Model 990 Computer
DX10 HDLC Communications Package
User's Guide

Part No. 2270526-9701 *A
1 October 1981

TEXAS INSTRUMENTS

MANUAL REVISION HISTORY

Model 990 Computer DX10 HDLC Communications Package User's Guide
(2270526-9701)

Original Issue . . . . 1 September 1980
Revision. . . . . 1 October 1981

The total number of pages in this publication is 274.

Preface

This manual describes the operation of the Texas Instruments DX10 HDLC Communications Package. The manual is primarily intended for systems personnel responsible for constructing the communications network and for applications programmers responsible for developing programs that utilize the communications package. The manual assumes that the reader is familiar with the DX10 and TX5 operating systems.

The protocols used by the communications package are based on, and are compatible with, the CCITT X.25 and high-level data link control (HDLC) recommendations. These protocols are transparent to users of the communications package.

This manual is organized into five sections and four appendixes as follows:

Section	
1	Introduction - Provides an overview of the communications package; describes the significant hardware and software features and possible network configurations; explains principles of system operation.
2	Planning the Network - Provides the background information necessary for constructing and operating the communications package; describes in detail the prerequisites for constructing the network.
3	Constructing and Activating the Network - Describes the procedures for constructing the communications package at the primary and secondary stations and for activating the network.
4	Applications Programming - Provides information necessary to write applications programs that use the communications package.
5	Applications Utilities - Describes the applications utilities available for generating, processing, downloading, and debugging programs that execute in secondary stations.

Appendix

A	Error Messages - Describes the error messages that may be encountered during activation and operation of the communications package.
B	Throughput - Provides detailed information on HDLC network polling and addressing techniques.
C	Generating a TX5 Operating System - Provides the information necessary to generate a TX5 operating system with communications capabilities.
D	Communication Structures - Provides information on transmission structures necessary to program secondary stations.

The following manuals contain information related to the DX10 HDLC Communications Package:

Title	Part Number
Model 990 Computer Four Channel Communications Controller Installation and Operation Manual	2263738-9701
Model 990 Computer 990/10 and 990/12 Assembly Language Reference Manual	2270509-9701
Model 990 Computer DX10 Operating System Reference Manual Volume 1 - Concepts and Facilities	946250-9701
Model 990 Computer DX10 Operating System Reference Manual Volume 2 - Production Operation	946250-9702
Model 990 Computer DX10 Operating System Reference Manual Volume 3 - Application Programming Guide	946250-9703
Model 990 Computer DX10 Operating System Reference Manual Volume 4 - Developement Operation	946250-9704
Model 990 Computer DX10 Operating System Reference Manual Volume 5 - System Programming Guide	946250-9705
Model 990 Computer DX10 Operating System Reference Manual Volume 6 - Error Reporting and Recovery	946250-9706

Model 990 Computer TX5 Operating System Programmer´s Guide	225035-9701
Model 990 Computer FORTRAN Programmer´s Reference Manual	946260-9701
Model 990 Computer TI Pascal User´s Manual	946290-9701
Model 990 Computer Local Line Module Operation and Maintenance Manual	2250676-9701

Contents

Paragraph Title Page

SECTION 1 Introduction

1.1 General 1-1
1.2 Network Configurations 1-1
1.3 Communications Package Features 1-4
1.3.1 Hardware Features 1-4
1.3.2 Software Features 1-5
1.4 Principles of System Operation 1-8
1.4.1 System Operation - General Description 1-8
1.4.1.1 Level 1 (Line Level) 1-10
1.4.1.2 Level 2 (HDLC Line Control) 1-12
1.4.1.3 Level 3 (Packet Level) 1-13
1.4.1.4 Level 4 (End-to-End Control) 1-13
1.4.1.5 User Task/Utilities Level 1-13
1.4.2 System Operation - Detailed Description 1-13
1.4.2.1 Level 1 (Line Level) 1-13
FCCC Board 1-15
LLM Board 1-15
1.4.2.2 Level 2 (HDLC Line Control) 1-16
1.4.2.3 Level 3 (Packet Level) 1-18
Reset Request (RESET) Packet 1-20
Reset Confirmation (RC) Packet 1-20
Receive Ready (RR) Packet 1-20
Receive Not Ready (RNR) Packet 1-20
1.4.2.4 Level 4 (User Interface) 1-20
Activation Services Requests 1-21
Network Information Tables (NITs) 1-21
1.4.2.5 Communications Package Utilities 1-22
Network Generator 1-23
Downloader 1-23
Download Preprocessor 1-24
Activate and Deactivate 1-24
Remote SCI 1-24
Statistics Accumulation 1-24

SECTION 2 Planning the Network

2.1 General 2-1
2.2 Network Addressing 2-1
2.2.1 Network Information Tables (NITs) 2-2
2.2.2 Planning the Network Assignments 2-2
2.2.2.1 Logical Unit Numbers (LUNOs) 2-3
2.2.2.2 Assigning Station Addresses 2-3
2.2.2.3 Assigning Logical Channel Identifiers (LCIs) . 2-4
2.2.2.4 Assigning Task IDs 2-4
2.2.2.5 Assigning Network IDs (NIDs) 2-4
2.2.3 Network Addressing Example 2-5
2.3 Documenting the Network Configuration 2-7
2.3.1 Documenting the Primary Station 2-7
2.3.2 Documenting the Secondary Stations 2-9

SECTION 3 Constructing and Activating the Network

3.1 General 3-1
3.2 DX10 Primary System 3-2
3.2.1 Building the Primary System 3-2
3.2.2 Installing the FCCC Board in the 990/10 3-4
3.3 Network Generation 3-5
3.3.1 Conventions and Examples 3-7
3.3.2 Activation and Termination 3-8
3.3.3 Generating a New Configuration 3-8
3.3.3.1 Generating a Non-TX5 Secondary Station 3-10
3.3.3.2 Generating a TX5 Secondary Station 3-12
3.3.3.3 Generating Additional Stations and Lines . . . 3-14
3.3.3.4 DX10 Task and Network ID Association 3-15
3.3.4 Modifying an Existing Network Configuration . . 3-16
3.3.4.1 Adding to an Existing Network Configuration . 3-16
Adding a Primary Task 3-17
Adding a Secondary Task 3-18
Adding a Secondary Station 3-19
Adding an FCCC Board 3-21
3.3.4.2 Changing an Existing Entry 3-22
Changing an FCCC Channel Entry 3-23
Changing a Queue Buffer Entry 3-24
Changing a Network Entry 3-25
Changing a Packet Time-Out 3-27
3.3.4.3 Deleting from an Existing Configuration . . . 3-28
3.4 TX5 Secondary System 3-29
3.4.1 Building the Secondary System 3-29
3.4.2 Building Remote SCI for a TX5 Secondary Station 3-30
3.4.3 Installing the LLM Board in the 990/5 Computer . 3-33
3.4.4 Assembling the TX5 Files 3-35
3.4.5 TX5 System Generation 3-36
3.5 Activating the Communications Network 3-37
3.5.1 Activating the Primary Station Communications Software 3-38
3.5.2 Downloading the Preprocessed Files 3-39

SECTION 4 Applications Programming

4.1 General 4-1
4.2 Supervisor Calls 4-2
4.2.1 Input/Output Calls 4-3
4.2.1.1 Input/Output Call Block 4-3
4.2.1.2 Write/Send Data - Opcode 2 4-7
4.2.1.3 Read Input Data - Opcode 3 4-8
4.2.2 Activation Services Calls 4-8
4.2.2.1 Activation Services Call Block 4-9
4.2.2.2 Request Activation from Suspension - Opcode 4 4-11
4.2.2.3 Request Activation from a Time Delay - Opcode 5 4-12
4.2.2.4 Request Activation from Termination - Opcode 6 4-12
4.2.2.5 Request Removal from Activation Services - Opcode 7 4-12
4.3 Programming Examples and Applications Notes . . . 4-13
4.4 Managing the Flow of Communications 4-16
4.5 Assembly Language Programming Examples 4-17
4.6 FORTRAN and Pascal Programming Examples 4-22
4.6.1 Assembly Language Subroutine for FORTRAN Interface 4-22
4.6.2 Subroutine Calls 4-27
4.6.2.1 READ4D Subroutine Call 4-27
4.6.2.2 WRIT4D Subroutine Call 4-29
4.6.2.3 WKUP4D Subroutine Call 4-32
4.6.3 FORTRAN Programming Examples 4-34

SECTION 5 Applications Utilities

5.1 General 5-1
5.2 Download Preprocessor Utility 5-2
5.2.1 Preprocessor Background Information. 5-3
5.2.2 Preprocessor Utility Execution 5-4
5.2.3 Preprocessor Command Prompts and Responses . . . 5-5
5.2.3.1 Preprocessor Alias Commands 5-6
Add (Assign) an Alias 5-6
Modify a Secondary Station Alias 5-7
Delete a Secondary Station Alias 5-7
5.2.3.2 Preprocessor File Operations 5-8
Build a Preprocessed Download File 5-8
Delete a Preprocessed Download File 5-9
List Download Files and Aliases 5-10
Modify (Patch) a Preprocessed File 5-10
Dump the Contents of a Preprocessed File . . 5-13
5.3 Downloader Utility 5-14
5.3.1 Downloader Utility Background Information . . . 5-14
5.3.2 Downloader Utility Execution 5-15
5.3.3 Downloader Prompts and Responses 5-15
5.3.3.1 Downloader Error and Procedural Messages . . . 5-16
5.3.3.2 Downloader Error Recovery 5-16
5.4 Remote SCI Utilities 5-19

5.4.1 Remote SCI Utility Activation and Termination . 5-20
5.4.2 Remote SCI Command Menus 5-21
5.4.2.1 Debug Command Menu 5-21
5.4.2.2 Task Command Menu 5-23
5.4.2.3 LUNO Command Menu 5-24
5.4.2.4 Miscellaneous Command Menu 5-24
5.4.3 Remote SCI Commands 5-25
5.4.3.1 Assign Breakpoint (AB) Command 5-28
5.4.3.2 Assign LUNO (AL) Command 5-29
5.4.3.3 Delete Breakpoint (DB) Command 5-30
5.4.3.4 Dump Workspace (DW) Command 5-30
5.4.3.5 Display Error Codes (ERRS) Command 5-31
5.4.3.6 Initialize Date and Time (IDT) Command 5-31
5.4.3.7 Install Task (IT) Command 5-32
5.4.3.8 Kill Task (KT) Command 5-33
5.4.3.9 List Breakpoints (LB) Command 5-33
5.4.3.10 List Memory (LM) Command 5-34
5.4.3.11 List System Memory (LSM) Command 5-35
5.4.3.12 Modify Memory (MM) Command 5-36
5.4.3.13 Modify System Memory (MSM) Command 5-37
5.4.3.14 Associate Network ID/Task ID (NID) Command . . 5-38
5.4.3.15 Read CRU (RCRU) Command 5-39
5.4.3.16 Release LUNO (RL) Command 5-39
5.4.3.17 Show Date and Time (SDT) Command 5-40
5.4.3.18 Show Input/Output Status (SIS) Command 5-40
5.4.3.19 Show Task Status (STS) Command 5-41
5.4.3.20 Show Value (SV) Command 5-41
5.4.3.21 Trace (TR) Command 5-42
5.4.3.22 Write CRU (WCRU) Command 5-43
5.4.3.23 Execute Task (XT) Command 5-44
5.4.4 Remote SCI Error Reporting and Recovery 5-45
5.5 Statistics Package 5-60
5.5.1 Network Statistics 5-62
5.5.1.1 Network Status (NS) Command 5-62
5.5.1.2 Network Traffic (NT) Command 5-64
5.5.1.3 Network Buffers (NB) Command 5-66
5.5.1.4 Network Queues (NQ) Command 5-67
5.5.1.5 Network Errors (NE) Command 5-68
5.5.1.6 Number of Network Messages (NM) Command . . . 5-69
5.5.1.7 Network Restart (NR) Command 5-70
5.5.2 Station Statistics 5-70
5.5.2.1 Status of a Station (SS) Command 5-70
5.5.2.2 Station Errors Reported (SE) Command 5-72
5.5.2.3 Station Message Totals (SM) Command 5-73
5.5.2.4 Station Reset (SR) Command 5-74
5.5.3 Line Statistics 5-74
5.5.3.1 Line Status (LL) Command 5-74
5.5.3.2 Status of a Station at Line Level (LS) Command 5-75
5.5.3.3 Line/Station Reset (LR) Command 5-76
5.5.4 Artificial Data 5-77
5.6 Activate/Deactivate Utilities 5-81

Appendixes

Appendix Title Page

A Error Messages A-1

B Throughput B-1

C Generating a TX5 Operating System C-1

D Communications Structures D-1

Glossary

Index

Tables

Table Title Page

1-1 HDLC Level Functions 1-16
3-1 HDLC Directories and Files 3-6
5-1 Download Messages and Error Recovery 5-17
5-2 Summary of Remote SCI Commands and Prompts 5-27
5-3 Remote SCI Messages and Error Recovery 5-46

Illustrations

Figure Title Page

1-1 Network Configuration Possibilities 1-3
1-2 Components of the HDLC Communications Package . . . 1-7
1-3 Level Diagram of HDLC Communications System 1-9
1-4 Information Format under the HDLC Communications Package 1-11
1-5 Typical System Configuration 1-14
1-6 HDLC Network Polling Sequence 1-17
1-7 Level 3 Packet-Level Block Diagram 1-19
1-8 Level 4 User Interface 1-21
2-1 Network Addressing on the Same Multipoint Line . . . 2-6
2-2 Primary Station Documentation 2-8
2-3 Secondary Station Documentation 2-9
4-1 Supervisor Call Block for Input/Output Calls 4-4
4-2 Supervisor Call Block for Requesting Activation Services 4-10
4-3 Data Flow within the Communications Package 4-15
4-4 Managing the Flow of Communications 4-16
4-5 Primary Station Assembly Language Program, ... 4-17
4-6 Secondary Station Assembly Language Program 4-20
4-7 Assembly Language Interface 4-23
4-8 FORTRAN Program in Primary Station 4-35
4-9 FORTRAN Program in Secondary Station 4-36

Section 1

Introduction

1.1 GENERAL

This section presents an overview of the Texas Instruments DX10 HDLC Communications Package. This package provides control of a network consisting of two or more Model 990 Computer systems connected by a communications line. The communications package supports communications between applications programs resident in physically separate and independent computers within the network. Typical communications package applications include distributed processing and the control of industrial processes.

The following paragraphs describe the network configurations supported by the communications package, features of the package, and general principles of system operation.

1.2 NETWORK CONFIGURATIONS

A minimum network configuration consists of two systems as follows:

* A primary (host) station. This must be a Model 990/10 Computer or Model 990/12 Computer system equipped with:
    - A DX10 operating system, release 3.4
    - The communications package software
    - A four-channel communications controller (FCCC) board
    - A Model 911 Video Display Terminal (VDT)
    - 192 kilobytes of memory

- * A secondary station. This may be either a Model 990/5 Computer system or a remote processing device, such as a TM990 or PM550 system. If the secondary station is a Model 990/5 Computer system, it must be equipped with the following:
    - A TX5 operating system, release 3.2.0, with communications package software
    - A local-line module (LLM), which functions as the communications controller
    - A DX10 HDLC Communications Package ROM loader to provide downloading capabilities
- * A communications line that interconnects the computers within the network. This can be one of the following:
    - A local-line cable consisting of a single-shielded, twisted-pair cable that is a maximum of 3000 meters in length
    - A modem line in accordance with EIA specification RS-232C.

A configuration can include a maximum of 32 secondary stations connected to the primary station on a single multipoint line, as shown in Figure 1-1. The secondary stations can be any combination of TX5 supported systems or remote devices.

The physical interface between the primary and secondary stations is provided by the FCCC board in the primary station interconnected through the communications line to the LLMs or other type of communications controller at the secondary stations. Features of the FCCC board and the LLM are described in subsequent paragraphs of this section.

Control of the network is provided by the primary station. Secondary stations can send data only to the primary station. Each secondary station is polled on a time-selected basis. A secondary station must wait until it is polled by the primary before sending any data.

