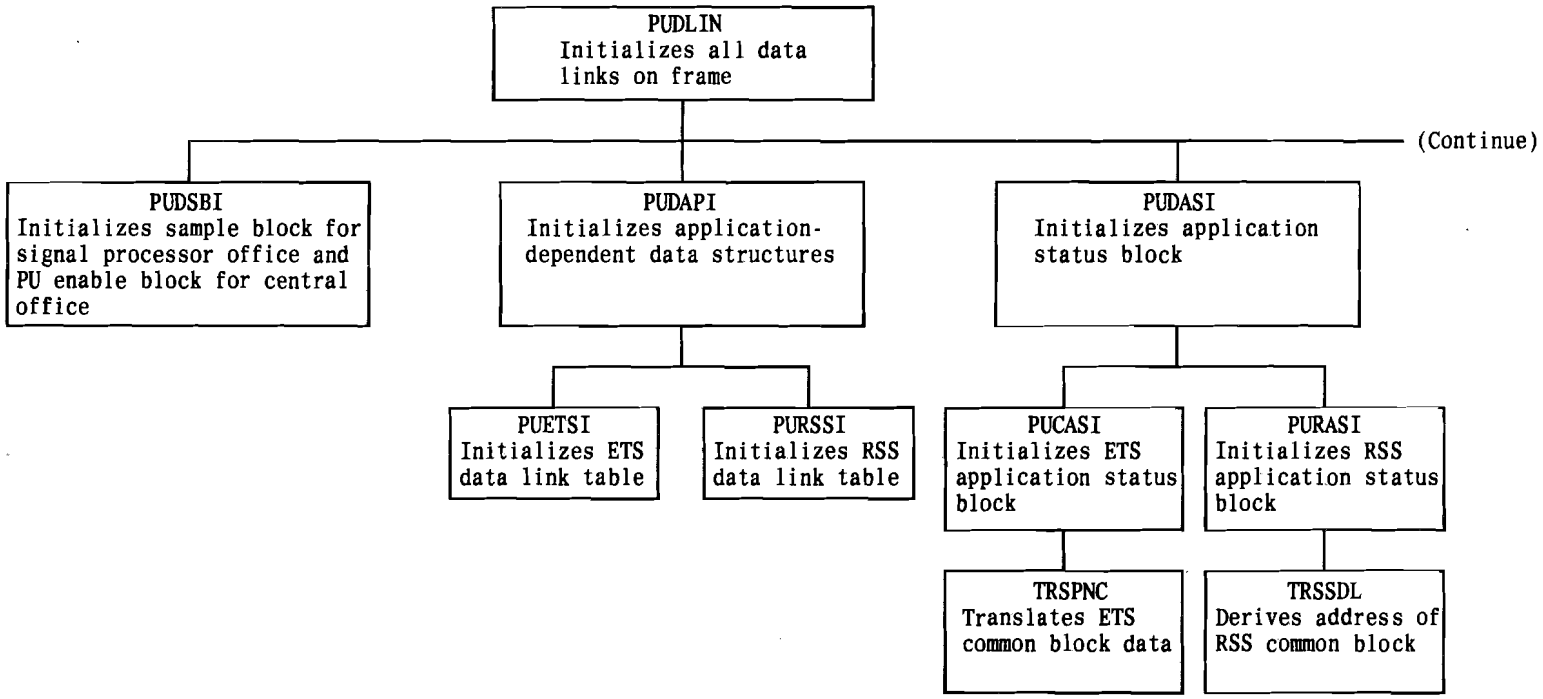


Fig. 10—Subroutine PUDLIN Structural Block Diagram in Pident PUID (Sheet 1 of 3)

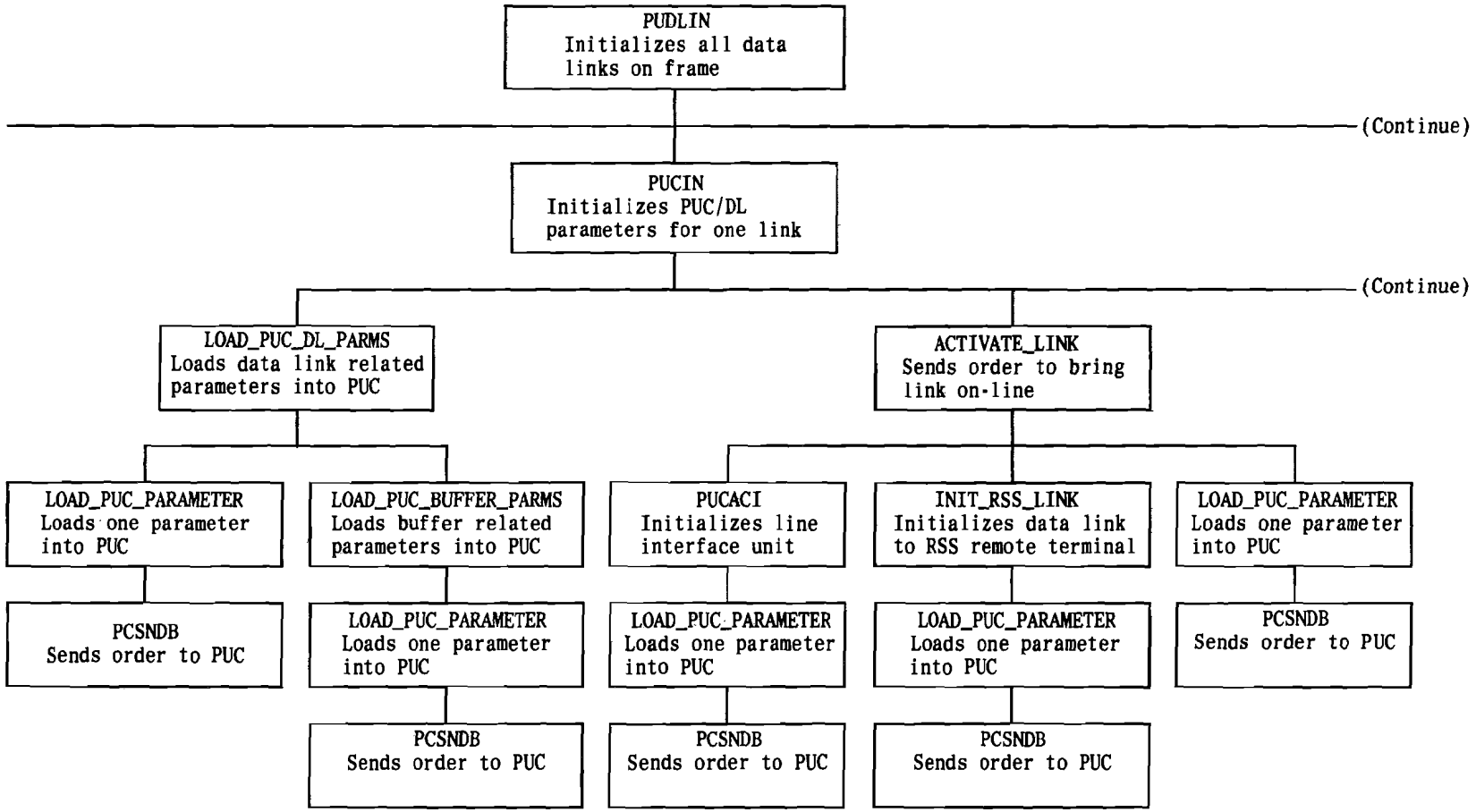


Fig. 10—Subroutine PUDLIN Structural Block Diagram in Pident PUID (Sheet 2 of 3)