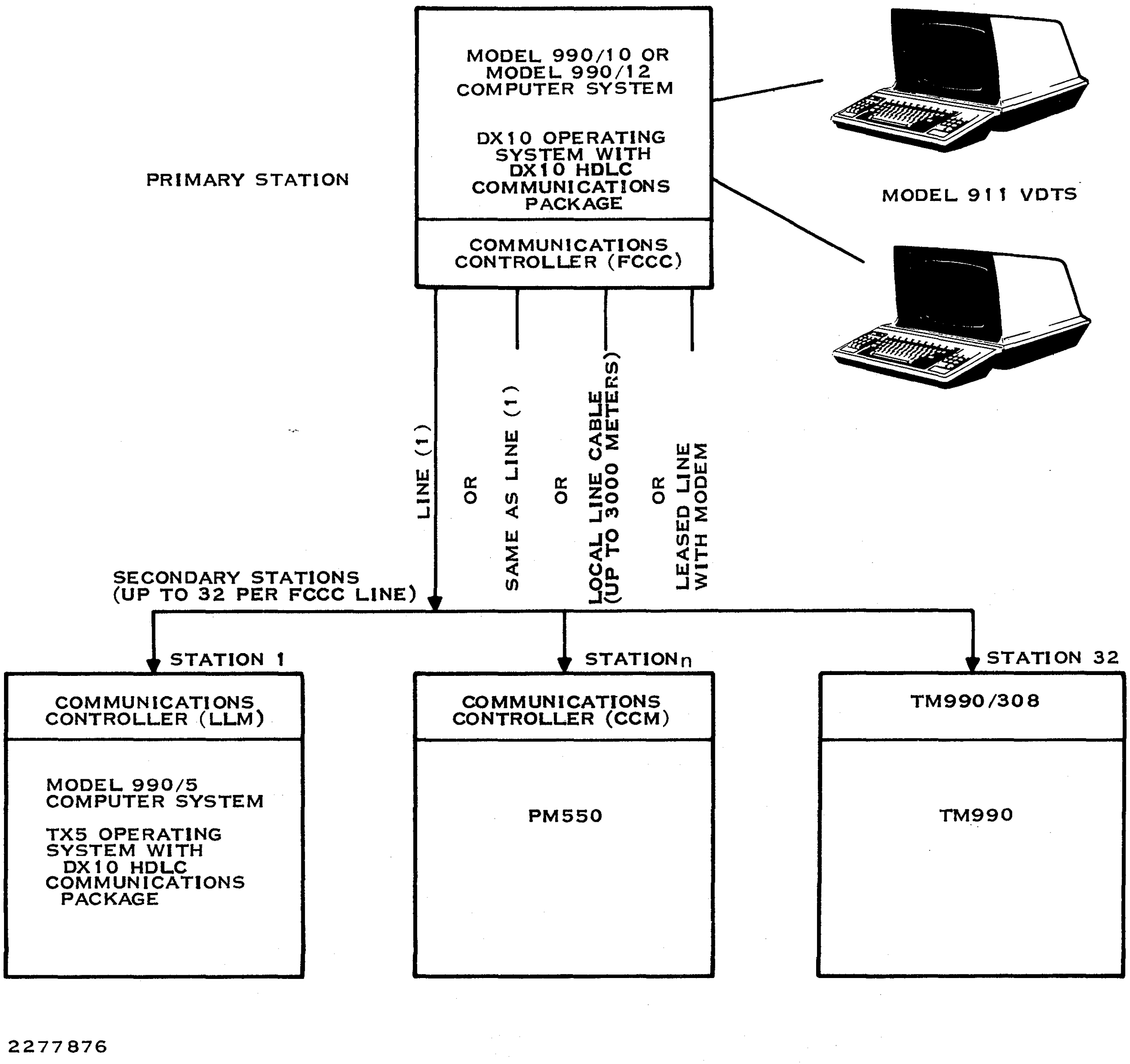


Figure 1-1 Network Configuration Possibilities

1.3 COMMUNICATIONS PACKAGE FEATURES

The communications package supports the following:

- Communications between applications programs installed in the primary station and those installed in secondary stations.
- Downloading of secondary stations from the primary station. For TX5 secondary stations, the Model 990/5 Computer must be equipped with an DX10 HDLC Communications Package ROM loader.
- Remote System Command Interpreter (SCI) capabilities for installing applications programs in TX5 secondary stations from the primary station and debugging these programs from a terminal at the primary station.

In order to perform these functions, the communications package includes hardware and software additions to both the primary station and TX secondary stations. These additions are described in the following paragraphs.

1.3.1 Hardware Features

The hardware additions required by the communications package include the FCCC board for the primary station and an LLM for each TX5 secondary station in the network. The FCCC board and LLMs provide the following:

- Bit-oriented, half-duplex, serial synchronous operation that can be a maximum of 9600 bits per second
- Multipoint operation
- Eight-bit character length
- Switch-selectable addresses (on LLM)
- Line termination selection (on LLM)
- Line isolation at a maximum of 1300 volts dc
- Data transparency
- 16-bit cyclic redundancy check (CRC-CCITT) to detect line errors

The FCCC board occupies a full slot in the primary computer chassis and communicates with the central processing unit (CPU) through the TILINE* interface. The LLM occupies a half-slot in the secondary computer chassis and interfaces with the CPU through the communications register unit (CRU). The FCCC board and LLMs contain switches and jumpers that must be set correctly before communication is attempted. These are described in subsequent sections of this manual. Additional information on these boards is provided in the appropriate installation and operation manual listed in the Preface.

1.3.2 Software Features

The communications package software includes additions to the operating systems of the primary station and the secondary stations. The software components that comprise the communications package interface with the respective operating systems to provide communications services to stations in the network. The software components within the communications package include the following:

* Network generation and start-up utilities executed by the systems personnel responsible for constructing the network. These programs and utilities are activated at the primary station by SCI commands as described in Section 3. These utilities include the following:

  - A network generator utility to define the software configuration of the network and create the data structures (network information tables) required for task-to-task communications.

  - A communications package generator utility to link the primary communications software to the network information tables.

  - A remote SCI generator utility to build the link control file for secondary remote SCI tasks. One must be built for each TX5 secondary station that will use remote SCI.

  - A download preprocessor utility for formatting the secondary stations´ operating systems and stand-alone programs prior to transmitting them to the secondary stations.

  - A network activator utility for initializing communications between the primary and secondary stations.

---

* Trademark of Texas Instruments Incorporated.

* Applications utilities that may be executed by users of the network as described in Section 5. These utilities include the following:

  - A downloader utility for transmitting preprocessed files to specified secondary stations.

  - A remote SCI utility (analogous to DX10 SCI) to aid in debugging tasks resident in TX5 secondary stations.

  - A statistics utility to monitor network activity.

  - Activate/deactivate utilities to provide control over communications activities between primary and secondary stations.

* A supervisor call (SVC) processor that provides the interface between the communications package and user applications programs that use the network. The communications SVC processor accesses extended operations (XOPs) at level 15 with an opcode of >4D. Calls to the SVC processor can be made from assembly language, FORTRAN, or Pascal programs to perform the following functions:

  - Write (transmit) information to tasks in other stations.

  - Read information received from another station.

  - Request activation (wake-up) services from the communications package.

* Communications tasks attached to a procedure that perform the following:

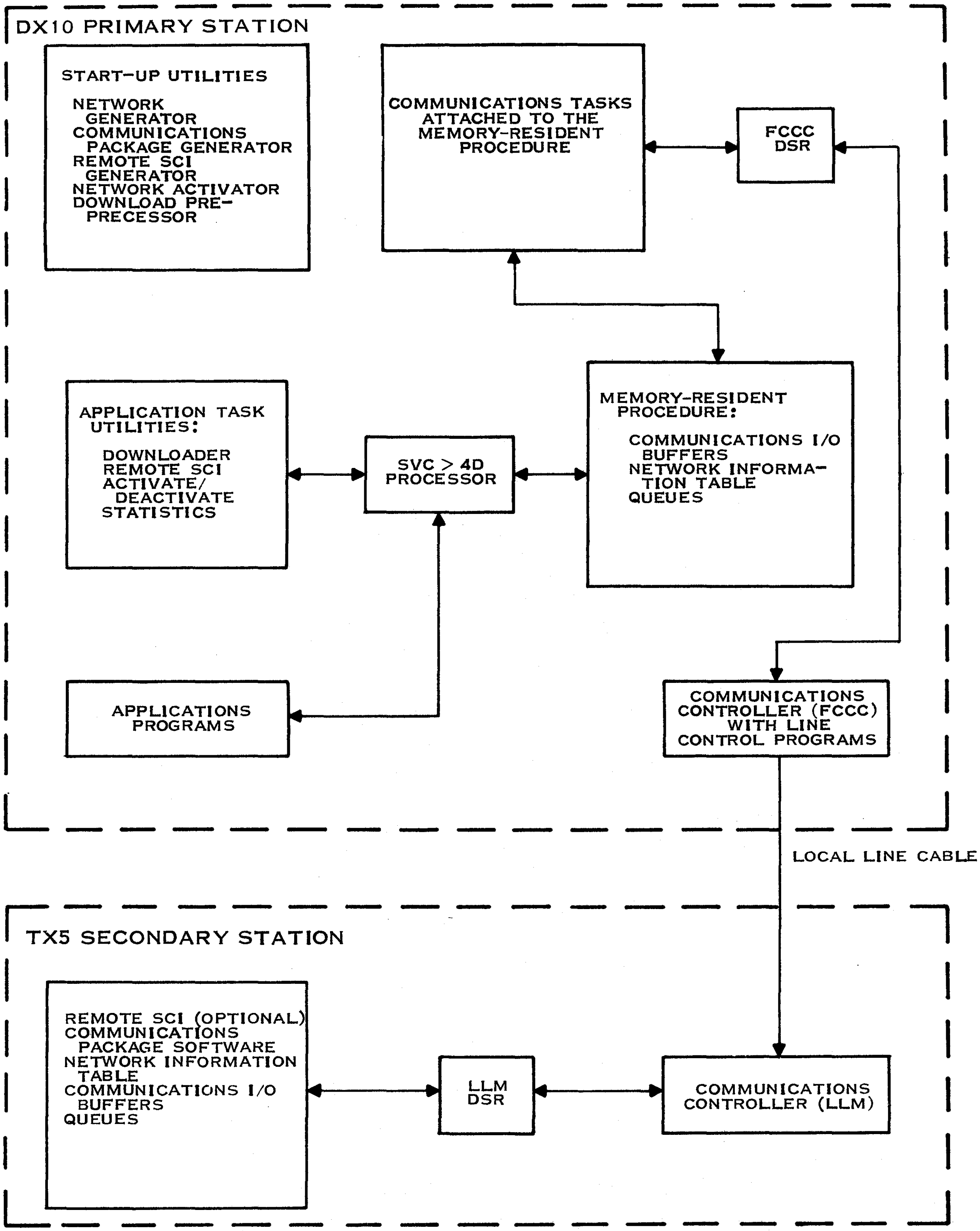
  - Handle all input/output (receive/transmit) operations using the communications protocol.

  - Control the flow of information that is initiated by polling secondary stations.

  - Report errors that occur during the operation of the communications package.

* Device service routines (DSRs) that provide the interface between the FCCC board and the DX10 communications package software and between the LLM board and the TX5 communications package

Each of these features is represented in the system diagram in Figure 1-2.



2277877

Figure 1-2 Components of the HDLC Communications Package

1.4 PRINCIPLES OF SYSTEM OPERATION

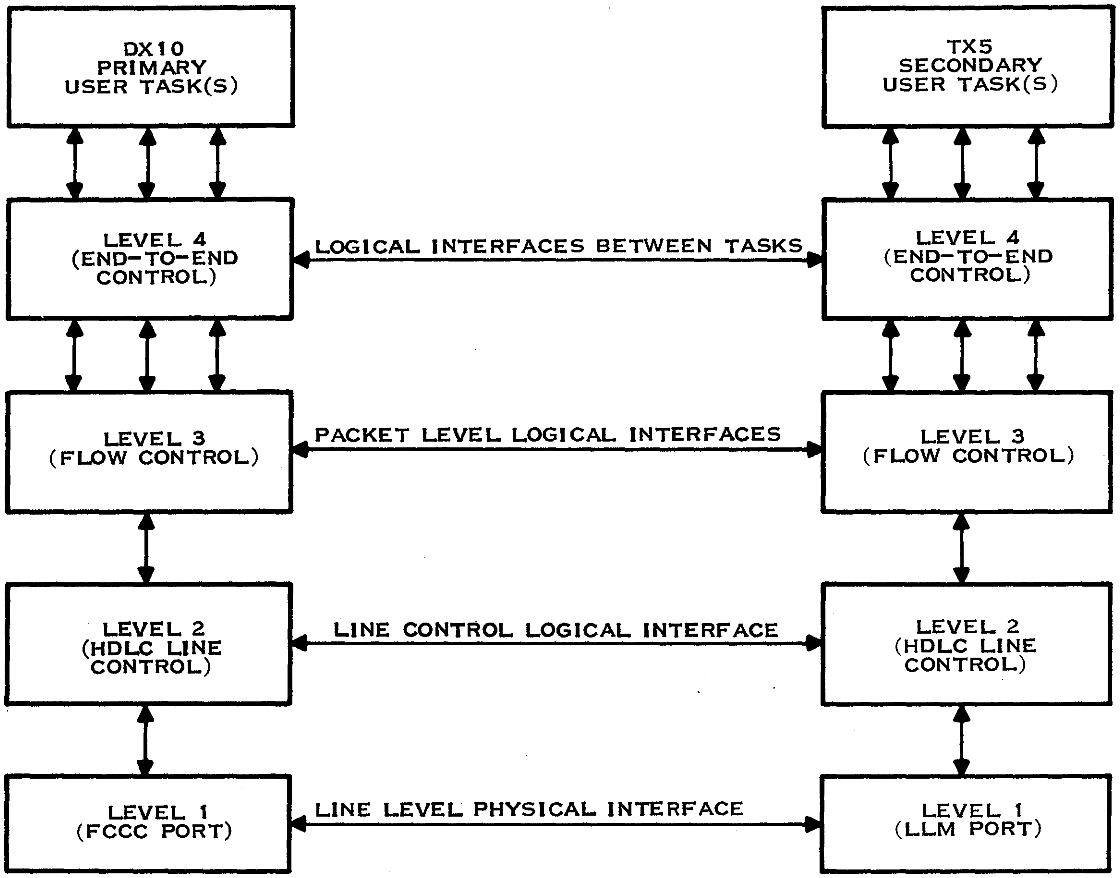
The following paragraphs describe the basic operation of the communications package. The software makes system operation largely user-tranparent. The information provided in the following paragraphs enables the user to make the most effective use of the communications package.

An overview of basic system operation is given first, followed by a more detailed description.

1.4.1 System Operation - General Description

The structure of the communications package is compatible with the lower levels of the International Standards Organization (ISO) Open System Interconnect model. Figure 1-3 shows the various levels of software that are implemented in the package at a DX10 primary and a TX5 secondary station.

A primary-resident task that communicates with a secondary-resident task does so via the intervening software levels. Data generated by the primary task for transmission to the secondary station is passed downward from level 4 to level 1, where it is transmitted to the secondary station over a physical communications line (data link). At the secondary station, the reverse process takes place and the data is passed upward to the secondary task. The reverse process occurs when the secondary station transmits to the primary station. Due to the transparency of the communications software, primary and secondary tasks appear to communicate directly with each other. A logical interface may be said to exist between the tasks, in contrast to the actual physical interface that exists between the primary and secondary stations. In the same manner, logical interfaces exist between primary and secondary stations at the packet level and the line-control level.



2280117

Figure 1-3 Level Diagram of HDLC Communications System

The software levels are as follows:

- Level 1 (line level). This is the actual physical communications line connection between a primary station and the associated secondary stations.

- Level 2 (line-control level). This area of the software controls the physical line in accordance with the High Level Data Link Control (HDLC) protocol.

- Level 3 (packet level). This software area controls and interleaves the flow of information to and from primary-resident tasks and the various secondary-resident tasks.

- Level 4 (end-to-end control). This is the interface between the tasks and the lower levels.

Figure 1-4 shows how the actual transmitted data block (frame) is built as it is passed from one level to another prior to transmission across the physical interface. The data or message generated by the primary-resident task is processed at level 4 and queued for the packet level. A header is added to the message at the packet level for identification and control purposes and the combined result is passed to the HDLC line-control level. At the line-control level, transmission flags, addresses, and commands are added, along with a cyclic redundancy (CRC) check that is generated for error detection purposes. The result, an HDLC frame, is then in a form suitable for transmission to a secondary station (via the data link), where the reverse process occurs. The data or message is stripped of all additions made at the primary station and is passed to the secondary application task for processing.

1.4.1.1 Level 1 (Line Level).

The physical interface (communications line) between the FCCC board in the primary station and the LLM board in a secondary station (diagrammed in Figure 1-1) can be one of the following:

- A differential local-line connection for distances of less than 3000 meters

- A connection via modems conforming to specification EIA RS-232-C

- A connection via modems conforming to specification EIA RS-423

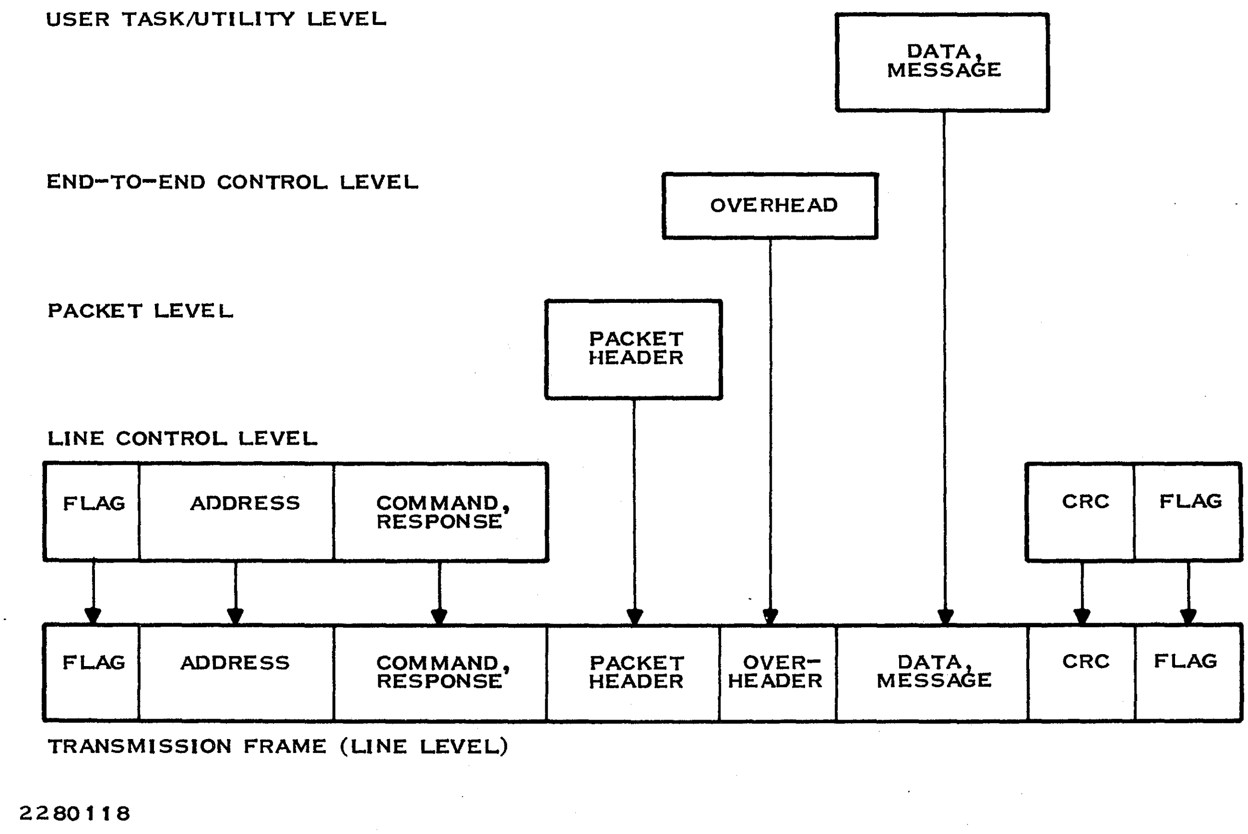


Figure 1-4 Information Format under the HDLC Communications Package

1.4.1.2 Level 2 (HDLC Line Control).

Level 2 is variously called the HDLC frame level, and the link or line-control level. It controls the communication line according to the HDLC protocol. In the primary station, it polls the secondary stations and sequences data transmissions. It also looks for and recognizes addresses received from the secondary stations with which it is communicating. Polling is the process of checking to see if any stations in the network have data to transmit. Polling is used by the primary station to monitor the status of the secondary stations in the network in addition to otherwise controlling the flow of information. At the secondary stations, as well as the primary, level 2 software performs all necessary verification of the data flowing between the stations. If an error is detected, the line-control software automatically initiates a retransmission of the data.

In the primary station, the line-control software is resident partially on the FCCC board and partially in main memory. In the secondary, most of the software is in main memory and very little is resident on the LLM board.

Four ports are available at the primary station (on the FCCC board) for the installation of lines to secondary stations in the network. Each line can be used as a multipoint line for connecting as many as 32 secondary stations into the network. Each secondary station has a unique address, which is embedded in all transmitted frames (see Figure 1-4) destined for that station. The secondary software recognizes its own address and proceeds to interpret and process the message contained in the frame.

1.4.1.3 Level 3 (Packet Level).

The primary packet-level software, although appearing logically to communicate directly with the secondary packet-level software, actually queues the task-generated data and messages to the line-control level, which in turn queues them to level 1. These messages are referred to as data "packets." If multiple HDLC communications tasks are executing simultaneously in the system, the packet level at the transmitting station interleaves (multiplexes) and queues the data packets passed from the various tasks into one data stream for level 2. Conversely, the packet level demultiplexes received messages for passing to the individual tasks.

As a consequence of multiplexing and queuing, a packet-level message to another station may not receive a response until after several other packet-level messages have been sent. The packet-level software writes a received message into an appropriate buffer and it is the responsibility of the receiving task to read that message.

1.4.1.4 Level 4 (End-to-End Control).

The routing of a message to the appropriate task is controlled by assigning identification (ID) numbers to each task and inserting these IDs in a network information table (NIT). The NIT also relates each task to the appropriate stations in the network.

Because direct communication between primary and secondary user tasks does not actually exist and only appears to exist because of the transparency of the software, a logical interface between the tasks is said to exist. The logical interface is termed a virtual circuit. It is the level 3 packet-level software used in the DX10 HDLC Communications Package that establishes permanent virtual circuits, that is, the permanent logical path or connection between specified tasks.

User tasks in the DX10 primary station and TX5 secondary stations access the communications package using supervisor calls (SVCs). SVCs are an implementation of the 990 computer assembly language extended operation (XOP). In the communications package, the level 15 XOP, normally reserved for system operation, is also used for the communications package user interface. The SVC >4D is the particular XOP 15 opcode used for communications. The SVC >4D processor interprets the call to determine the particular communications function requested.

1.4.1.5 User Task/Utilities Level.

The user applications tasks and the utilities available with the communications package are the highest level of operation under the operating system. All tasks, and some utilities, access the communications facilities via SVC calls. Refer to paragraphs 1.4.2.4 and 1.4.2.5 for further descriptions of SVC processing and the utilities, respectively.

1.4.2 System Operation - Detailed Description

The following paragraphs describe the operation of the communications package in greater detail.

1.4.2.1 Level 1 (Line Level).

An example configuration at the line level is illustrated in Figure 1-5. An FCCC board at the primary station is shown with two of its four ports connected to various types of secondary stations. The type of communications controller used at a secondary station is a function of the type of secondary station. Examples are as follows:

* TX5 station - uses an LLM

* PM550 station - uses a communications controller module (CCM)

* TM990 station - uses a TM990/308 controller board

The following paragraphs discuss the FCCC board for the DX10 primary station and the LLM for a TX5 secondary station. For information on other types of secondary stations and communications controllers, refer to the appropriate installation and operation manuals.

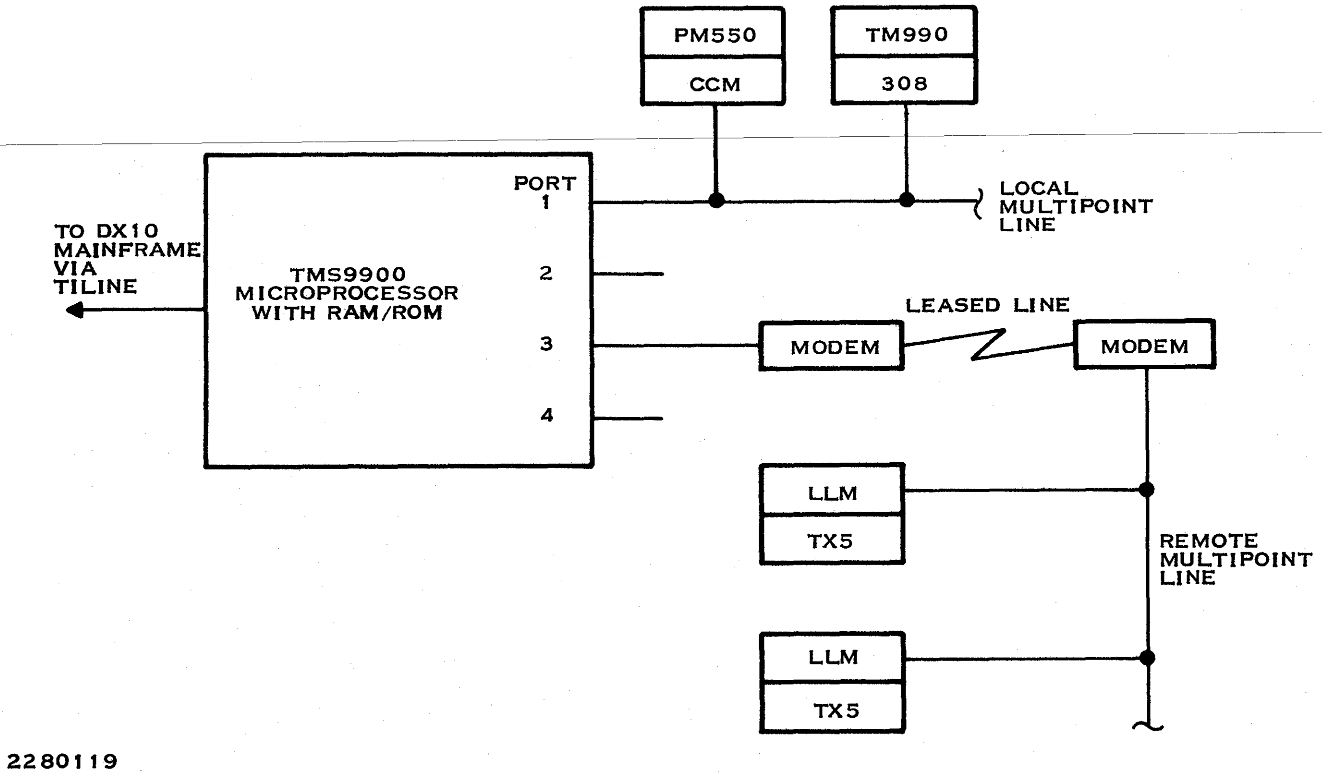


Figure 1-5 Typical System Configuration

FCCC Board.

Master control of the FCCC board, and consequently of level 1 communications, is provided by a TMS 9900 microprocessor. Both RAM and ROM are with the TMS 9900 and serve both as a source of commands for the four TMS 9903 communications controllers on the board and as storage for part of the HDLC software. The communications controllers take the data to be transmitted and convert it to the serial bit streams necessary for feeding the physical communications lines.

Drivers associated with each of the four TMS 9903s provide the appropriate line signals to the corresponding communications channel connector located on the outer edge of the FCCC board. The communications lines connect to the four communications channel connectors. Selection of either the EIA RS-232-C (or 423) or differential drive option is accomplished by reconnecting to the proper pins on the connectors. No software changes or FCCC wiring changes are required to change a line from the differential mode to an EIA mode.

Several presetting operations, including the allocation of a TILINE address, are required on the FCCC board to allow operation as a primary station in an HDLC network. Refer to the FCCC Installation and Operation Manual for complete instructions on all FCCC options that can be preset.

LLM Board.

Different types of secondary stations can be used in addition to the TX5, as shown in Figure 1-5. However, this manual concerns itself mainly with the TX5 secondary station equipped with the local-line module (LLM). For a discussion of the operation of other types of secondary stations, refer to the appropriate manual.

The LLM provides only one communication line connection and uses the TMS 9980 microprocessor for control, as well as a small amount of ROM and RAM. Compared to the FCCC board, the LLM is considerably more dependent on main memory software support and connects to the computer's CRU instead of the TILINE bus. However, the LLM contains the same drivers and receivers as the FCCC board, and it therefore supports the same local-line interface. A modem option is also available.

Two switch settings must be made on the LLM board for it to function correctly with the communications system. One is the data rate switch; the other is the address switch. The address switch must be set to the secondary address of the TX5 secondary station. Jumper connections must also be made. Refer to Section 3 of this manual and to the Local Line Module Operation and Maintenance Manual for further information on required settings.

1.4.2.2 Level 2 (HDLC Line Control).

Because secondary station requirements are less severe than those for the primary station, the secondary HDLC-level software is less complex than the primary HDLC-level software. Consequently, not all the functions of the primary and secondary software at this level match exactly. Table 1-1 lists the functions performed at the HDLC level at both the primary and secondary stations.

Table 1-1 HDLC Level Functions

Primary Station	Secondary Station
Polling	Responding
Error checking	Error checking
Addresses out	Addresses in
Addresses in	Command recognition
Command generation	Initialization recognition
Response recognition	Response generation
Sequence counts	Sequence counts

In conformance with HDLC protocol for multipoint operations, the HDLC level at the primary station builds frames to enclose data destined for secondary stations in the network. The primary station maintains network control with these frames by polling the individual secondary stations in a predetermined sequence, as described later in this paragraph. Figure 1-4 shows the following components of the transmitted frame:

* A flag byte for start-of-frame

* The destination secondary station address

* A command/response component

* A packet header

* A level 4 overhead component

* The user task data or message

* A cyclic redundancy check (CRC) code for error checking

* Another flag byte for end-of-frame

Figure 1-6 shows an example of a polling sequence that may be implemented in an HDLC network.

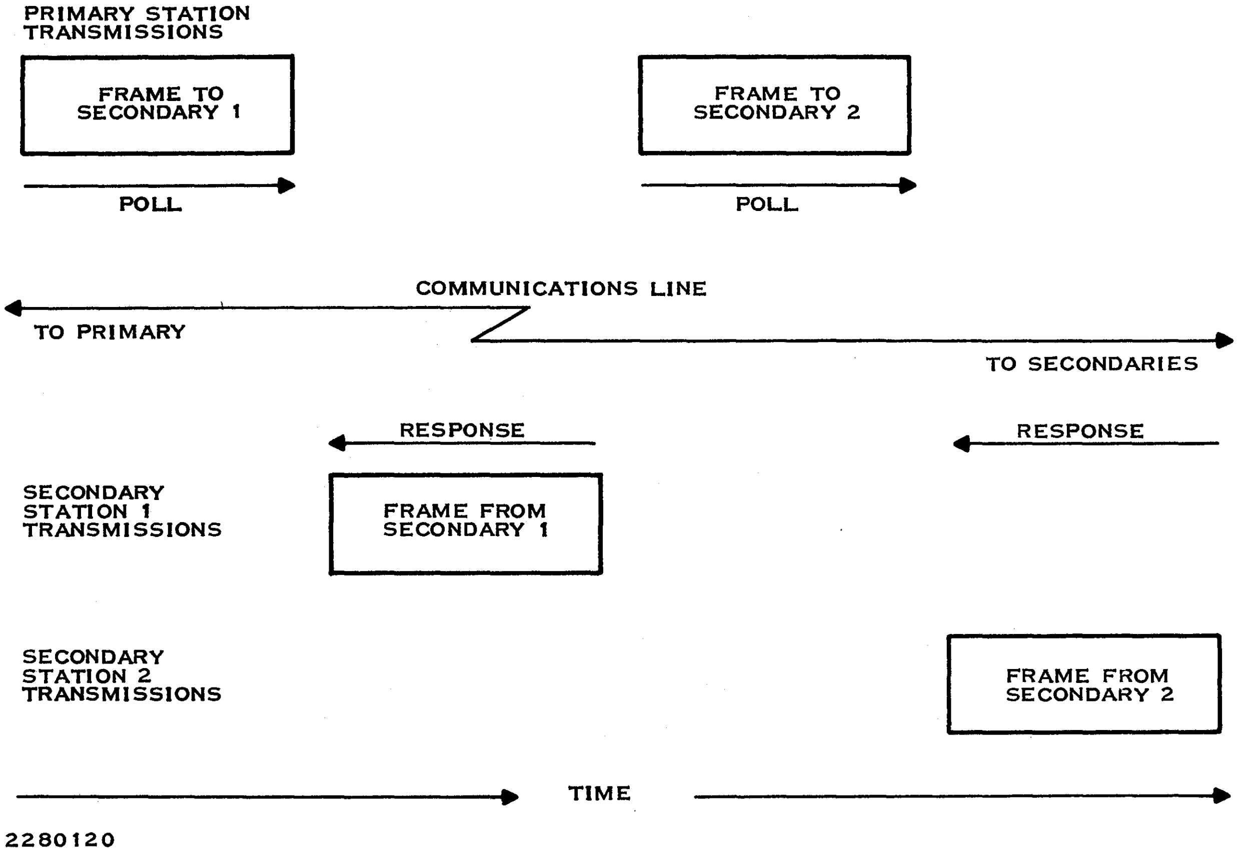


Figure 1-6 HDLC Network Polling Sequence

The HDLC polling software operates in a handshaking mode, that is, a response is expected to each frame transmitted by the primary station. (This requisite does not necessarily apply at the higher levels.) The primary station, therefore, acts as a central controller for all operations at the HDLC level. The primary station maintains a sequence count of the number of messages that remain unanswered by a given secondary station at a given time and relates this to the maximum number allowed for that secondary. This maximum number is termed a "window size". It varies according to the type of secondary station and the software level. Consider the following examples:

* Level 3. Due to the relative sophistication of the operating system, a TX5 secondary station can have a maximum of seven messages outstanding; that is, the window size of the station is seven. On the other hand, a PM550 with no operating system has a window size of one.