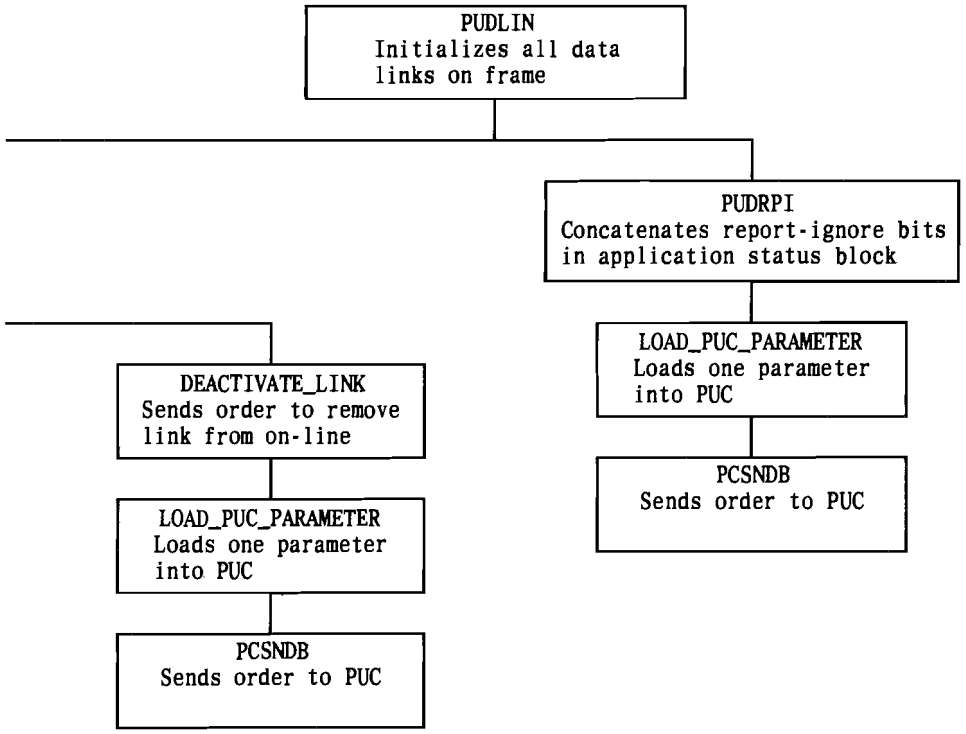


Fig. 10—Subroutine PUDLIN Structural Block Diagram in Pident PUID (Sheet 3 of 3)

(d) Maximum number of retrys to be performed

(e) Time-out

(f) Error threshold

(g) Error increment.

3.48 Subroutine LOAD_PUC_BUFFER_PARMS is called by PUCIN to load and send the following data which defines the transmit and receive buffers associated with a particular application member number:

(a) Transmit buffer length

(b) Receive buffer SCAM allocation

(c) Scratch block length.

3.49 The data link is activated by subroutine ACTIVATE_LINK. The activation function includes the loading of the data link parameter into the PUC and bringing the link on-line. The parameter is loaded and sent to the PUC by subroutine LOAD_PUC_PARAMETER. The ACTIVATE_LINK brings the link on-line by calling the specific subroutine for the application. Subroutine INIR_RSS_LINK is called which sends a 3-word message to be loaded in the RSS transmit buffer. This message indicates which link is being activated on the line. Subroutine PUCACI is called to send the orders which initialize the line unit.

3.50 The PUCIN calls subroutine DEACTIVATE_LINK to remove the link from on-line. The remove order is sent by subroutine LOAD_PUC_PARAMETER and PCSNDB.

Report-Ignore Parameter Initialization (PUDRPI)

3.51 The PUDRPI is called by PUDLIN to concatenate the report-ignore bit data in the application status blocks associated with all the links to ensure that this data in the PUC and central control will match for each link.

Data Link Application Parameter Initialization (PUDLAI)

3.52 The PUC/DL fault recovery pident PUDR enters PUID at subroutine PUDLAI (Fig. 11) to initialize the application data link. This subroutine initializes only those links that are associated with a given data link application member number. Its actions include:

(a) Searching the PUC hopper parameter table for the application and member number

(b) Setting up a pseudo-application status block for the purpose of buffering the PUC order

(c) Buffering the PUC order required for initialization

(d) Sending the orders

(e) Initializing the call store memory

(f) Sending the PUC report-ignore parameter.

3.53 The PUDLAI performs a search of the PUC hopper parameter table to determine the application and member number of the data link. Subroutine PSEUDO_INIT_RSS is called to initialize the pseudo status block for the RSS application. Subroutine PSEUDO_INIT_ETS is called to initialize the pseudo status block for the ETS application.

Data Link Parameter Initialization (PUCIN)

3.54 The actions taken by subroutine PUCIN to initialize the data link parameters for subroutine PUDLAI are the same as those actions performed by PUCIN in subroutine PUDLIN.

Application Data Link Initialization (PUDLAI_SEND_ORDERS)

3.55 The purpose of subroutine PUDLAI_SEND_ORDERS is to initialize a single data link application member number.

3.56 Subroutine PUDSBI is called to initialize the memory data area associated with data link application member number. For the signal processor office, this data is referred to as the sample block. For the central office, it is referred to as the peripheral unit enable block. The PUC hopper parameter table contains the pointer into the memory area.