* Level 2 (HDLC level). All the window sizes are one because the HDLC software expects a response to every message sent to a secondary station.

The communications system uses a polling method called "data priority polling with delay select." When the network is generated (see Section 3), the user may specify a delay of 250 milliseconds to 8000 seconds. This delay is the maximum time that may be allowed to elapse between individual polls to a given secondary station during a period of low communication activity between the primary and that secondary. When a primary task passes a message to the network, the priority of that secondary station rises so that the time between polls is decreased to the minimum possible in the prevailing situation. If no further activity from that secondary station is seen, its priority drops and the time between polls increases. After a specified number of polls with no activity, the secondary station is polled in accordance with the time established during network generation.

1.4.2.3 Level 3 (Packet Level).

Figure 1-7 illustrates the flow of data at the packet level. For a transmission, the packet level generates a packet header, attaches it to the message passed from a user task and then forwards the combination to the next lower level (HDLC level). For a reception, the packet level removes the header from the data received from the lower level and forwards the message to the appropriate task buffer. The packet level also maintains records of the sending and receiving addresses associated with each task message.

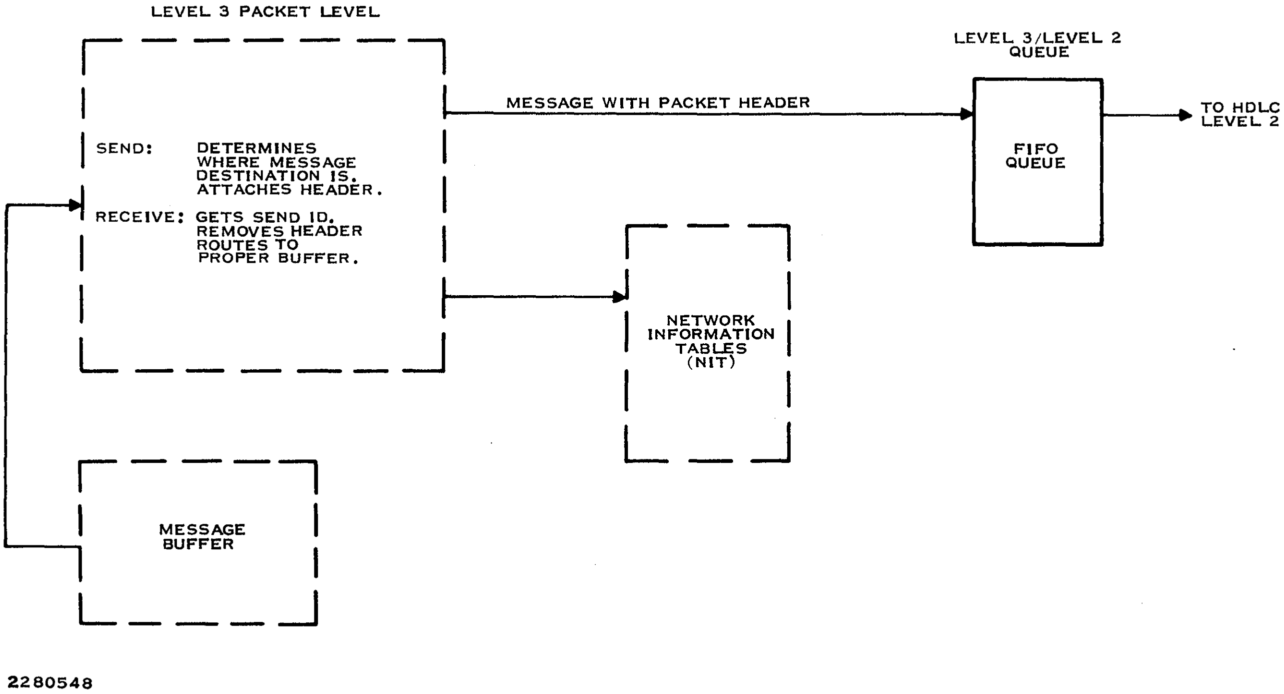


Figure 1-7 Level 3 Packet-Level Block Diagram

Message IDs are stored in the network information tables (NITs), which are further discussed under level 4 (paragraph 1.4.2.4). For the packet and higher levels, the NIT functions as a look-up table to determine message routing. For a transmission, the packet level software extracts data from the NIT for inclusion in the packet header. The combined message and header is then forwarded to the HDLC level via a first-in first-out (FIFO) queue.

Window sizes at the packet level were discussed in the preceding paragraph. At this level, the window size is largely a function of the degree of sophistication of the operating system; for example, a TX5 secondary has a window size of 7 while a PM550 or a TM990 has a window size of only 1. Any attempt to transmit a message when the corresponding window is full results in an error situation from which the system attempts to recover.

A packet that contains data or a message destined for another task is termed a data packet, and the associated packet header contains relevant address and control information, including a sequence count. The sequence count is provided as a means of countering the loss or duplication of messages between stations.

For certain control functions, control and address information is transmitted without any accompanying message and independently of a task. These control packets are described in the following paragraphs.

Reset Request (RESET) Packet.

This packet requests that communications be restarted between the primary station and a specified secondary station after network errors have occurred.

Reset Confirmation (RC) Packet.

This packet is sent to acknowledge the receipt of a RESET packet and to confirm that the reset procedures have been performed.

Receive Ready (RR) Packet.

This packet is sent to indicate that the transmitting station is ready to receive another data packet. This occurs after the station window fills and acknowledges any received data packets.

Receive Not Ready (RNR) Packet.

This is sent to indicate that the transmitting station cannot receive more messages. The RNR and RR packets are used in conjunction to regulate the flow of data packets through the network.

1.4.2.4 Level 4 (User Interface).

Level 4 provides the user with an interface to the HDLC communications system. A task may access the HDLC communications system using a supervisor call, SVC >4D. When a task performs a read or write operation (that is, when it receives or transmits), the associated message is passed to or from a DX10 communications buffer via this SVC. After the operation has been attempted, the SVC returns control to the task, along with a status code indicating the successful (or unsuccessful) completion of the operation. Operation codes 2 and 3 are used with the SVC to request read and write operations, respectively (discussed further in Section 4). Figure 1-8 shows the flow of information at the user interface level, including the relationship of activation services and the NIT to this level. These areas are described in the following paragraphs.

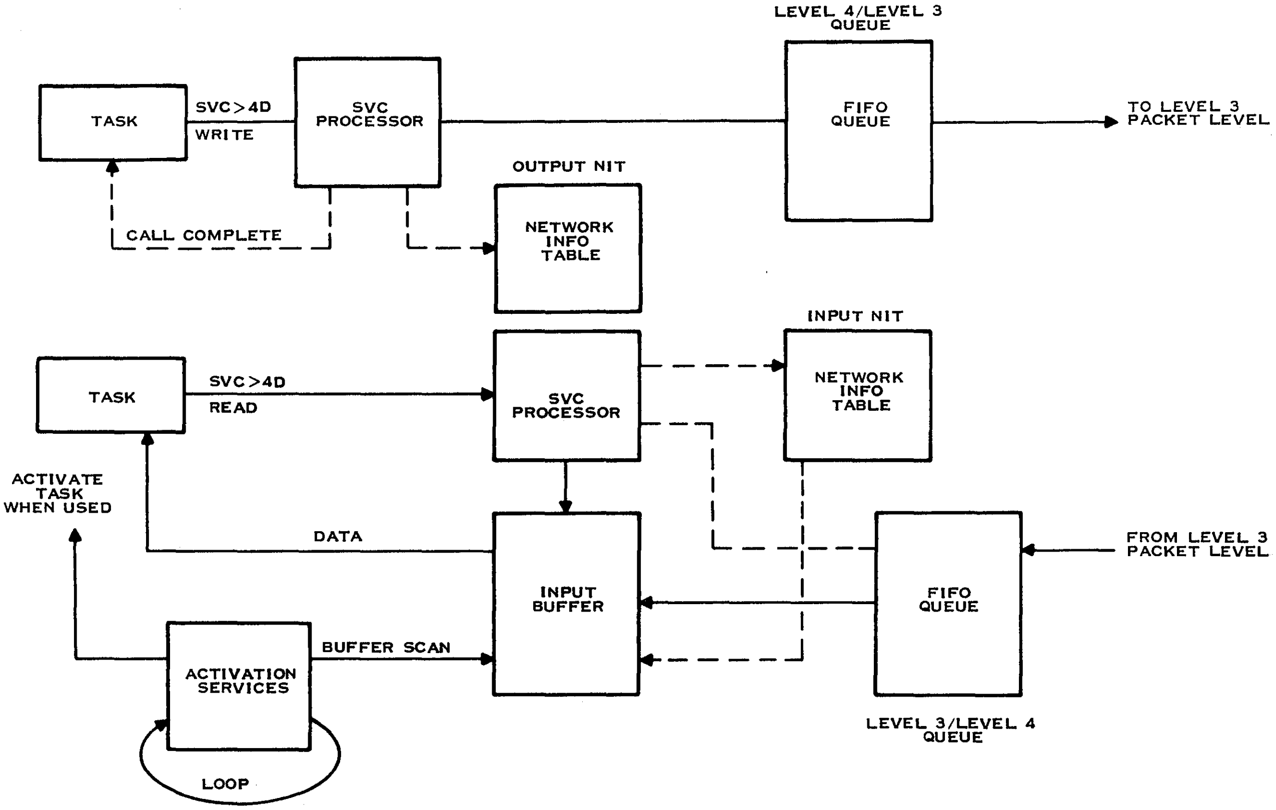


Figure 1-8 Level 4 User Interface

Activation Services Requests.

When a request for activation services is received, the requesting task ID is stored in the activation services queue. The requesting task then assumes an inactive state (suspended, terminated, or time delayed) while the activation services program searches the queue for a message addressed to the requesting task. When one is found, the requesting task is reactivated. The task then executes a read call operation to retrieve the message from the communications buffer. Operation codes 4 through 7 may be used by a task when requesting activation services using SVC >4D (see Section 4).

Network Information Tables (NITs).

The NITs contain several parameters that maintain control of the message streams flowing in the network. There are two sets of NITs: one at the primary station and one at each TX5 secondary station. The primary station requires an NIT for each primary (DX10) task that uses the HDLC communications system and for each secondary station in the network. Each TX5 secondary station requires an NIT for the secondary station, itself, and for each of its own tasks that use the HDLC communications system. Although the primary station maintains entries for all secondary stations, no NITs are required by the less sophisticated secondary stations, such as the PM550 and the TM990, at those stations. NITs are constructed at network generation time by the

NETGEN utility as a prelude to communication operations (see Section 3).

Each NIT may include the following types of data:

- Allowable message size
- Destination and source addresses
- Packet counters
- Secondary station status
- Communication event timers
- Task IDs
- Permanent virtual circuit IDs

When an SVC >4D call occurs, the SVC processor accesses the NIT to verify the validity of the call. The checks made by the processor include verification of source and destination addresses, acceptability of message size, and availability of buffers for the receiving station (the window size is not exceeded). If all parameters are validated, the caller's request is serviced.

1.4.2.5 Communications Package Utilities.

Several utilities are provided with the communications package to assist the user in the various phases of operation. The utilities provide the following facilities:

- Network generation
- Downloading a secondary station
- Preprocessing a program prior to downloading to a secondary station
- Activating/deactivating a secondary station
- Remote SCI
- Communication statistics accumulation

These utilities are described briefly in the following paragraphs, and in more detail in Section 3.

Network Generator.

The network generator (NETGEN) utility provides a means of defining the network configuration to the communications package. Each session between the user and NETGEN creates a file containing data that reflects the network configuration. In addition to defining the network to the software, this file can be used as a record for future reference (for example, when network expansions are under consideration). NETGEN requires the generation of decimal addresses (all between 0 and 9999). These addresses are assigned by the user. One address is required for each secondary station and for each task. These addresses are termed destination IDs (DIDs) and source IDs (SIDs).

Section 3 describes the network generator in further detail.

Downloader.

The downloader utility, as its name implies, is a means of transmitting a specified program from the primary station to a secondary station, where it is loaded into memory in preparation for execution. The program file(s) transmitted must already be in the object format applicable to that type of secondary station. This may be performed using the download preprocessor utility (see the next paragraph).

The download task may be activated by either of the following methods:

* By a secondary station requesting downloading. The secondary station transmits the request to the primary station.

* By the user. The user enters the SCI command DOWNLOAD at the primary station and supplies the appropriate responses to the prompts.

In both cases, download start and complete messages are written to the system log to note the start and end of the download attempt. If for any reason the download is aborted, an error message is written to the system log, showing the DID of the secondary station. The downloader includes facilities that allow a secondary station with a limited operating system (or none at all) to support HDLC communications and to receive the downloaded program(s). Section 5 contains a detailed description of the downloader utility.

Download Preprocessor.

The download preprocessor utility allows the user to convert DX10 object files to binary formatted object files compatible with the requirements of specified types of secondary stations. After conversion, the files are stored in a download directory and then downloaded under the control of the downloader utility. Section 5 contains a detailed description of the download preprocessor utility.

Activate and Deactivate.

The activate and deactivate utilities, available at the primary station, can be used to start up or shut down any or all of the secondary stations in the network. These utilities are called by the SCI commands ACT and DEACT, respectively, and are described in detail in Section 5.

Remote SCI.

The remote SCI utility provides the user at the primary station with a means of interacting with a TX5 secondary station that has been downloaded with the TX5 operating system and the TX5 Remote SCI package. Remote SCI is analogous to DX10 SCI. It is used, via a 911 VDT at the primary station, to perform some of the same operations at the secondary station that DX10 SCI performs for the primary station. Remote SCI is intended to completely replace the TX5 operator communication package (OCP) in the secondary station and provides the same overall capability.

Section 5 describes the remote SCI utility in detail.

Statistics Accumulation.

Statistics regarding communication line errors, receive/transmit operations, aborts, and so on, are continuously accumulated by the communications package during network operation. The statistical data is stored in counters at the primary station that are distributed both in main memory and in FCCC memory. All counter locations are reset to zero when the network is activated.

The data pertains to network operations as well as primary and secondary operations. Section 5 describes the statistics facilities in more detail.

Section 2

Planning the Network

2.1 GENERAL

This section provides background information on planning the communications network prior to actually constructing it. Once the network has been constructed at the primary station, additional secondary stations and applications programs can be added by repeating the network generation process. However, it is possible to make provisions during the initial design for the inclusion of future secondary stations and applications programs. Therefore, careful consideration of future expansion needs, both short term and long term, is needed during the planning phase. By including anticipated additions in the initial design, they can be brought online with a minimum of effort.

The network planning process involves developing plans for constructing the primary station and each secondary station. In each case, the principle concern is assigning numerical values to identify each station and each applications program that will be part of the network.

Examples of the required network parameters are given in this section, as well as sample forms that can be used to provide the users of the network with permanent documentation.

2.2 NETWORK ADDRESSING

The DX10 HDLC Communications Package provides support for as many as 32 secondary stations attached to each multipoint line. This is an electrical limitation. Within the primary station, the number of applications programs that can use the communications package is limited only by the capacity of the computer.

The individual responsible for constructing the network assigns a network identification number (network ID) to each station address and each applications program (task) that will use the communications package. This enables the communications package to route information to the correct location. The network IDs assigned to the stations and applications programs are stored in network information tables (NITs).

2.2.1 Network Information Tables (NITs)

The NITs are contained in the communications package at the primary station and at each TX5 secondary station. PM550 and TM990 secondary stations do not contain NITs.

The network assignments contained in the NITs are of two types. The long form NIT entry is used to identify stations, while the short form NIT entry is used to identify programs in a station. In the primary station, there is one long form NIT entry for each secondary station in the network. At each secondary station, there is only one long form NIT entry, since all information sent from a secondary station is sent to the primary station. Each program in the primary and secondary stations that use the communications package has a short form NIT entry associated with it. The short form NIT entry includes only the minimum information necessary to route input data to the proper program. The short form NIT relates a task ID to the program's network ID.

2.2.2 Planning the Network Assignments

Several addresses and IDs are associated with the DX10 primary station and with each secondary station in the network. Since the secondary is addressed by more than one layer or level of software, each software component that requires an address or ID is assigned an appropriate identifier during system generation and/or during network generation. These IDs and/or addresses are as follows:

- The LUNO assigned to each four-channel communications controller (FCCC) channel or line. These are assigned automatically during network generation and TX5 system generation.

- The (physical) switch address (also called the drop address) of each secondary station. This address is the HDLC destination address and is assigned during network generation.

- The logical channel identifier (LCI) that is assigned to each permanent virtual circuit (PVC) within the network. This is assigned automatically during network generation.

- The task ID for each DX10 and TX5 task that uses the communications package. These are assigned during system generation and network generation.

- The network ID for each sending task and for each receiving task. These are assigned during network generation.

All addresses should be specified and documented before either system generation or network generation begins. Planning the network in this way facilitates both system and network generation and may prevent aborting either process due to nonavailability of pertinent data.

2.2.2.1 Logical Unit Numbers (LUNOs).

The LUNOs required are as follows:

- Those associated with each FCCC channel to be used by the communications package at the DX10 primary station
- The LUNO assigned to the local-line module (LLM) at each TX5 secondary station

All the required DX10 LUNOs are assigned automatically during DX10 network generation. TX5 LUNO´s are assigned by the user during TX5 system generation. For DX10 network generation, communication device CM01 (channel zero) is assigned LUNO >C0, device CM02 (channel 1) is assigned LUNO >C1, and so on in sequence. For the TX5 system generation, the LLM is assigned LUNO >20. The FCCC channel number must be entered during network generation to properly associate FCCC channels and secondary stations.

2.2.2.2 Assigning Station Addresses.

Each secondary station in the network must be assigned a station address, which is determined by the address switch setting on the LLM. The address is entered as a hexadecimal number and has a range of >00 through >3F (0 through 63). After each address is set physically, it should be recorded so that it can be entered correctly during network generation. A switch address must be selected for each secondary in the network. It must be unique within a single multipoint line but not necessarily unique within the whole network. For example, if all four channels of an FCCC board are used, it is permissible for the same station address to be used for four different secondary stations if they are all on different lines.

2.2.2.3 Assigning Logical Channel Identifiers (LCIs).

In multipoint operation, a unique LCI assignment is made for each secondary station in the network. The LCI is normally associated with packet-switching network operation and is usually assigned by a network authority. The LCI consists of a four-bit virtual channel group and an eight-bit logical channel number, thus providing for a maximum of 4096 LCIs. The LCI assignent is arbitary and is based on the FCCC channel number to which the station is attached and on the switch address of the station. For example, a secondary station with a switch address >1F on FCCC channel 1 is automatically assigned an LCI of >11F.

2.2.2.4 Assigning Task IDs.

During network generation an association is made between the task ID and the corresponding network ID for each DX10 and TX5 task that uses the communications package. These associations should be determined and documented prior to network generation. The user should ensure that all DX10 tasks that use the communications package are installed in the proper program file. Similarly, all TX5 tasks that use the communications package must be included in TX5 system generation. Those DX10 and TX5 task IDs that are reserved for the communications package are noted in later sections of this manual. None of these reserved IDs are available for assignment to other tasks.

2.2.2.5 Assigning Network IDs (NIDs).

A unique network ID (NID) must be assigned to each physical and/or logical component within the network. This NID is a four-digit decimal number and is referred to as a source ID (SID) or a desination ID (DID) when associated with a transmitting program or a receiving program, respectively. NIDs are assigned as follows:

* One NID to each PM550 secondary station

* One NID to each TX5 secondary switch address

* One NID to each TX5 task that uses the communications package

* One NID to each DX10 task that uses the communications package

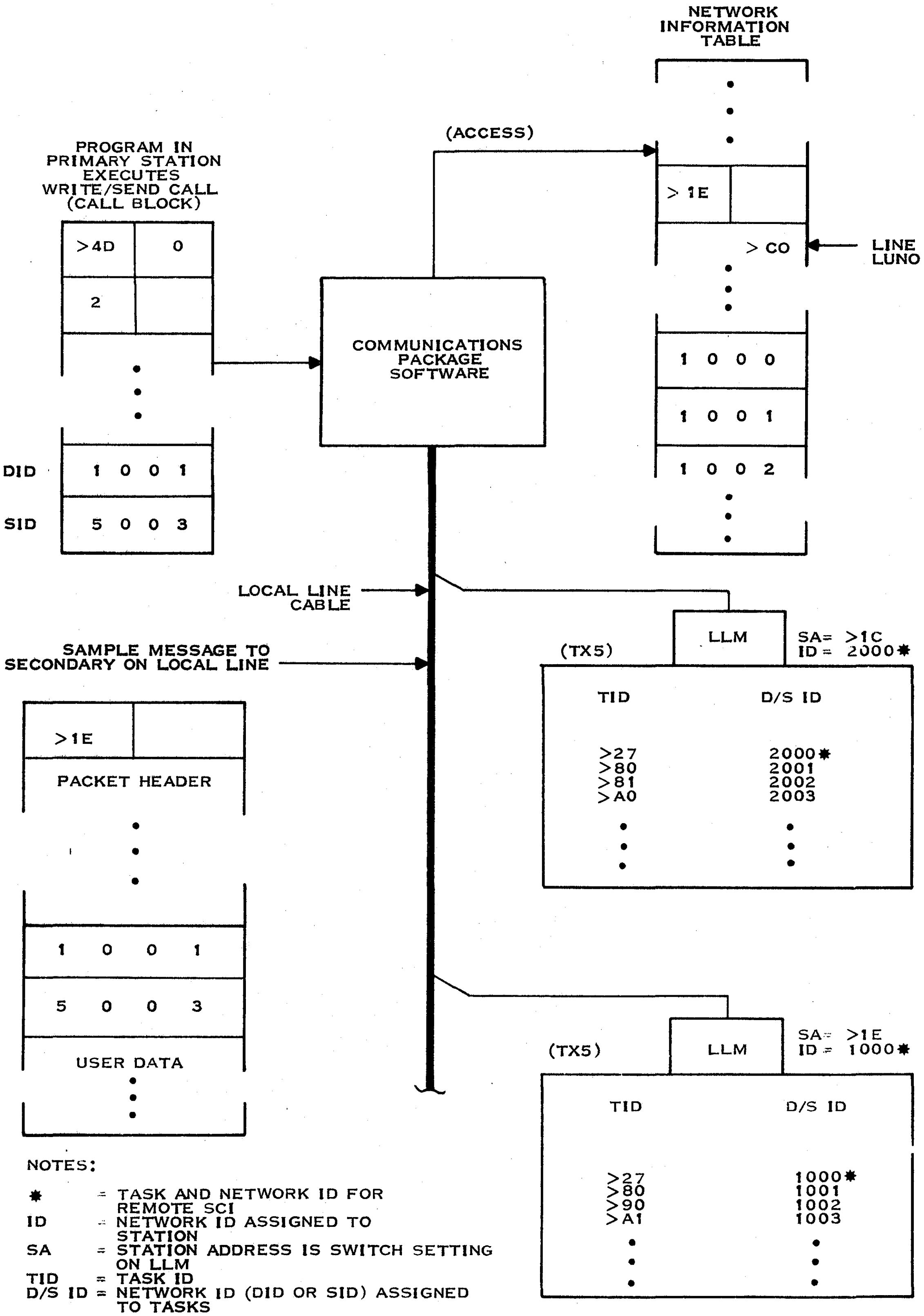
* One or more NIDs to each TM990 secondary station

The NIDs assigned are entered by the network generator utility (NETGEN) into the data structures used by the communications software at the DX10 primary station and at each TX5 secondary station.

2.2.3 Network Addressing Example

In Figure 2-1, a network consisting of a single multipoint line is used to illustrate the assignment of station addresses, network IDs, and task IDs within the communications network. In this example, an applications program in the primary station with network ID 5003 (that is, source ID (SID) 5003) is sending information that is addressed to network ID 1001 (that is, destination ID (DID) 1001). This DID equates to task >80 in the TX5 secondary station that has been assigned a station address of >1E. The program that sends the data is required to supply the correct DID (1001). The communications software in the primary station determines from a network information table that DID 1001 is in the secondary station that has the station address >1E. The routing of the data to the proper secondary station is made by the communications software based on the information contained in the long form NIT entry.

This example illustrates that task IDs can be duplicated in secondary stations that share the same multipoint line.



2277884

Figure 2-1 Network Addressing on the Same Multipoint Line

2.3 DOCUMENTING THE NETWORK CONFIGURATION

Figure 2-2 and Figure 2-3 illustrate suggested formats for documenting primary station and secondary station parameters, respectively. Most of this information is required to be entered during network generation, DX10 system generation, and TX5 system generation. This type of documentation on network assignments should be available for reference before network generation, DX10 system generation, and TX5 system generation are started. The network planner should cross-reference the information sheets to facilitate the association of the software and hardware components of the communications package.

2.3.1 Documenting the Primary Station

The example form in Figure 2-2 shows the following primary station information:

* FCCC card and line data
* The total number of secondary stations
* The type of each secondary station, its switch address, and the FCCC line to which it is connected
* The total number of DX10 tasks using the communications package
* User program filename and LUNO
* The association between task IDs and network IDs at the primary
* The association (if any) between each DX10 task and each secondary task and/or network ID

PRIMARY STATION INFORMATION SHEET

NO. OF FCCC CARDS: 1
NO. OF FCCC LINES: 3
NO. OF ATTACHED SECONDARIES: 36
NO. OF DX10 TASKS USING COMM. PACKAGE: 23
USER PROGRAM FILENAME: .INDSCOMM.USERPROG
PROGRAM FILE LUNO: >40

BID TASK ID/NID FOR DX10 TASKS -	TASK ID/NETWORK ID
	>50/5000
	>51/5001
	>90/5090

TASK AND/OR DEVICE ASSOCIATION

DX10 TASK TO	SECONDARY TASK ID/NID	FCCC CHANNEL/ DEVICE	SECONDARY SWITCH ADDRESS	SECONDARY TYPE
REMOTE SCI	>27/1000	0/CM01	>1E	990/5
>50/5000	>A0/1001	0/CM01	>1E	990/5
	>A5/1002	0/CM01	>1E	990/5
	>B1/1003	0/CM01	>1E	990/5
	>B1/1015	0/CM01	>03	990/5
	>A0/2005	1/CM02	>1F	990/5
>50/5001	>90/3220	2/CM03	>2A	990/5
	>90/3245	2/CM03	>3F	990/5
>77/5020	---/3500	2/CM03	>2E	TM990
>7A/5030	---/3600	2/CM03	>2D	PM550
..	..	.	..	..
..	..	.	..	..
..	..	.	..	..

Figure 2-2 Primary Station Documentation

2.3.2 Documenting the Secondary Stations

The example form in Figure 2-3 shows the following secondary station information:

- FCCC channel used by the secondary station and the LUNO assigned to it
- Each secondary station type and location
- Each secondary station address and network ID
- Each secondary station buffer size
- Each secondary station LCI assignment
- The association between task IDs (if any) and network IDs at each secondary station

SECONDARY STATION INFORMATION SHEET

FCCC CHANNEL (LINE) NO: 0   LUNO ASSIGNED: >C0

SECONDARY LOCATION	SYSTEM or TYPE	BUFFER SIZE	VCG/LCN (LCI)	STATION ADDRESS	NETWORK ID	TASK ID/ NETWORK ID ASSIGNMENTS
ROOM 12	990/5	256	00/01E	>1E	1000	>27/1000 * >A0/1001 >A5/1002 >B1/1003 >B2/1004 >B3/1005
MFG BAY 0110	990/5	256	00/021	>21	1100	>90/1101 >A0/1102
FACILITY ROOM #2	TM990	128	00/010	>10	1200	---/1200
BLDG. T01	PM990	260	00/012	>12	1300	---/1300

* Note that task ID >27 (remote SCI) has the same network ID as the station.

Figure 2-3 Secondary Station Documentation

Section 3

Constructing and Activating the Network

3.1 GENERAL

This section describes the procedures necessary to construct and activate the DX10 HDLC Communications Package. To construct the network, a fully operational DX10 operating system, release 3.4, is required. The steps necessary to provide the primary station and each secondary station with communications capabilities are outlined first and then described in more detail.

After the network has been planned and documented, you must begin the process of constructing the software required to operate the network. This consists of building DX10 and TX5 communications data structures, generating the operating systems and linking the appropriate modules to build the desired operating system and the communications package. The primary and secondary stations must be customized to a certain extent to provide the software modules required by the communications package.

Specifically, the major steps required to construct and activate an HDLC communications network are as follows:

1. Installation of the HDLC modules in a fully operational DX10 system and DX10 system generation to include the FCCC device service routine (DSR).

2. HDLC network generation to create the data structures that describe the the network.

3. TX5 secondary station system generation for each TX5 station in the network.

4. Assembly and linkage of all the programs required to operate the communications package.

5. A DX10 warm-start to include the communications package software in the DX10 system.

6. An HDLC communications warm-start to activate the HDLC communications package software.

Those steps concerned with the primary station are further discussed in paragraph 3.2. The steps concerned with TX5 secondary station construction are described in paragraph 3.4.

3.2 DX10 PRIMARY SYSTEM

A correctly functioning DX10, release 3.4, is required to operate the communications package at a primary station. Also, a four-channel communications controller (FCCC) board must be prepared and installed in the system. The following paragraphs describe the steps required to build the primary communications system and install the FCCC board.

3.2.1 Building the Primary System

Building a DX10 system with HDLC communications capabilities requires that the FCCC DSR be included in DX10 system generation to provide the interface between the host software and the FCCC software. The following steps are necessary to accomplish DX10 system generation with the FCCC DSR:

1. Preparation and installation of the FCCC board (see paragraph 3.2.2).
2. Installation of the HDLC communications object files using the procedures given in the DX10 HDLC Communications Package Object Installation Guide. The principal steps in this procedure are as follows:
    a. Patching the GEN990 task for DX10 systems, releases 3.4.1 and earlier. This step is not required for DX10 systems, release 3.4.2 and later.

    b. A DX10 operating system generation using the XGEN utility.

    c. Execution of the communications DSR installation procedure for FCCC device service routine generation.

    d. Assembly and linkage of the new operating system using the ALGS (Assemble and Link Generated System) command.

    e. Application of the patch file using the PGS (Patch Generated System) command.

    f. Application of patches to enable the SVC >4D processor for the HDLC package using the XPS (Execute Patch Synonym Processor) command.

   g. Test and installation of the resulting DX10 system using the TGS (Test Generated System) and IGS (Install Generated System) commands.

3. Network generation using the NETGEN utility (see paragraph 3.3) to create the data structures that describe the following:

   a. Each secondary station attached to the DX10 primary.

   b. Each task in the TX5 secondary stations that will communicate with DX10 tasks.

   c. Each local (DX10) task that uses the communications package to communicate with secondary tasks or devices.

NOTE

If the network includes one or more TX5 secondary stations, the operating systems for those stations must be generated before proceeding to the next step. See paragraph 3.4.

4. Link editing to link all of the programs required to operate the communications package using the SCI command COMMGEN at a DX10 video display terminal (VDT) (see paragraph 3.5)

5. Execution of a DX10 warm-start (halt, reset, load) to include the communications package software in the DX10 system (see paragraph 3.5).

6. Execution of a communications warm-start to activate the communications package software using the SCI command COMMGO at a DX10 VDT (see paragraph 3.5)

The communications warm-start program sends a message to the system log at the start and end of the warm-start process. After the warm-start program has completed its functions (indicated by a warm-start completion message written to the system log), you can download to any secondary station and activate any applications programs that use the communications package.

3.2.2 Installing the FCCC Board in the 990/10

The FCCC board provides the interface between the CPU and the communications line. Use of the FCCC board requires that the FCCC DSR be included with the DX10 operating system during system generation.

The DSR provides the TILINE interface support between the CPU and the FCCC board. The DSR accepts call block formats for read, write, and special operations. These formats are described in the Model 990 Computer Four Channel Communications Controller Installation and Operation Manual. The user of the communications package does not interact directly with the FCCC board.