3.57 Subroutine PUDAPI is called to initialize the application data structure for the data link. This subroutine calls subroutine PURSSI to initialize the RSS data link table. Subroutine PUETSI is called

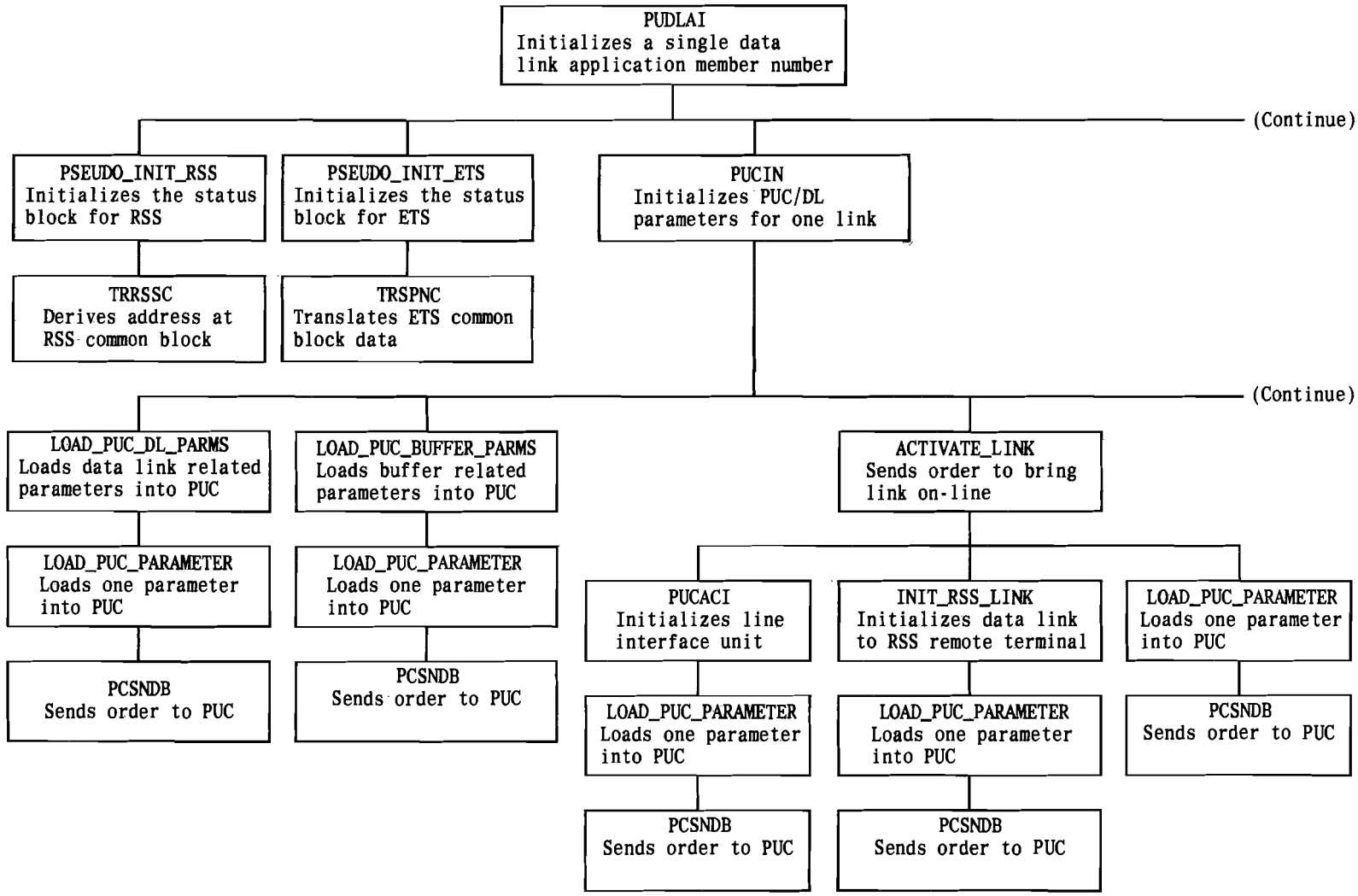


Fig. 11—Subroutine PUDLAI Structural Block Diagram in Pident PUID (Sheet 1 of 3)