The FCCC board requires one full slot in the 990 computer chassis, through which it communicates with the host computer via the TILINE interface. Before the board is installed in the host computer chassis, a chassis slot must be selected. The TILINE access-granted and interrupt level option jumpers must be configured to allow the software in the host computer to recognize the FCCC. Additionally, the TILINE slave address switches (SYY2) and the polled network ID switches (SYY20) on the FCCC board must be set as follows:

<table>
<tr><th>Switch ID</th><th>Directions</th></tr>
<tr><td>SYY2</td><td>Set this switch combination to the TILINE address specified during system generation. The following is an example that corresponds to a TILINE address of >F900:
<table>
<tr><th>Section ID</th><th>Position</th></tr>
<tr><td>1 through 5</td><td>OFF</td></tr>
<tr><td>6</td><td>ON</td></tr>
<tr><td>7</td><td>OFF</td></tr>
</table>
</td></tr>
<tr><td>SYY20</td><td>Set sections 1 through 8 on this switch to the OFF position.</td></tr>
</table>

Ensure that computer power is turned off before installing the FCCC board in the selected chassis slot. Refer to the FCCC installation and operation manual for more detailed information on the selection and preparation of a chassis slot for the FCCC board.

3.3 NETWORK GENERATION

Network generation is accomplished by using the network generator utility NETGEN, which is activated by the SCI command NETGEN. This is a DX10 program that issues prompts via a VDT. NETGEN uses the responses to the prompts to create the data tables and structures required for proper operation of the communications package at the DX10 primary and TX5 secondary stations. Structures are not required for inclusion with PM550 and TM990 secondary station software.

Before activating NETGEN, the user must know how the network is physically configured and which logical assignments (such as task IDs) are required during the network generation process. It is suggested that appropriate documentation be created prior to NETGEN activation (see Section 2). Attempting to create network tables without sufficient documentation may lead to errors and, as a result, be unneccessarily time-consuming.

The prompts issued by NETGEN are requests for the values of key parameters associated with the stations in the network.

The data derived from a NETGEN session is stored on a disk file (the configuration file) for ease of access and maintainability. It contains the latest (chronologically) entered data for all stations defined for the network and can be interrogated at any time. This accessibility provides the user with a way to easily modify network-related structures without having to start over at the beginning. It should be noted, however, that a change to the configuration file does not cause an immediate, automatic change to the operation of the communications system software. Some of the procedures used to initially build the tables for a station must be repeated when a change is made; the tables must be reassembled and relinked with the communications package software, and a system warm-start must be executed.

The files created by NETGEN are under the directory .INDSCOMM. Table 3-1 lists the subdirectories and files that are also created and lists the use of each file.

Table 3-1 HDLC Directories and Files

Directory and Filename	Description
.INDSCOMM.NETGEN.CONFIG	System configuration file, containing data that describes the network. It is created and maintained by NETGEN. The user need not access this file.
.INDSCOMM.C$SOURCE	File containing the NITs for the DX10 primary. It is assembled and linked with the communications software.
.INDSCOMM.NETBAT	File containing the batch stream built by NETGEN. It is bid by the communications system generator (COMMGEN) and builds one level 2 table and one offload control file for each FCCC board. The results of executing this batch stream are written to .INDSCOMM.NETLST.
.INDSCOMM.C$TABLEn	File containing the data structures required for the nth FCCC board where n = 1, 2, 3, etc. The data structures define each line and the associated secondary stations. They are assembled and linked with the FCCC software by the COMMGEN procedure.
.INDSCOMM.CONTROLn	File used to offload the nth FCCC board, where n is 1, 2, 3, etc.
.INDSCOMM.NETGEN.SECnnnn	File that is unique to each TX5 secondary station included in the system. The filename is created by using a four-digit, binary-coded decimal number (nnnn) that is unique within the network. This file must be assembled and linked with the TX5 secondary system. The object and listing files must be named by the user and it is recommended that those files be included in the same directory. An example follows: .INDSCOMM.NETGEN.OBJECT.SECnnnn .INDSCOMM.NETGEN.LIST.SECnnnn

NETGEN has the following capabilities:

* Generating a new configuration
* Modifying an existing configuration as follows:
  - Adding to an existing configuration
  - Changing an existing configuration
  - Deleting from an existing configuration

Each of these capabilities is described in the following paragraphs. However, before NETGEN can be activated, the communications package must be installed and global LUNO >90 must be assigned to the program file .INDSCOMM.PROGRAM, using the SCI command Assign Global LUNO (AGL). Refer to the DX10 reference manuals for detailed instructions. After the initial assignment, the COMMGO command automatically assigns global LUNO >90.

3.3.1 Conventions and Examples

In the following examples of NETGEN capabilities, the prompts are given in the identical sequence that they appear on the VDT at which NETGEN is active. The prompts are displayed in the TTY mode; that is, each prompt is displayed after the previous prompt has received a response. The responses shown in the examples are all user-entered; NETGEN does not display any default values. All responses must be followed by pressing the RETURN key, referred to in some prompts as CR (carriage return). When there are no more entries to be made, pressing the RETURN key either causes groups of related prompts to be skipped or causes NETGEN to be terminated.

Numerical responses to NETGEN prompts can be in decimal or hexadecimal for the following parameters:

* Data rates
* Network IDs
* Poll intervals
* Message sizes
* Queue buffer sizes
* Task IDs
* TILINE addresses

If a value is entered in hexadecimal, it must be preceded by a zero (0) or angle bracket (>).

3.3.2 Activation and Termination

To activate the network generator, enter the SCI command NETGEN. This produces the following display:

```
[] NETGEN
   NETWORK GENERATION
   GENERATE NEW CONFIGURATION (Y/N):
```

The response to this prompt determines the specific group(s) of prompts that will be displayed next and their sequence. An entry of Y (yes), followed by a RETURN, starts the sequence of prompts described in paragraph 3.3.3. An entry of N (no), followed by a RETURN, starts the sequence described in paragraph 3.3.4.

NETGEN terminates normally after a response has been entered to the DO YOU WANT TO SAVE THE CONFIGURATION prompt as described in the following paragraphs. To terminate NETGEN at any time, enter Q and press RETURN in response to any prompt.

3.3.3 Generating a New Configuration

The generation of a new configuration requires the entry of the following information in response to the appropriate prompts:

- FCCC TILINE address
- Communications device name
- FCCC channel number
- Channel data rate (baud rate)
- Internal/external clocking requirement
- Message retry requirement

To start generating a new configuration, enter a Y in response to the GENERATE NEW CONFIGURATION (Y/N) prompt and press the RETURN key. The next prompt displayed is the first in a series that describes an FCCC port and its associated line to NETGEN. The prompt is as follows:

```
  FCCC BOARD TILINE ADDRESS:  0F900
```

Enter the value of the TILINE address used during system generation (>F900 in the preceding example). If more than one FCCC board is included in the system, enter the TILINE address of the first communications device. The next prompt is as follows:

```
  DEVICE NAME:  CM01
```

Enter the device name assigned to one of the FCCC channels during system generation (CM01 in the preceding example). Each FCCC channel is considered to be a single device and is named CMxx where xx has the value 01 through 99. The communications package software will ultimately assign a LUNO to this device.

The next prompt is as follows:

```
  FCCC CHANNEL NUMBER (0 - 3):  0
```

Enter the FCCC channel number associated with the device name just entered (0 in the example). For example CM01 may be assigned to channel 0 of this board, CM02 to channel 1, and so on. Alternatively, CM02 may be assigned to a channel on another FCCC board. The NETGEN user must be aware of the entries made during DX10 system generation to ensure that the correct association is made between channel numbers.

```
  CHANNEL DATA RATE:  7200
```

Enter the data rate in bps (bits per second) chosen for this channel (7200 in the example). The values can be 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, or 9600. Refer to the FCCC data rate table in the FCCC Installation and Operation Manual to determine an optimum selection. The next prompt is as follows:

```
  INTERNAL CLOCKING (Y/N):  Y
```

Enter Y (as in the example) for local-line operation or for operation with modems that do not supply their own clocks. Enter N for modem operation with modems that supply their own clocks.

The next prompt is as follows:

```
  MESSAGE TRANSMIT RETRIES (0 - 3):  2
```

Enter the number of transmission attempts to be made to any secondary on this line before the next attempt is aborted. Note that if 2 is entered (as in the example), a total of 2 transmission attempts will be made to the secondary before aborting. An attempt is judged unsuccessful if no response is received within the time-out period of two seconds.

The preceding responses and prompts describe an FCCC port and line to NETGEN. After receiving a response to the prompt MESSAGE TRANSMIT RETRIES, NETGEN continues with prompts that request descriptive information for the first secondary station. The following paragraphs assume that the first station to be described is not a TX5. An example describing a TX5 station is provided later.

3.3.3.1 Generating a Non-TX5 Secondary Station.

The generation of a non-TX5 secondary station requires the entry of the following information in response to the appropriate prompts:

* Station drop address
* Maximum time required between polls
* Station network address
* Maximum message size
* Type of station (PM550 or TM990)

The following prompts describe (to NETGEN) a non-TX5 secondary station attached to the FCCC line just defined. A maximum of 32 secondary stations can be attached to each FCCC line. The first prompt is as follows:

```
STATION DROP ADDRESS (0 - >3F, CR TO CONTINUE):  62
```

The response to this prompt (62 in the example) is the secondary station drop address as set on the pencil switches on its communication controller card. The response must be a value of 0 through 63 decimal (0 through >3F). (A RETURN response (CR) to this prompt does not result in the generation of more prompts for the current FCCC line. Instead, prompts for the next FCCC line are given.) The next prompt is as follows:

```
MAX TIME BETWEEN POLLS (0-32000/1 = 250 MSEC):  8
```

The response to this prompt is the number (8 in the example) of 250-millisecond intervals of time selected as a delay between polls to this secondary station. The example response results in a delay of two seconds between polls. A response of 0 indicates that this secondary station should be polled at every opportunity. The next prompt is as follows:

```
STATION NETWORK ADDRESS (0100 - 9999): 0100
```

Enter the four-digit BCD network address (0100 in the example) assigned to this station during the planning stage. This network address cannot be assigned to any other secondary station or to any primary or secondary station or task. Once assigned, it is unique within the network and any attempt to assign the address again during NETGEN results in an error message to the terminal.

The next prompt is as follows:

```
SECONDARYS MAX MESSAGE SIZE (1 - 512 BYTES): 128
```

The required response to this prompt (128 in the example) varies for different types of secondary stations. Refer to the appropriate manual for the specific type of secondary station. For TX5, enter 256; for PM550, enter 260; and for TM990, enter 128. An entry of 512 is accepted by NETGEN but a transmitted message of that size to a TX5, PM550, or TM990 secondary station results in an error at the secondary station. The next prompt is as follows:

```
TX5 SECONDARY STATION (Y/N): N
```

The response to this prompt (N in the example) indicates whether or not the station is a TX5. This information is used to select the correct window size (see paragraph 1.4.2.3).

The next prompt is as follows:

```
TM990 SECONDARY NETWORK ID (0100 - 9999): 0200
```

If the station is a PM550, enter RETURN. If it is a TM990, the response (0200 in the example) assigns a network ID to the TM990. The network ID is then entered in the output NIT (long form) at the primary station. The TM990 secondary station possesses a multitasking capability and can therefore have more than one network ID. This prompt is repeated until all TM990 network IDs are entered. When none remain, enter RETURN to continue network generation.

The preceding prompts, beginning with STATION DROP ADDRESS, described one non-TX5 secondary station for FCCC line 0 to NETGEN. NETGEN continues displaying prompts to assign more secondary stations to line 0 until a RETURN is entered (without a value) in response to the prompt STATION DROP ADDRESS. NETGEN then proceeds to issue prompts for the next FCCC port and line included in the system.

3.3.3.2 Generating a TX5 Secondary Station.

The generation of a TX5 secondary station requires the entry of the following information in response to the appropriate prompts:

- Station drop address
- Maximum time between polls
- Station network address
- Maximum message size
- Type of station (TX5)
- Remote SCI requirements
- Remote logging requirements
- Task IDs and task network IDs

This paragraph gives an example that describes (to NETGEN) a TX5 secondary station attached to FCCC line 0. Descriptions of the first five prompts were given in paragraph 3.3.3.1 and are not repeated here. These five prompts are as follows:

```
STATION DROP ADDRESS (0 - >3F, CR TO CONTINUE):  01

MAX TIME BETWEEN POLLS (0-32000/1 = 250 MSEC):  8

STATION NETWORK ADDRESS (0100 - 9999):  1100

SECONDARY'S MAX MESSAGE SIZE (1 - 512 BYTES):  256

TX5 SECONDARY STATION (Y/N):  Y
```

A Y response to the last prompt is required to identify a TX5 secondary station. This initiates another series of prompts to determine the requirements and characteristics of the TX5. The first of these prompts is as follows:

```
REMOTE SCI INCLUDED (Y/N):  Y
```

This prompt provides the option to include the utility REMSCI in the secondary system. If the remote SCI task is not to be included in this TX5 secondary, enter N. This causes NETGEN to skip the next two prompts and begin prompting for TX5 task IDs. If the remote SCI is to be included, enter Y. This causes NETGEN to issue the following procedural message:

```
REMOTE SCI (TASK >27) ASSIGNED NETWORK ID 1100
```

This message indicates that remote SCI has been assigned the same network ID as the secondary station, in this case 1100. NETGEN continues to prompt as follows:

```
REMOTE LOGGING INCLUDED (Y/N):  Y
```

This prompt provides the option of logging at either the primary or the secondary station. If remote logging to the primary system log is desired, enter Y. If N is entered, no TX5 system error messages will be written to the primary station. In this case, if OCP is not included in the TX5, system error messages will be lost. If N is entered, NETGEN skips to the TX5 TASK ID prompt. If Y is entered, the next prompt is as follows:

```
REMOTE LOG NETWORK ID (0100 - 9999):  1101
```

The response to this prompt (1101 in the example) is the the network ID assigned to the remote log task. NETGEN then issues the following procedural message to indicate that the ID entered in response to the previous prompt has been successfully assigned to the remote log task:

```
REMOTE LOGGER (TASK >28) ASSIGNED NETWORK ID 1101
```

The next prompt is as follows:

```
TX TASK ID (0 - >FF):  >10
```

The response to this prompt (>10 in the example) is the task ID of a TX5 task that requires a network ID. The same task must now be assigned a network ID if the artificial data statistic will ever be run to this TX5 secondary station (see Section 5). The next prompt is as follows:

```
TX TASK NETWORK ID (0100 - 9999):  1102
```

The response to this prompt must be the network ID assigned to the TX5 task (ID >10) that was entered in response to the previous prompt. NETGEN continues to prompt for TX5 task IDs until all are entered and assigned network IDs. When no more tasks require ID assignments, enter a RETURN in reponse to the last TX TASK ID prompt displayed.

The description of the TX5 secondary station is now complete; NETGEN proceeds to prompt for the next station, as described in the following paragraph.

NOTE

> It is not necessary to assign network IDs via NETGEN to the tasks that comprise the TX5 secondary communications package included in the TX5 system generation. No problem results from making such assignments except the inefficient use of memory and network IDs in both the primary and secondary stations.

3.3.3.3 Generating Additional Stations and Lines.

NETGEN continues to prompt for more secondary stations, more devices attached to the FCCC line, and more FCCC lines. The next prompt requests the drop address of the next secondary station to be described.

```
STATION DROP ADDRESS (0 - >3F, CR TO CONTINUE):
```

If no more secondary stations on the current FCCC line remain to be described, enter a RETURN (CR) in response. If an address is entered, the cycle of prompts for the specified secondary station begins again. Otherwise, NETGEN proceeds to prompt for any other communication devices attached to this FCCC line as follows:

```
DEVICE NAME: <return>
```

If the response to this prompt is a valid device name and number, the prompt sequence previously described is repeated for the specified FCCC line. If the response is a RETURN, NETGEN prompts for the next TILINE address as follows:

```
FCCC BOARD TILINE ADDRESS: <return>
```

Entering a RETURN in response to this prompt indicates that there are no more FCCC lines to add to the communications network. If a valid TILINE address is entered, the next prompt requests the name of the first device attached to that line. This is again followed by the cycle of prompts previously described.

When all secondary stations, lines, and devices have been described, NETGEN prompts for the data needed to create the structures that associate DX10 task IDs with network IDs. These are described in the following paragraph.

3.3.3.4 DX10 Task and Network ID Association.

After completing the descriptions of the FCCC lines and secondary stations, NETGEN prompts for the information that associates specific DX10 tasks and network IDs. The resulting structures are referred to as input, or short-form, NITs. After this association is complete, NETGEN prompts for confirmation of the descriptions and offers the opportunity to modify the entries.