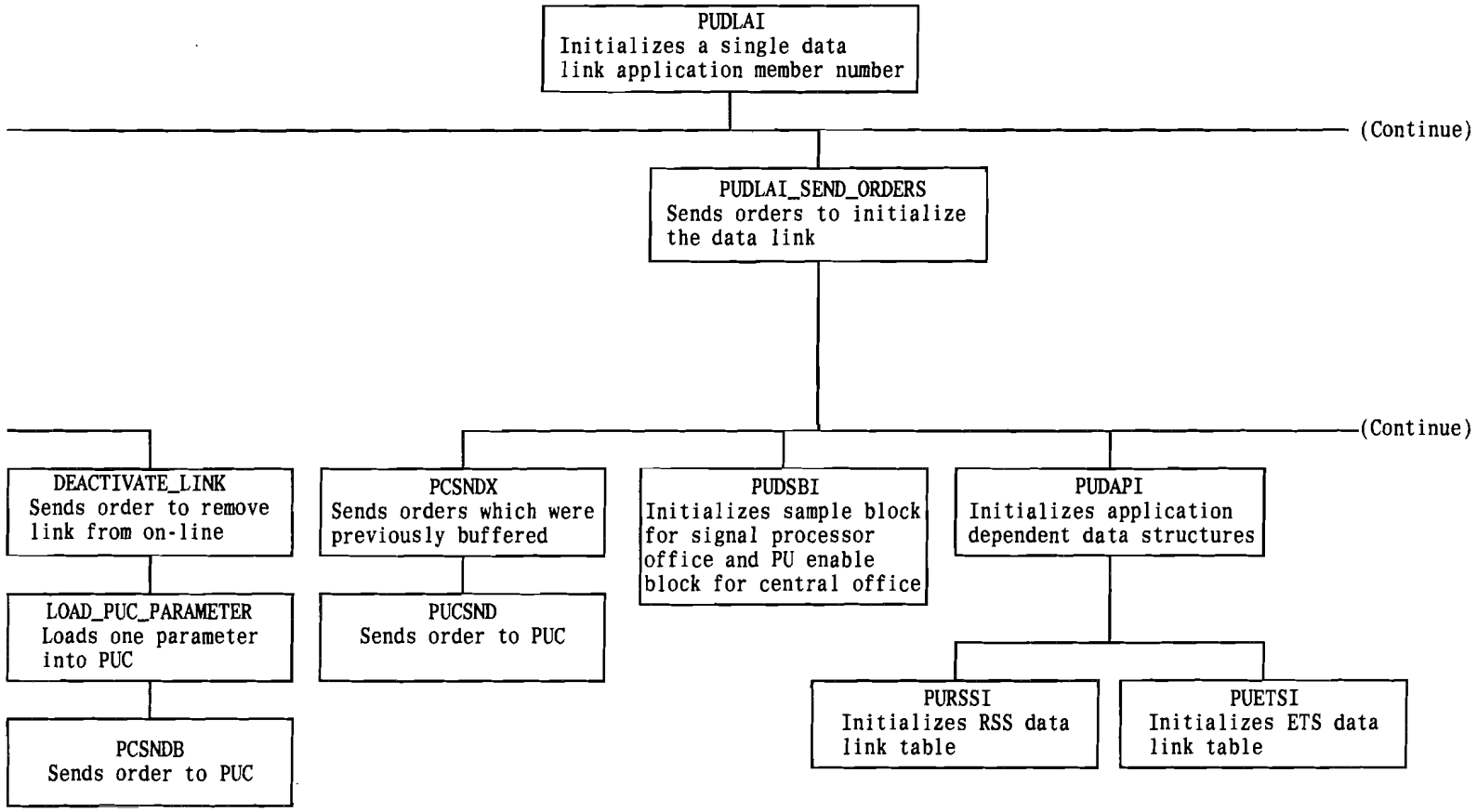


Fig. 11—Subroutine PUDLAI Structural Block Diagram in Pident PUID (Sheet 2 of 3)

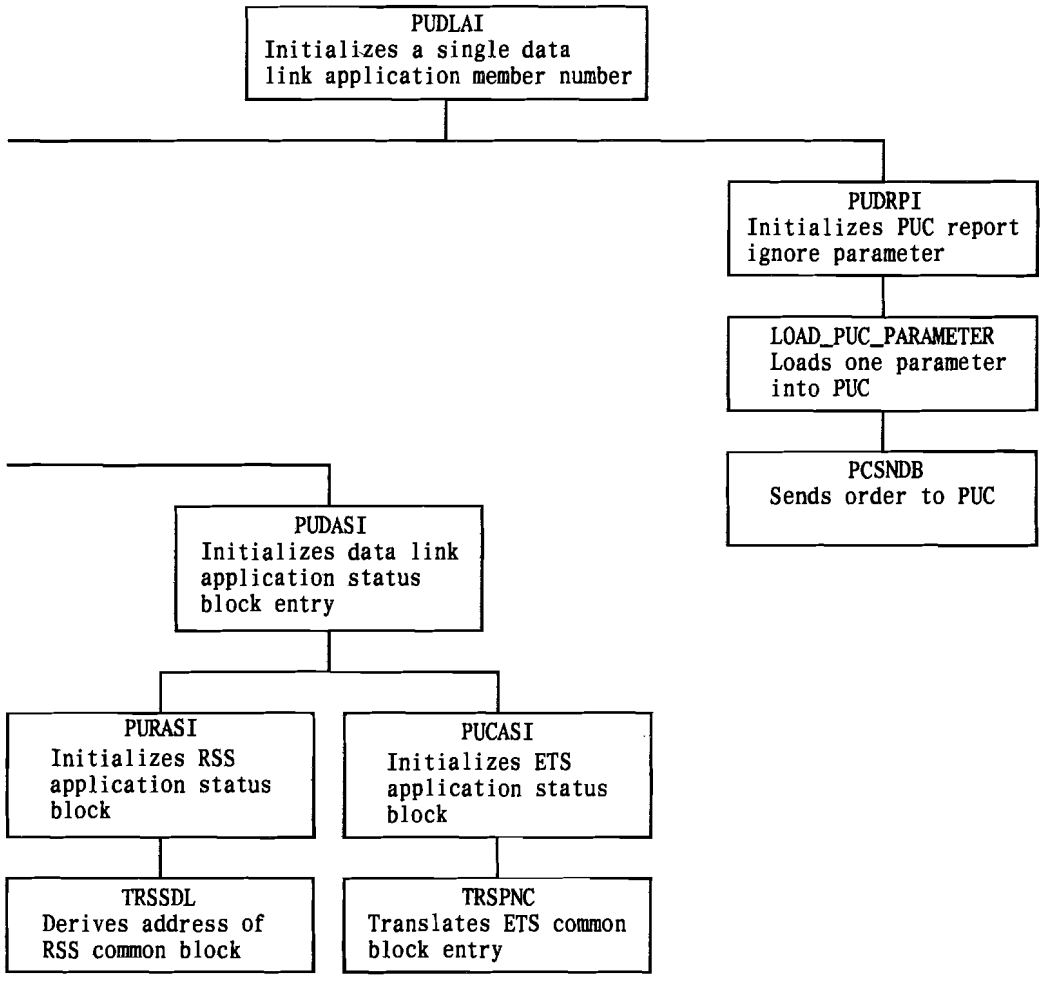


Fig. 11—Subroutine PUDLAI Structural Block Diagram in Pident PUID (Sheet 3 of 3)

to initialize the data link table for the ETS application.

3.58 The application status block of the data link application is initialized by subroutine PUDASI. Depending upon the data link application, the specific subroutine is called to initialize the status blocks associated with the data link application. Subroutine PURASI is called to initialize the status block associated with the RSS application. Subroutine PUCASI is called to initialize the status block associated with the ETS application. The address of the RSS common block is obtained by subroutine TRSSDL, whereas the common block associated with the ETS application is processed by subroutine TRSPNC.

Report-Ignore Parameter Initialization (PUDRPI)

3.59 The PUDRPI is called by subroutine PDLRPI to concatenate the report-ignore data in the data link application status blocks associated with the links. This concatenation ensures that the report-ignore data in the PUC and central control matches for each data link application.

PUC/DL TTY INTERFACE—PUTY

A. General

3.60 The PUTY pident provides the routines for processing TTY input messages PDL_APPLIC and PDL_LNK. Routines are also provided for processing PUC/DL output messages pertaining to the operation and maintenance of the PUC data links. This pident contains the following translation tables:

(a) PUDROX—Translates stimulus to American Standard Code for Information Interchange (ASCII) code.

(b) PUSTMP—Translates stimulus print priority

(c) PUDDPX—Translates disposition to ASCII code

(d) PUDSTX—Translates state to ASCII code

(e) PUDLAV—Verifies application.

B. Functional Routines

3.61 The two subroutines used to process TTY input messages are as follows:

(a) PUDATI—Processes input message PDL_APPLIC

(b) PUDLTI—Processes input message PDL_LNK

Data Link Application Request (PUDATI)

3.62 Subroutine PUDATI (Fig. 12) serves as the interfacing entry point from pident TTIA (TTY Input Message Directory and Catalog pident) to process TTY input message PDL_APPLIC. This message allows the subroutine PUDATI to:

- Obtain a status report on links
- Switch duplex links
- Initialize links.

3.63 The format of the message is PDL_APPLIC_AAA BBB CC where:

AAA = Request

BBB = Application name

CC = Data link member number.

3.64 Subroutine PUDATI calls subroutine EXECUTE_TTY_APPLIC_REQUEST to process the message. This subroutine then calls an appropriate subroutine to perform a specific action on the message. The called subroutines are:

- TRANSLATE_REQUEST
- TRANSLATE_APPLICATION
- FIND DATA_LINK
- SCHEDULE_REQUEST.

3.65 The ASCII request characters AAA of the input message are checked and converted into a processing stimulus by subroutine TRANSLATE_REQUEST. If the input request is not valid, an NG (no good) execution code is returned.

3.66 The application name BBB in the input message is converted into binary information by subroutine TRANSLATE_APPLICATION. This subroutine utilizes the information contained in transla-

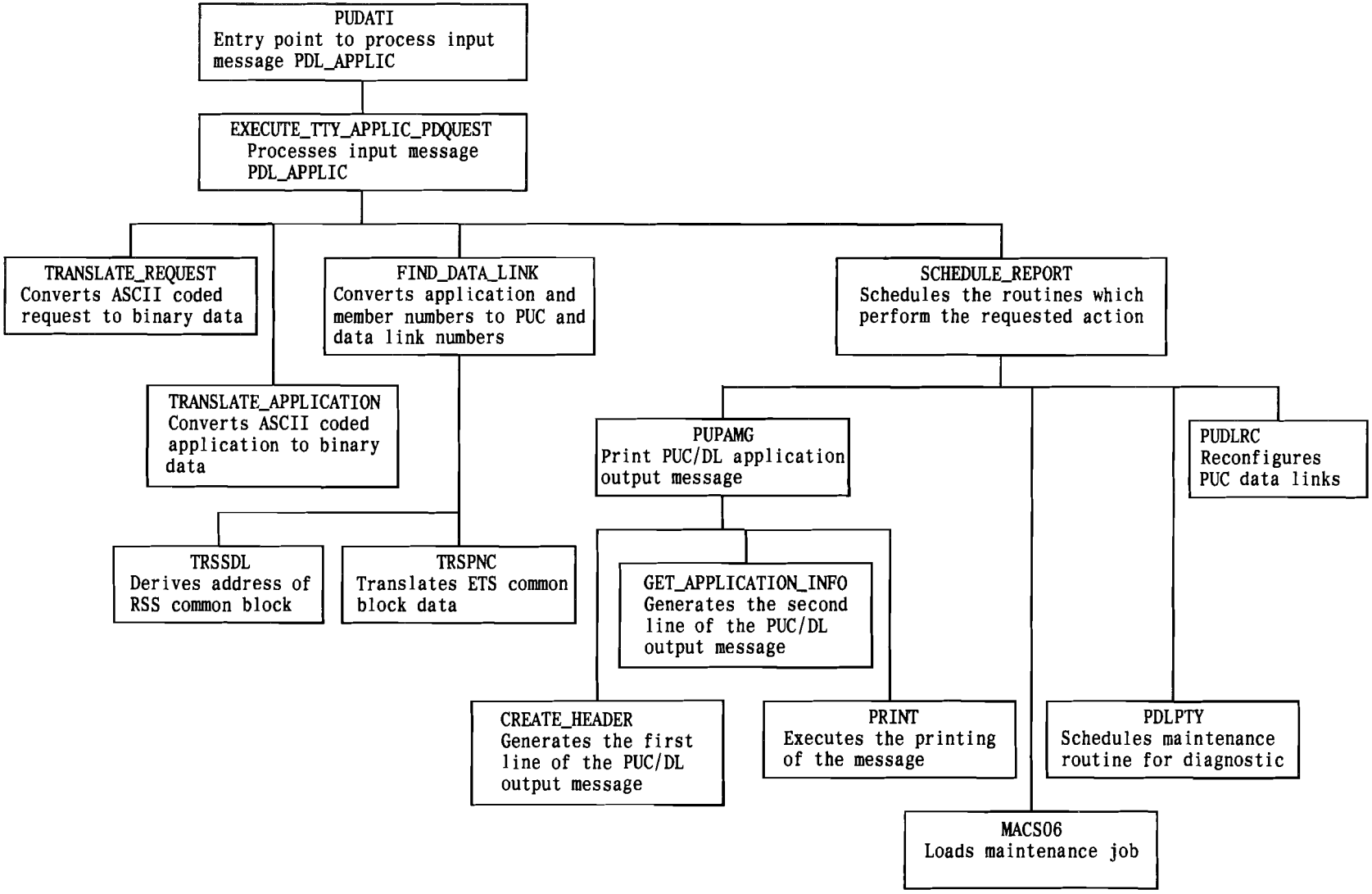


Fig. 12—Subroutine PUDATI Structural Block Diagram in Pident PUTY

tion table PUDLAV to verify and translate the application name.

3.67 The data link application member number CC is translated by subroutine FIND_DATA_LINK. Translator TRSSDL is called to derive the address of the RSS common block. The TRSPNC is called to derive the address of the ETC/ETM common block (ETC represents the ETS-CACS common block, and ETM represents the ETS-message detail recording common block). If the application member number is invalid or unequipped, an NG execution code is returned.

3.68 The input message request, after being screened and converted, is then handled by subroutine SCHEDULE_REQUEST. Depending upon the requested information, the following subroutines are called.

(a) PUPAMG—Prints the PUC/DL output message. (Paragraph 3.77 describes the printing functions of this subroutine.)

(b) MACSO6—Loads the maintenance job and requests reconfiguration of links.

(c) PDLPTY—Schedules maintenance routine request for diagnostic.

(d) PUDLRC—Initializes/reconfigures data link.

Data Link Maintenance Request (PUDLTI)

3.69 Subroutine PUDLTI serves as the interfacing entry point from pident TTIA to process TTY input message PDL_LNK. This message allows the subroutine PUDLTI to:

- Restore link to service
- Switch link with its mate
- Remove an active link
- Detain an out-of-service link from being automatically restored to service
- Force a link to become active
- Force a link to become unavailable
- Determine the status of a link
- Diagnose a link.

3.70 The format of the message is PDL_LNK_AAA BB CC where:

AAA = Request

BB = PUC member number

CC = Data link member number.