NETGEN proceeds to prompt for task and network IDs as follows:

```
DX10 TASK ID (>15 - >FF):  >40
```

The example response, >40, identifies a DX10 task that uses the communications package to exchange information with secondary stations or tasks. Tasks installed on the program file .INDSCOMM.USERPROG must be assigned a task ID of >15 or larger. Task IDs from 0 to >15 are reserved for the HDLC communications package. DX10 task IDs cannot be duplicated even though they may be on different program files. The next prompt is as follows:

```
DX10 TASK NETWORK ID (0100 - 9999):  1124
```

The response to this prompt assigns a network ID (1124 in the example) to the DX10 task ID previously entered (>40). This task is now addressable using this ID. NETGEN continues to prompt for DX10 task IDs until all are entered. When no more remain, enter a RETURN to the last DX10 TASK ID prompt. NETGEN now offers choices as follows:

```
ADD, BUILD, CHANGE, OR DELETE (A,B,C,D):  B
```

This prompt allows the user to make any desired changes before the configuration is built by the communications system. If NETGEN was performed correctly, enter B (as in the example) to build the new configuration. (If modifications are required enter A, C, or D as appropriate and refer to paragraph 3.3.4.) The next prompt is as follows:

```
DO YOU WANT TO SAVE CONFIGURATION (Y/N):  Y
```

The Y response shown in the example saves the configuration just entered on the file .INDSCOMM.NETGEN.CONFIG. Saving the configuration file allows changes to be made later, using NETGEN. An N response leaves the last configuration intact; that is, the configuration file is not modified in any way. After the response to this prompt is entered, NETGEN terminates.

Network generation creates the source files needed to operate the DX10 primary station communications programs and the TX5 secondary station communications programs. NETGEN builds all the data structures needed for these two types of systems to operate.

3.3.4 Modifying an Existing Network Configuration

The following paragraphs describe the NETGEN procedure that allows modification of a previously built configuration. NETGEN allows a user to augment, change, or delete from the network configuration. It is possible to add a new secondary station, a TX task to an existing secondary station, or a DX10 task to the primary station. It is also possible to change an entry for an FCCC line or a secondary station, to change the length of the internal communications queues, or to delete a network ID from the network. If all network IDs on a line are deleted the line is also deleted.

The initial conditions necessary for modifying the network are that LUNO >90 is assigned to the DX10 program file .INDSCOMM.PROGRAM and that the configuration was saved from a previous network generation.

After the network generator has been activated using the command NETGEN (see paragraph 3.3.2), the following display appears:

```
NETWORK GENERATION
GENERATE NEW CONFIGURATION (Y/N):
```

The response to this prompt determines the specific group(s) of prompts displayed next and their sequence. An entry of Y (yes), followed by a RETURN, starts the sequence appropriate to a new network configuration. An entry of N (no), followed by a RETURN, starts one of the modification sequences described in the following paragraphs.

3.3.4.1 Adding to an Existing Network Configuration.

This paragraph describes the processes required to add an entry to an existing network. The addition can be one of the following:

- A new task to an existing TX5 or TM990 secondary station. This may require that the TX5 system be linked, preprocessed, and downloaded again after the addition is made.
- A new task to the DX10 system (added to the program file .INDSCOMM.USERPROG) that will use the communications facility.
- A new secondary station such as a PM550, TM990, or 990/5.

The following examples illustrate the use of the add facility.

Adding a Primary Task.

Adding a primary task requires the entry of the following information in response to the appropriate prompts:

- The DX10 task ID for each additional task
- The DX10 network ID for each additional task

To add a DX10 primary task to the communications system, enter an N (as shown), followed by a RETURN, to the following prompt:

```
GENERATE NEW CONFIGURATION Y/N: N
```

To specify an addition, enter an A in response to the next prompt:

```
ADD, BUILD, CHANGE, OR DELETE (A,B,C,D): A
```

Then, to specify a DX10 task, enter D in response to the next prompt:

```
ADD DX10 TASK, TX5 TASK/TM990 ID OR NEW SECONDARY (D,T,S): D
```

The next prompt is as follows:

```
DX10 TASK ID (>15 - >FF): >50
```

The response to this prompt (>50 in the example) identifies a DX10 task that must be added to the communications network. The next prompt is as follows:

```
DX10 TASK NETWORK ID (0100 - 9999): 1200
```

The response to this prompt (1200 in the example) assigns a network ID to the DX10 task ID (>50) just entered. This task is now addressable by other tasks, through the communications package, using this network ID.

To accommodate the entry of more DX10 tasks, the preceding cycle of prompts is repeated. Enter the appropriate responses if more DX10 task entries are required. If other types of entries are required, proceed to the following paragraphs.

Adding a Secondary Task.

Adding a secondary task requires the entry of the following information in response to the appropriate prompts:

- The network ID of the secondary station
- The task ID for each additional task if it is a TX5 secondary station
- The network ID for each additional task if it is a TX5 secondary station

To add a TX5, TM990, or PM550 secondary task to the communications system, enter A and RETURN in response to the following prompt:

```
ADD, BUILD, CHANGE, OR DELETE (A,B,C,D):  A
```

Then enter T and RETURN in response to the next prompt:

```
ADD DX10 TASK, TX5 TASK/TM990 ID OR NEW SECONDARY (D,T,S): T
```

The next prompt is as follows:

```
TO WHICH SECONDARY (NETWORK ID):  0200
```

The response to this prompt (0200 in the example) identifies the secondary station where the additional task will be added. The next prompt is as follows:

```
TX TASK ID (0 - >FF):  >32
```

The response to this prompt (>32 in the example) identifies a TX5 task that must be added to the existing secondary station. The next prompt is as follows:

```
TX TASK NETWORK ID (0100 - 9999):  0202
```

The response to this prompt (0202 in the example) assigns a network ID to the secondary task (>32) just entered.

If the response to the TO WHICH SECONDARY (NETWORK ID) prompt is a value assigned to a TM990 or PM550 station, the following prompt appears:

```
TM990 SECONDARY NETWORK ID (0100 - 999):  1011
```

If it is a PM550 secondary station, enter RETURN. If it is a TM990 the response to this prompt (1011 in the example) assigns a network ID to the TM990 station.

Adding a Secondary Station.

Adding a secondary station requires the entry of the following information in response to the appropriate prompts:

- FCCC TILINE address
- Communications device name
- FCCC channel number
- Station drop address
- Maximum time between polls
- Station network address
- Maximum message size
- Type of station (TX5 or other)
- Network ID for the station

To add a secondary station to the network, enter A and a RETURN to the following prompt, as shown:

```
ADD, BUILD, CHANGE, OR DELETE (A,B,C,D):  A
```

Then, to specify a secondary station, enter S and a RETURN to the next prompt, as follows:

```
ADD DX10 TASK, TX5 TASK/TM990 ID OR NEW SECONDARY (D,T,S): S
```

The next prompt is the first in a series that requests data describing the secondary station. These are the same prompts used to describe an FCCC line and its associated secondary stations in paragraphs 3.3.3, 3.3.3.1, and 3.3.3.2. The descriptions are not repeated here. A different sequence is used, as shown in the following example:

```
FCCC BOARD TILINE ADDRESS: 0F900

DEVICE NAME: CM01

FCCC CHANNEL NUMBER (0 - 3): 0

STATION DROP ADDRESS (0 - >3F, CR TO CONTINUE): 40
```

```
MAX TIME BETWEEN POLLS (0-32000/1 = 250 MSEC):  100

STATION NETWORK ADDRESS (0100 - 9999):  9500

SECONDARY'S MAX MESSAGE SIZE (1 - 512):  128
```

For a non-TX5 secondary station, enter N and RETURN to the next prompt:

```
TX5 SECONDARY STATION (Y/N):  N
```

If the secondary station is a TM990, enter the appropriate ID to the following prompt (9501 in the example). If it is neither a TM990 nor a TX5, enter only RETURN:

```
TM990 SECONDARY NETWORK ID (0100 - 9999):  9501
```

NETGEN repeats the last prompt until all IDs are entered. When this is accomplished, enter only a RETURN.

Adding an FCCC Board.

Adding an FCCC board requires the entry of the following information in response to the appropriate prompts:

* FCCC TILINE address
* Communications device name
* FCCC channel number
* Channel data rate
* Internal clocking requirements
* Transmit retry requirements for the line
* The drop address for each station on the line
* The maximum time between polls for each station
* The network address of each station
* The maximum message size for each station
* The type of each station (TX5 or other)
* The network ID for each station

To add an FCCC board to the communication system (that is, to describe it to NETGEN after physical installation) enter A and RETURN to the following prompt, as shown:

```
ADD, BUILD, CHANGE OR DELETE (A,B,C,D):  A
```

Then, to specify that this modification affects one or more secondary stations enter S and RETURN to the following prompt:

```
ADD DX10 TASK, TX5 TASK/TM990 ID OR NEW SECONDARY (D,T,S): S
```

NETGEN now issues a series of prompts that request descriptions of the new line and the associated secondary stations. This series of prompts is similar to that described in paragraph 3.3.3. The sequence, with example responses, is as follows:

```
FCCC BOARD TILINE ADDRESS:  0FB00

DEVICE NAME:  CM04

FCCC CHANNEL NUMBER (0 - 3):  0

CHANNEL DATA RATE:  4800
```

```
INTERNAL CLOCKING (Y/N):  N

MESSAGE TRANSMIT RETRIES (0 - 3):  0

STATION DROP ADDRESS (0 - >3F, CR TO CONTINUE):  01

MAX TIME BETWEEN POLLS (0-32000 / 1 = 250 MSEC):  8

STATION NETWORK ADDRESS (0100 - 9999):  1400

SECONDARYS MAXIMUM MESSAGE SIZE (1 - 512 BYTES):  260
```

Enter appropriate responses to the following prompts (refer to the previous paragraph, Adding a Secondary Station):

```
TX5 SECONDARY STATION (Y/N):

TM990 SECONDARY NETWORK ID (0100 - 9999):
```

After adding the secondary station, the prompt ADD, BUILD, CHANGE OR DELETE appears. A response of B signifies that the changes can be built into the new network. A Q response terminates NETGEN and the previous configuration is preserved.

3.3.4.2 Changing an Existing Entry.

To change an existing entry in the network, enter C and RETURN to the following prompt, as shown:

```
ADD,BUILD, CHANGE, OR DELETE (A,B,C,D):  C
```

The next prompt offers the choice of changing either an FCCC channel entry, a queue buffer size, an NIT entry, or a packet time-out, as follows:

```
CHANGE CHANNEL, QUEUE SIZE, NIT ENTRY OR USER INPUT PACKET
   TIME OUT WORD (C,S,N,U):
```

The following paragraphs describe the various valid responses to this prompt.

Changing an FCCC Channel Entry.

Changing an FCCC channel entry requires the entry or confirmation of the following information in response to the appropriate prompts:

- The FCCC TILINE address of the channel
- The name of the communications device for the channel
- The channel data rate
- The internal clocking requirements for the channel
- The transmit retry requirements for the channel

If C is entered in response to the above prompt, NETGEN initiates a series of prompts as follows:

```
FCCC TILINE ADDRESS: 0F900

DEVICE NAME: CM01
```

The next three prompts display current values. If a change is required, enter the new value; if no change is required, enter only a RETURN.

```
CHANNEL DATA RATE = 9600 : 7200

INTERNAL CLOCKING = Y :

MESSAGE TRANSMIT RETRIES = 3 :  1
```

This terminates the FCCC channel changes and NETGEN again displays the following prompt:

```
ADD, BUILD, CHANGE, OR DELETE (A,B,C,D):
```

Changing a Queue Buffer Entry.

Changing a queue buffer size requires the entry of new data or the confirmation of existing data that specifies the number of buffers at levels 2 through 4 for both input and output.

Enter C and RETURN to the preceding prompt to proceed with changing a queue buffer size. The next prompt requires a response of S and RETURN, as shown:

```
CHANGE CHANNEL, QUEUE SIZE, NIT ENTRY OR USER INPUT PACKET
   TIME OUT WORD (C,S,N,U):  S
```

NETGEN responds by displaying the current buffer status. If a change is required, enter the new value (in decimal). Otherwise, enter only RETURN. The sum of the changes cannot exceed 86 (decimal). An example follows:

```
LEVEL 2 INPUT  BUFFERS = 22 :  20
LEVEL 2 OUTPUT BUFFERS = 22 :  20
LEVEL 3 INPUT  BUFFERS = 14 :  16
LEVEL 3 OUTPUT BUFFERS = 14 :  16
LEVEL 4 INPUT  BUFFERS = 14 :
```

Queue size values should be changed only if the statistics program shows a high queue full count. Refer to Section 5 for information on the statistics program.

NETGEN now issues the following prompt:

```
ADD, BUILD, CHANGE, OR DELETE (A,B,C,D):
```

Changing a Network Entry.

Changing a network entry requires the entry or confirmation of the following information in response to the appropriate prompts:

- The network ID to be changed
- The parameters of the entity associated with that network ID

To change a network entry enter C and RETURN in response to the preceding prompt. Then enter N and RETURN in response to the following prompt:

```
CHANGE CHANNEL, QUEUE SIZE, NIT ENTRY OR USER INPUT PACKET
   TIME OUT WORD (C,S,N,U):  N
```

Next enter the network ID that needs to be changed (1100 in the following example):

```
NETWORK ID TO CHANGE: 1100
```

NETGEN responds with a procedural message indicating that the value entered in response to the previous prompt is an output entry, that is, not a DX10 task, but a secondary station or task:

```
ID 1100 IS AN OUTPUT ENTRY
```

An output entry causes the following prompts to be displayed. To change any of the values associated with the network entry, enter the new value after the colon. The prompts are displayed one at a time, and in this example, they refer to the characteristics of the TX5 secondary station with a network ID of 1100. Enter a RETURN to retain a current value.

```
  MAXIMUM MESSAGE SIZE =  256 :  <return>
   STATION DROP ADDRESS =  >01 :  <return>
MAX TIME BETWEEN POLLS =    8 :  100      (* Changes delay)
  VIRTUAL CHANNEL GROUP =    0 :  <return>
 VIRTUAL CHANNEL NUMBER =    1 :  <return>
        ID 1100 TX TASK =  >27 :  <return>
        ID 1101 TX TASK =  >28 :  <return>
        ID 1102 TX TASK =  >10 :  >35      (* Changes task ID)
```

The preceding changes increase the time between polls from 2 seconds to 25 seconds and assign the network ID of task >35 to 1102.

To change an input entry respond with C to the ADD, BUILD, CHANGE OR DELETE prompt, and respond with N to the following prompt:

```
CHANGE CHANNEL, QUEUE SIZE, NIT ENTRY OR USER INPUT PACKET
   TIME OUT WORD (C,S,N,U):  N
```

Enter an input value to the following prompt:

```
NETWORK ID TO CHANGE: 1101
```

NETGEN responds with a procedural message confirming that the value entered is an input entry, that is, a DX10 or a TX5 task, or a TM990 entry.

```
ID 1101 IS AN INPUT ENTRY
```

For a DX10 or TX5 entry the following prompt is displayed next:

```
TASK ID = >28 :
```

For a TM990 entry the following prompt is displayed:

```
TM990 SECONDARY ID = 1200 :
```

In either case, enter the new network ID and RETURN. NETGEN then displays the ADD, BUILD, CHANGE, OR DELETE prompt.

Changing a Packet Time-Out.

The packet time-out count is used to prevent buffer deadlock in the primary station. Each input message is timed to ensure that it does not remain buffered for longer than 10 seconds. After 10 seconds, the message is erased and an error message is written to the system log. To modify this time-out, respond with C and RETURN to the following prompt, as shown:

```
ADD, BUILD, CHANGE, OR DELETE (A,B,C,D):  C
```

Next, enter U and RETURN to the following prompt:

```
CHANGE CHANNEL, QUEUE SIZE, NIT ENTRY OR USER INPUT PACKET
   PACKET TIME OUT WORD (C,S,N,U):  U
```

Enter the new value (5 seconds) in response to the following prompt, which shows the current time-out value (10 seconds):

```
USER INPUT PACKET TIME OUT WORD = 10 : 5
```

This response changes the time allowed to retrieve input messages from 10 seconds to 5 seconds.

3.3.4.3 Deleting from an Existing Configuration.

The following example shows how to delete an existing network ID. The ID may be assigned to a DX10 task, a TX5 task or a secondary station. If a DX10 task network ID is designated, that ID is deleted from the appropriate primary data structure. If a TX5 task is designated, that ID is deleted from both the secondary and the appropriate primary data structures. If a secondary station is designated, the DX10 structure associated with the secondary station is deleted as well as the file for that secondary station (that is, .INDSCOMM.NETGEN.SECnnnn).