3.71 Subroutine PUDLTI calls subroutine EXECUTE_TTY_DL_REQUEST to process the message. This subroutine then calls the appropriate subroutine to perform a specific action on the message. The called subroutines include:

- TRANSLATE_REQUEST
- TRANSLATE_DL_MEMN
- VALIDATE_MEMBER_NUMBERS
- SCHEDULE_REQUEST.

3.72 Subroutine TRANSLATE_REQUEST converts the input message request ASCII characters AAA into a processing stimulus. If the input request is not valid, an NG execution code is returned.

3.73 The data link member number, specified by CC, is converted into binary information by subroutine TRANSLATE_DL_MEMN. If the specified member number is not equipped, an NG response is returned. If the member number is specified as AL and the request indicates STR, a printout is returned on the status of all data links on the specified frame.

3.74 The PUC member number, specified by BB, is validated by subroutine VALIDATE_MEMBER_NUMBERS. It calls subroutine TRUTYN to obtain the application and application member numbers from the associated auxiliary block.

3.75 After the input message is screened and the ASCII characters converted and translated, subroutine SCHEDULE_REQUEST is called to handle the request. Depending upon the input information, the following subroutines are called:

(a) PUPAMG—Prints the PUC/DL output message

(b) MACSO6—Loads the maintenance job and requests reconfiguration of links

(c) PDLPTY—Schedules maintenance routine request for diagnostic

(d) PUDLRC—Initializes/reconfigures data link.

3.76 The following subroutines are used to process PUC/DL output messages:

(a) PUPAMG—Prints the PUC/DL output message

(b) PUPPST—Prints the status of all data links.

Data Link Application Report (PUPAMG)

3.77 The printing function for the PUC/DL output message is handled by subroutine PUPAMG. The following subroutines are called to translate, format, and print the output message:

(a) CREATE_HEADER—Translates and formats input data for first line of report

(b) GET_APPLIC_INFO—Translates and formats data for second line of report

(c) PRINT—A macro used for printing messages.

3.78 The TTY input message ASCII characters (subroutines PUDATI and PUDLTI) are stored by subroutine CREATE_HEADER in call store. It translates and formats the following items for the first line of the report:

- Designate data link member number
- PUC member number
- Stimulus
- Disposition.

3.79 The application status block data is translated and formatted by subroutine GET_APPLIC_INFO for the second line of the output message report. This line contains:

- Data link application
- Data link member number
- Data link state
- Data link state qualifier.

3.80 The macro PRINT is called to manage the printing of the output message. The pointer to the output data area is received from the CREATE_HEADER and GET_APPLIC_INFO subroutines. Macro PRINT then calls subroutine TTOUTM to load the data in the output message buffer for the printing functions.

Data Link Status Report (PUPPST)

3.81 If the request is made to print the status of all data links on a frame, subroutine PUPPST is called. This subroutine then calls PRINT_STATUS_ALL to load the data in the output message buffer for the print routines.

PUC/DL TABLES—PUDT

A. General

3.82 The PUDT pident contains four translation tables used by the PUTY pident. The tables are:

(a) PURDNS—Duplex links designate next state

(b) PURMNS—Duplex links mate next state

(c) PURDAT—Duplex links designate action

(d) PURMAT—Duplex links mate action.

B. Functional Routines

3.83 The PUDT pident does not contain any executable code. It provides tables for translating the next states and actions for the duplex links.

Duplex Links Designate Next State Table (PURDNS)

3.84 The Duplex Links Designate Next State Table is used by subroutine SCHEDULE_REQUEST (pident PUTY) to obtain the next valid state required for maintenance and diagnostics of the data links. Table A depicts the stimulus and resulting state of the designate link when the designate link is in a given state.

Duplex Links Mate Next State Table (PURMNS)

3.85 Subroutine SCHEDULE_REQUEST (pident PUTY) utilizes the Duplex Links Mate Next Table to obtain the next valid state for the mate link

TABLE A

DUPLEX LINKS DESIGNATE NEXT STATE TABLE

STIMULUS	DESIGNATE LINK PRESENT STATE — NEXT STATE										
	ACTIVE	STANDBY	ACTIVE DEGRADED	OOS DEGRADED	OOS MANUAL	OOS REMOVED	OOS FAULT	ACTIVE FORCED	UAVLBL	OOS TBL ANAL	OOS RMV ACTIVE
Designate Mate OOS	Active Degraded	—	—	—	—	—	—	—	—	—	—
TTY Switch	Standby	Active	—	—	—	—	—	—	—	—	—
TTY Remove	OOS Manual	OOS Manual	OOS Manual	—	—	—	—	OOS Manual	—	—	—
TTY Force Act	Active Forced	Active Forced	Active Forced	Active Forced	Active Forced	Active Forced	Active Forced	—	Active Forced	Active Forced	Active Forced
TTY Force Uavlbl	Uavlbl	Uavlbl	Uavlbl	Uavlbl	Uavlbl	Uavlbl	Uavlbl	Uavlbl	—	Uavlbl	Uavlbl
Diagnostic	Standby	—	OOS Degraded	—	—	—	—	OOS Removed	—	—	—
Carrier Fail	OOS Removed	OOS Removed	OOS Removed	OOS Removed	—	—	—	—	—	—	—
Excessive Error Rate	OOS Degraded	—	OOS Removed	—	—	—	—	—	—	—	—
Link Restore	Active	—	Active Degraded	—	—	—	—	Active Forced	—	—	Active Degraded
Circuit Failure	OOS Removed	OOS Removed	OOS Removed	OOS Removed	—	—	—	—	—	—	—
Protocol Fail	OOS Removed	—	OOS Removed	—	—	—	—	—	—	—	—
Switch Abort	OOS Removed	—	OOS Removed	—	—	—	—	Switch Abort Link Forced	—	—	—
Trouble Analysis	OOS Tbl Anal	OOS Tbl Anal	OOS Tbl Anal	OOS Tbl Anal	—	OOS Tbl Anal	—	—	—	—	—

TABLE A (Contd)

DUPLEX LINKS DESIGNATE NEXT STATE TABLE

STIMULUS	DESIGNATE LINK PRESENT STATE — NEXT STATE										
	ACTIVE	STANDBY	ACTIVE DEGRADED	OOS DEGRADED	OOS MANUAL	OOS REMOVED	OOS FAULT	ACTIVE FORCED	UAVLBL	OOS TBL ANAL	OOS RMV ACTIVE
Restore Mate Frcd	—	Standby	—	Standby	OOS Manual	Standby	Standby	—	Uavlbl	OOS Tbl Anal	—
Diag STP Mate Act	—	Standby	—	Standby	OOS Manual	Standby	Standby	—	Uavlbl	OOS Tbl Anal	—
Diag ATP	—	Active	—	Active	—	Active	Active	—	—	—	OOS Active
Diag STF Hdw	—	OOS Fault	—	OOS Fault	OOS Manual	OOS Fault	OOS Fault	—	Uavlbl	OOS Tbl Anal	OOS Fault
Diag STF PD	—	OOS Degraded	—	OOS Degraded	OOS Manual	OOS Degraded	OOS Degraded	—	Uavlbl	—	OOS Degraded
Diag STF PU	—	OOS Removed	—	OOS Removed	OOS Manual	OOS Removed	OOS Removed	—	Uavlbl	—	OOS Rmv Act
Diag STP	—	Active	—	Active	—	Active	Active	—	—	—	Active
Perf Desig Mate OOS	—	—	OOS Removed	—	—	—	—	Active Forced	—	—	—
TTY Restore	—	—	Active	OOS Removed	OOS Removed	OOS Removed	OOS Removed	Active	OOS Removed	OOS Removed	OOS Rmv Act
TTY Detain	—	—	—	OOS Manual	—	OOS Manual	OOS Manual	—	OOS Manual	OOS Manual	OOS Manual
Switch Abort Link Forced	—	—	—	—	—	—	—	Active Forced	—	—	—

in response to a given stimulus. The next state for the mate link resulting from a given stimulus is shown in Table B.

Duplex Links Designate Action Table (PURDAT)

3.86 The next action for the designate link is specified in the Duplex Links Designate Action Table. These actions are shown in Table C.

Duplex Links Mate Action Table (PURMAT)

3.87 The next action for the mate link is specified in the Duplex Links Mate Action Table. These actions are shown in Table D.

4. ABBREVIATIONS AND ACRONYMS

4.01 The following is a list of abbreviations and acronyms used in this section.

ACT	Active
ANAL	Analysis
ASM	Auxiliary Scan Memory
ATP	All Tests Passed
CACS	Customer Administration Center System
CCIS	Common Channel Interoffice Signaling
DEG	Degraded
DIAG	Diagnostic
ESS	Electronic Switching System
ETS	Electronic Tandem Switching
FIFO	First-In/First-Out
FRCD	Forced
HDW	Hardware
LCAT	Local Customer Administration System Terminal
NG	No Good
OOS	Out of Service
PERF	Perforate
PD	Pulse Distributor
Pident	Program Identification
PR	Program Listing
PU	Peripheral Unit
PUAB	Peripheral Unit Address Bus
PUC	Peripheral Unit Controller
PUC/DL	Peripheral Unit Controller/Data Link
RMV	Remove
RSS	Remote Switching System
RST	Restore
SCAM	Scanner Answer Memory
STF	Some Tests Failed
STP	Some Tests Passed
TBL	Trouble
UAVLBL	Unavailable
UTS	Universal Trunk Scanner

5. REFERENCES

5.01 The following documents contain information pertaining to the PUC or to equipment used with the PUC.

NUMBER	TITLE
PR-1A455	No. 1 and No. 1A ESS PUC/DL Maintenance Guide

SECTION	TITLE
231-037-020	Peripheral Unit Controller—Description
231-045-410	Digital Carrier Trunk—Software Description
231-045-430	Peripheral Unit Controller—Software Description

TABLE B

DUPLEX LINKS MATE NEXT STATE TABLE

STIMULUS	MATE LINK PRESENT STATE — NEXT STATE										
	ACTIVE	STANDBY	ACTIVE DEGRADED	OOS DEGRADED	OOS MANUAL	OOS REMOVED	OOS FAULT	ACTIVE FORCED	UNAVLBL	OOS TBL ANAL	OOS RMV ACTIVE
Diag STP Mate Act	Active	—	—	—	—	—	—	—	—	—	—
TTY Switch	Standby	Active	—	—	—	—	—	—	—	—	—
TTY Restore	Active	Standby	Active Degraded	OOS Degraded	OOS Manual	OOS Removed	OOS Fault	Active Forced	Uavlbl	OOS Tbl Anal	OOS Rmv Act
TTY Remove	Active	Active	Active Degraded	Active Degraded	—	—	—	Active Forced	—	—	—
TTY Detain	Active	—	Active Degraded	—	OOS Manual	OOS Removed	OOS Fault	Active Forced	Uavlbl	Tbl Anal	OOS Rmv Act
TTY Force Act	Standby	Standby	OOS Degraded	OOS Degraded	OOS Manual	OOS Removed	OOS Fault	—	Uavlbl	OOS Tbl Anal	OOS Removed
TTY Force Uavlbl	Active	Active	Active Degraded	Active Degraded	Active Degraded	OOS Removed	OOS Fault	Active Forced	Uavlbl	Active Degraded	OOS Rmv Act
Diagnostic	Active	Active	Active Degraded	Active Degraded	—	—	—	Active Forced	—	—	—
Diag ATP	Standby	—	OOS Degraded	—	OOS Manual	OOS Removed	OOS Fault	Restore Mate Frcd	Uavlbl	OOS Tbl Anal	OOS Removed
Diag STF Hdw	Active	—	Active Degraded	—	Active Degraded	OOS Removed	OOS Fault	Active Forced	Uavlbl	Active Degraded	OOS Rmv Act
Diag STF PD	Active	—	Active Degraded	—	OOS Manual	OOS Removed	OOS Fault	Active Forced	Uavlbl	OOS Tbl Anal	—
Diag STF PU	Active	—	Active Degraded	—	OOS Manual	OOS Removed	OOS Fault	Active Forced	Uavlbl	OOS Tbl Anal	OS Rmv Act
Diag STP	Diag STP Mate Act	—	OOS Degraded	—	OOS Manual	OOS Removed	OOS Fault	Restore Mate Frcd	Uavlbl	OOS Tbl Anal	—

TABLE B (Contd)

DUPLEX LINKS MATE NEXT STATE TABLE

STIMULUS	MATE LINK PRESENT STATE — NEXT STATE										
	ACTIVE	STANDBY	ACTIVE DEGRADED	OOS DEGRADED	OOS MANUAL	OOS REMOVED	OOS FAULT	ACTIVE FORCED	UAVLBL	OOS TBL ANAL	OOS RMV ACTIVE
Carrier Fail	Active	Active	Active Degraded	Active Degraded	Active Degraded	—	OOS Fault	Active Forced	Uavlbl	Active Degraded	—
Circuit Failure	Active	Active	Active Degraded	Active Degraded	Active Degraded	—	OOS Fault	Active Forced	Uavlbl	Active Degraded	OOS Rmv Act
Trouble Analysis	Active	Active	Active Degraded	Active Degraded	Active Degraded	—	OOS Fault	Active Forced	Uavlbl	OOS Tbl Anal	OOS Rmv Act
Switch Abort Link Forced	—	Standby	—	OOS Degraded	OOS Manual	OOS Removed	OOS Fault	—	—	—	—
Excessive Error Rate	—	Active	—	Active Degraded	Perf Deg Mate OOS	—	Perf Deg Mate OOS	—	Perf Deg Mate OOS	Perf Deg Mate OOS	—
Link Restore	—	Standby	—	OOS Degraded	OOS Manual	—	OOS Fault	Rst Mate Frcd	Uavlbl	OOS Tbl Anal	OOS Removed
Protocol Fail	—	Active	—	Active Degraded	Active Degraded	—	OOS Fault	—	Uavlbl	Active Degraded	OOS Rmv Act
Switch Abort	—	Active	—	Active Degraded	Active Degraded	—	OOS Fault	—	Uavlbl	Active Degraded	—
Perf Deg Mate OOS	—	—	—	—	OOS Manual	OOS Removed	OOS Fault	—	Uavlbl	OOS Tbl Anal	—
Switch Abort Link Forced	—	—	—	—	—	—	—	—	Uavlbl	OOS Tbl Anal	—
Restore Mate Forced	—	—	—	—	—	—	—	Active Forced	—	—	—

TABLE C

DUPLEX LINKS DESIGNATE ACTION TABLE

STIMULUS	DUPLEX LINKS DESIGNATE ACTION TABLE										
	ACTIVE	STANDBY	ACTIVE DEGRADED	OOS DEGRADED	OOS MANUAL	OOS REMOVED	OOS FAULT	ACTIVE FORCED	UAVLBL	OOS TBL ANAL	OOS RMV ACTIVE
TTY Switch	Remove	Make Active	—	—	—	—	—	—	—	—	—
TTY Remove	Remove	Diagnose	Remove	—	—	—	—	Remove	—	—	—
TTY Force Uavlbl	Remove	—	Remove	—	—	—	—	Remove	—	—	Remove
Diagnostic	Remove	—	Remove	—	—	—	—	Remove	—	—	—
Carrier Fail	Remove Diagnose	Diagnose	Remove Diagnose	Diagnose	—	—	—	—	—	—	—
Excessive Error Rate	Remove Diagnose	—	Remove Diagnose	—	—	—	—	—	—	—	—
Circuit Failure	Remove Diagnose	Diagnose	Remove Diagnose	Diagnose	—	—	—	—	—	—	—
Protocol Fail	Remove Diagnose	—	Remove Diagnose	—	—	—	—	—	—	—	—
Switch Abort	Diagnose	—	Diagnose	—	—	—	—	—	—	—	—
Trouble Analysis	Remove Diagnose	—	Remove Diagnose	—	—	—	—	—	—	—	—
TTY Force Act	—	Make Active	—	Make Active	Make Active	Make Active	Make Active	—	Make Active	Make Active	Make Active
Diag ATP	—	Make Active	—	Make Active	—	Make Active	Make Active	—	—	—	Make Active
Diag STP	—	Make Active	—	Make Active	—	Make Active	Make Active	—	—	—	Make Active

TABLE C (Contd)

DUPLEX LINKS DESIGNATE ACTION TABLE

STIMULUS	DUPLEX LINKS DESIGNATE ACTION TABLE										
	ACTIVE	STANDBY	ACTIVE DEGRADED	OOS DEGRADED	OOS MANUAL	OOS REMOVED	OOS FAULT	ACTIVE FORCED	UAVLBL	OOS TBL ANAL	OOS RMV ACTIVE
Perf Deg Mate OOS	—	—	Remove Diagnose	—	—	—	—	—	—	—	—
TTY Restore	—	—	—	Diagnose	Diagnose	Diagnose	Diagnose	—	Diagnose	Diagnose	Diagnose
TTY Detain	—	—	—	—	—	—	—	—	—	—	Remove
Diag STF Hdw	—	—	—	—	—	—	—	—	—	—	Remove
Switch Abort Link Forced	—	—	—	—	—	—	—	Active Forced	—	—	—
Diag STF PD	—	—	—	—	—	—	—	—	—	—	Remove

TABLE D

DUPLEX LINKS MATE ACTION TABLE

STIMULUS	DUPLEX LINKS MATE ACTION TABLE									
	ACTIVE	STANDBY	ACTIVE DEGRADED	OOS DEGRADED	OOS MANUAL	OOS REMOVED	OOS FAULT	UAVLBL	OOS TBL ANAL	OOS RMV ACTIVE
TTY Switch	Remove	Make Active	—	Make Active	—	—	—	—	—	—
TTY Remove	—	Make Active	—	—	—	—	—	—	—	—
TTY Force Act	Remove	—	Remove	—	—	—	—	—	—	Remove
TTY Force Uavlbl	—	Make Active	—	Make Active	Make Active	—	—	—	Make Active	—
Diag	—	Make Active	—	Make Active	—	—	—	—	—	—
Diag ATP	Remove	—	Remove	—	—	—	—	—	—	Remove
Diag STP	—	—	Remove	—	—	—	—	—	—	Remove
Diag STF Hdw	—	—	—	—	Make Active	—	—	—	Make Active	—
Carrier Fail	—	Make Active	—	Make Active	Make Active	—	—	—	Make Active	—
Circuit Failure	—	Make Active	—	Make Active	Make Active	—	—	—	Make Active	—
Protocol Fail	—	Make Active	—	Make Active	Make Active	—	—	—	Make Active	—
Excessive Error Rate	—	Make Active	—	Make Active	—	—	—	—	—	—

TABLE D (Contd)

DUPLEX LINKS MATE ACTION TABLE

STIMULUS	DUPLEX LINKS MATE ACTION TABLE									
	ACTIVE	STANDBY	ACTIVE DEGRADED	OOS DEGRADED	OOS MANUAL	OOS REMOVED	OOS FAULT	UAVLBL	OOS TBL ANAL	OOS RMV ACTIVE
Switch Abort	—	Reinform	—	Reinform	Reinform	Remove	Remove	Remove	Reinform	—
Switch Abort Link Forced	—	Remove	—	Remove	Remove	Remove	Remove	Remove	Remove	—
Trouble Analysis	—	Make Active	—	Make Active	Make Active	—	—	—	—	—
Link Restore	—	—	—	—	—	—	—	—	—	Remove

231-050-027	Peripheral Unit Controller/Data Link Frame (Task Oriented Practice document)	231-090-062	Peripheral Unit Controller/Data Link Feature