An example of a deletion follows:

```
  ADD,BUILD, CHANGE, OR DELETE (A,B,C,D): D

  NETWORK ID TO DELETE: 1100

  NETWORK ID 1100 (SECONDARY STATION) DELETED
```

The last message confirms that the designated ID was deleted.

The deletion is now complete. Two final prompts offer the option of including all changes entered in the network configuration. To accomplish this, enter B in response to the first prompt and Y in response to the second prompt, as follows:

```
  ADD,BUILD, CHANGE, OR DELETE (A,B,C,D):  B

  DO YOU WANT TO SAVE CONFIGURATION (Y/N):Y
```

An N response to the last prompt causes NETGEN to terminate and to discard all changes just entered. The files required to generate the system are built but the existing configuration is not updated. Entering a Q to any prompt terminates NETGEN and retains the existing configuration.

3.4 TX5 SECONDARY SYSTEM

A correctly functioning TX5 operating system is required to operate the communications package at the secondary station. Also, a local-line module (LLM) must be prepared and installed in the secondary station. The following paragraphs describe the steps required to build the secondary communications system and to install the LLM board.

3.4.1 Building the Secondary System

The TX5 system is first constructed at the DX10 primary station and then downloaded to the secondary station. The following utilities are available at the primary station for this purpose:

* The network generator (NETGEN) that builds the communicaton data structures used by the TX5 communications package software.

* The system generator that builds the TX5 operating system

* The download preprocessor that reformats the TX5 operating system object format into a binary object format suitable for downloading

* The downloader (DOWNLOAD) that transmits the preprocessed object to the appropriate TX5 secondary station

The major steps required to build a TX5 secondary system are as follows:

1. Create the data structures required by the secondary station. This is accomplished by activating the NETGEN program, which simultaneously creates data structures for the primary station and each secondary station.

2. Assemble these data structures.

3. Perform the TX5 system generation.

4. Assemble the TXDATA and TASKDF files that are created during TX5 system generation.

5. Create and/or edit a TX5 link control file. The link control file must include the name of the assembled TXDATA and TASKDF files, the filename of the user's assembled data structures built by NETGEN, the HDLC communications package programs, and the applications programs required to customize the user's TX5 system. The remote SCI program is optional. If this utility is required, it must be built at this time (see the following paragraph).

6. Perform the link edit. The link edit process produces the TX5 operating system. This object file must be preprocessed before downloading.

7. Preprocess the TX5 operating system (OS) object file. After the TX5 has been linked, it must be preprocessed and entered into the DX10 download directory. This completes the process of constructing the TX5 system. The resulting TX5 OS can be downloaded if the LLM board has been installed at the secondary station (see paragraph 3.4.3 for information on installing the LLM board).

8. Download the TX5 OS. After preprocessing, the TX5 operating system can be downloaded to the appropriate secondary station using the downloader utility. However, refer to paragraph 3.5.2 for further information on downloading.

If remote SCI is included in the TX5 system, the secondary software can be tested with that facility after the download is complete. Some simple remote SCI function, such as List Memory (LM) or Show Task Status (STS), can be performed to determine if the secondary station is operating. The use of remote SCI is suggested as a simple technique to test the basic operation of the communications package at the secondary station.

3.4.2 Building Remote SCI for a TX5 Secondary Station

Remote SCI must be generated for each TX5 secondary station before the TX5 link edit can be performed. A remote SCI generator program enables the user to select the SCI commands required for each secondary station and to generate a remote SCI program for the secondary station according to those requirements. The remote SCI generator is activated by entering the DX10 SCI command LREMSCI at a DX10 VDT.

When the command LREMSCI is entered, a series of prompts is displayed on the VDT. For each command required for inclusion in remote SCI, the user must respond to the corresponding prompt by accepting the default YES. A NO response to any prompt excludes the function associated with the prompt. For example, a NO response to READ/WRITE CRU will exclude the CRU functions RCRU and WCRU from the remote SCI program. Attempting to use these functions results in an error message display at the primary station VDT. When the desired commands have been selected, the user enters the path name <control> of a file that will store the link edit control stream used to link the selected remote SCI module.

Two successive displays are required to present all options to the user after the command LREMSCI has been entered. In the following example, all commands listed except MODIFY MEMORY and READ/WRITE CRU are accepted.

The first display is as follows:

```
[] LREMSCI

     LINK REMOTE SCI COMMANDS FOR A SECONDARY
                   LIST MEMORY: YES
                 MODIFY MEMORY: NO
              SHOW TASK STATUS: YES
                  EXECUTE TASK: YES
                     KILL TASK: YES
           ASSIGN/RELEASE LUNO: YES
                READ/WRITE CRU: NO
```

After the READ/WRITE CRU prompt is answered, DX10 SCI clears the screen and displays the second set of prompts as follows:

```
     LINK REMOTE SCI COMMANDS FOR A SECONDARY
                         TRACE: YES
           BREAKPOINT COMMANDS: YES
               SHOW I/O STATUS: YES
                DUMP WORKSPACE: YES
        DATE AND TIME COMMANDS: YES
                  INSTALL TASK: YES
            RETURN NETWORK IDS: YES
      CONTROL FILE ACCESS NAME: <control>
```

After a filename is entered in response to the CONTROL FILE ACCESS NAME prompt, a link edit control stream is created on this pathname. In the following example of a control stream, all remote SCI commands are selected:

```
PARTIAL
PHASE 0,REMSCI
INCLUDE O.MAIN
INCLUDE O.LM
INCLUDE O.MM
INCLUDE O.STS
INCLUDE O.XT
INCLUDE O.KT
INCLUDE O.LUNO
INCLUDE O.CRU
INCLUDE O.TRACE
INCLUDE O.BRKPNT
INCLUDE O.SIS
INCLUDE O.SWR
INCLUDE O.TIME
INCLUDE O.LOADER
INCLUDE O.NID
NOTGLOBAL
GLOBAL REMSCI
END
```

If MODIFY MEMORY and READ/WRITE CRU were not selected, as in the previous example, the statements INCLUDE O.MM and INCLUDE O.CRU would be omitted.

After link control file creation, the TX5 Remote SCI program is ready to be linked. Prior to entering the SCI Execute Linkage Editor (XLE) command, the synonym O must be assigned using the SCI Assign Synonym (AS) command as follows:

```
AS S=O, V=.INDSCOMM.TXOBJ
```

After the synonym is assigned, the remote SCI can be linked by entering the XLE command as follows:

```
[] XLE
   EXECUTE LINKAGE EDITOR

          CONTROL ACCESS NAME:  <control>
   LINKED OUTPUT ACCESS NAME:  <linkfile>
          LISTING ACCESS NAME:  <listing>
                  PRINT WIDTH:  80
```

where:

<control>	is the filename specified in response to the LREMSCI command previously executed.
<linkfile>	is the name of the file that must be included in the TX5 link control stream. The same remote SCI object file can be used for all TX5 secondary links.
<listing>	is the name of the file to which the messages generated during the linking process are written.

The default print width value, 80, should be taken for a standard listing file or device.

It is possible to include both the operator communication package (OCP) and remote SCI (REMSCI) in a TX5 secondary station. If REMSCI is required but OCP is not, include the REMSCI log task and the dummy log DSR. In this case, the TX5 warm-start and diagnostic messages are directed to the DX10 system log.

If both REMSCI and OCP are required, do not include the REMSCI log task or the dummy log DSR. In this case, the warm-start and diagnostic messages are directed to the TX5 log.

3.4.3 Installing the LLM Board in the 990/5 Computer

The LLM switches and jumpers must be set correctly and the LLM card must be installed at CRU address >20 in the TX5 secondary station before communications can be established. Additionally, a suitable interrupt level for the board must be selected. This interrupt level must be correctly wired and correctly specified during TX5 system generation. The priority level selected should be sufficiently high to permit the communications software access to the LLM card when an interrupt is pending. Any preemption of the secondary station's communications facilities result in the primary station aborting communications after a specified time and termination of the request to the secondary station. User programs should not be assigned priority level 0 and any excessively busy CRU device should not be assigned an interrupt level above the LLM card.

To select the LLM address, set switches S1-1 through S6-1 to the binary equivalent of the complement of that address, as shown in the following:

SWITCH S1						
S1-1	S1-2	S1-3	S1-4	S1-5	S1-6	ADDRESS
ON	ON	ON	ON	ON	ON	0
ON	ON	ON	ON	ON	OFF	1
ON	ON	ON	ON	OFF	ON	2
ON	ON	ON	ON	OFF	OFF	3
•	•	•	•	•	•	•
OFF	ON	ON	ON	ON	ON	32
•	•	•	•	•	•	•
OFF	OFF	OFF	OFF	OFF	OFF	63

2280129

Switch S2 is a dual-purpose switch; it controls the enabling of the LLM address decoding feature (S2-4) and also selects the LLM data rate (S2-1 through S2-3). These two functions are completely independent of each other; the setting of S2-4 has no effect on S2-1 through S2-3, and vice versa.

The selected data rate must correspond exactly to the data rate selected for the primary station. S2 settings for various data rates are as follows:

SWITCH S2				DATA RATE IN BITS PER SECOND
S2-1	S2-2	S2-3	S2-4	
ON	ON	ON	OFF	1200
ON	ON	OFF	OFF	1800
ON	OFF	ON	OFF	2400
ON	OFF	OFF	OFF	3600
OFF	ON	ON	OFF	4800
OFF	ON	OFF	OFF	7200
OFF	OFF	ON	OFF	9600

2280130

There are also several jumpers that must be positioned correctly. These jumpers set the line termination condition and select the external interface for either local-line or EIA mode.

Jumper positions are numbered J1 through J12. J1 through J6 select an appropriate line termination and J7 through J9 select the line interface type. J10 through J12 are preset at the factory and should not be changed by the user. The following gives brief instructions on setting J1 through J9. Refer to the Model 990 Computer Local Line Module Operation and Maintenance Manual for further information.

Condition or Requirement	Directions
Termination on the LLM	Install jumper between locations J1 and J3, and between locations J2 and J4.
No termination on the LLM	Install jumper between locations J3 and J5, and between J4 and J6.
Default termination	If no jumpers are installed in J1 through J6, the module is not terminated.
Local-line interface	Install jumper between locations J7 and J8.
EIA interface	Install jumper between locations J8 and J9.
Default interface	If no jumpers are installed in locations J7 through J9, the local-line interface is selected.

3.4.4 Assembling the TX5 Files

When network generation is completed, the files for the TX5 secondary stations are ready to be assembled and linked with the TX5 systems. (The files created for the primary station software will be automatically assembled and linked when the SCI command COMMGEN is entered, see paragraph 3.5.)

Assemble the TX5 secondary station file using the SCI command Execute Macro Assembler (XMA). The user must specify the object, list and error access names as follows:

```
[] XMA
    EXECUTE MACRO ASSEMBLER
           SOURCE ACCESS NAME: .INDSCOMM.NETGEN.SECnnnn
           OBJECT ACCESS NAME: .<user-specified filename>
          LISTING ACCESS NAME: .<user-specified filename>
            ERROR ACCESS NAME:
                      OPTIONS:
        MACRO LIBRARY PATHNAME:
                  PRINT WIDTH: 80
                 PRINT LENGTH: 80
```

The object file specified must then be included in the TX5 link edit control stream. The link control file must be edited to include this object file or the secondary station communications software will not function correctly.

3.4.5 TX5 System Generation

The TX5 secondary operating system can be generated on a DX10 system by executing the GENTX5 utility. Refer to the TX5 Operating System Programmer's Guide to become familiar with TX5 system generation.

Before beginning system generation, the user should be prepared to include the appropriate tasks in the system. Determine task labels, initial states, and priorities before starting. For a secondary system that is to be downloaded from the primary station, it is preferable to include user tasks with the TX5 system at this time and link those tasks with the TX5 system as resident tasks. These tasks are then preprocessed and downloaded along with the TX5 operating system. For further information and examples of TX5 system generation, refer to Appendix C.

3.5 ACTIVATING THE COMMUNICATIONS NETWORK

To activate the communications network successfully, perform the following steps:

1. Assemble and link the primary station software created during network generation using the COMMGEN utility.

   ```
   [] COMMGEN
      COMM GENERATION
        BATCH LISTING: <device or file>
        WAIT?: YES
   ```

   Respond to the prompts as follows:

   - * BATCH LISTING. Enter the name of the device or file where the results of the batch stream execution will be written. It is recommended that the batch listing be retained for later reference.

   - * WAIT? Accept the default YES for foreground execution of the COMMGEN batch stream. Respond NO for background operation to allow other terminal usage in the foreground mode.

   Entering the SCI command COMMGEN at a DX10 VDT activates a batch stream that performs the required assembly and linking functions.

2. Perform a normal DX10 warm-start by pressing the HALT, RESET, and LOAD controls on the 990 computer console in that order.

3. Activate the HDLC communications package using the COMMGO utility. Entering the SCI command COMMGO at a DX10 VDT initiates a communications warm-start. This step is further described in the next paragraph.

4. Download the secondary stations either automatically or using the DOWNLOAD utility (see paragraphs 3.5.2 and 5.3 respectively).

Once these procedures have been performed, applications programs can utilize the communications package for task-to-task communications.

3.5.1 Activating the Primary Station Communications Software

A communications warm-start program is executed to activate the communications package software. This warm-start is initiated by entering the SCI command COMMGO. The warm-start program downloads HDLC software to all FCCC cards in the system that use the HDLC package. For each FCCC card downloaded a message in the following format is written to the system log:

```
(date and time) COMMDWNL(CMxx, nn.mmm), L=yyyy,
                P=<program filename>
```

where:

COMMDWNL	is an acronym for an FCCC download.
CMxx	is the communication device name and number.
nn.mmm	is the revision date of the FCCC ROM.
L	equates to the address for the downloaded code.
P	equates to the pathname of the HDLC program file.

An example of the message is as follows:

```
239:1128+       COMMDWNL(CM05, 80.354),L=8A96,
                P=.INDSCOMM.PROGRAM
```

When all FCCC cards have been loaded, the following message appears:

```
(date and time) COMMUNICATION WARMSTART
```

If the communications package cannot be activated, an error message is displayed on the VDT. Reasons for failure to activate include the following:

- The communications package is already activated.
- One or more errors occurred during COMMGEN execution, that is, during assembly and linkage of the results of network generation to the communications software. Inspect the listing file resulting from COMMGEN execution to ascertain the cause of the errors. Take appropriate corrective action and re-attempt to execute COMMGEN. See Table 3-1 for listing file information.

Ensure that all cables are properly connected to the FCCC boards before executing the COMMGO command. Without correct cable connection, the warm-start procedure will not complete. No error message is generated for this condition. Any other errors that occur during warm-start are indicated by error messages written to the system log. See Appendix A for further information on warm-start error messages.

After the warm-start program has completed its functions, the tasks within the communications package are enabled, and communications can commence. The warm-start program writes the following completion message to the system log:

```
(date and time) INITIALIZATION COMPLETE
```

3.5.2 Downloading the Preprocessed Files

After the warm-start at the primary station is complete, polling of the secondary stations begins. Any secondary that is not initialized and ready to operate (that is, not previously loaded or downloaded) should return a download request to the primary station from its ROM loader (if the HDLC communications ROM loader is installed in that secondary station). When this occurs, the download process to that secondary station automatically begins and a download message is printed on the system log. The download data is read from the download files at the primary station.

A download can also be started at the primary station by entering the SCI command DOWNLOAD at a primary terminal and specifying the network ID and option number for the secondary station to be downloaded. The download process is terminated when the last download record is sent and acknowledged. A download complete message is then printed on the system log. The downloaded secondary station is identified by its network ID.

If a downloaded TX secondary station includes the remote SCI log task, a TX warm-start message is printed on the system log after the download complete message is printed. The warm-start message prints the network ID assigned to the log task during TX5 system generation.

NOTE

> If remote SCI (REMSCI) is included in the downloaded secondary software, the secondary station can be tested using remote SCI. This is suggested as a limited test to ascertain only that the secondary station is operational, able to communicate, and contains the (system) generated tasks and LUNOs that were specified during system generation.

Section 4

Applications Programming

4.1 GENERAL

This section provides the information necessary to write applications programs that utilize the DX10 HDLC Communications Package. Applications programs may be written in assembly language, FORTRAN, or Pascal, for execution under DX10, release 3.4, or TX5, release 3.2.0.

The interface between applications programs and the communications package is provided by DX10 and TX5 supervisor calls (SVCs) that perform the following functions:

* Write (send/transmit) data

* Read (receive) input data

* Request activation services

In order to use the communications package effectively, the applications programmer must be familiar with the following:

* The network configuration

* The network ID of each station and applications program that sends or receives information

* The output of the network generator program, NETGEN, particularly:

  - The maximum buffer size of each station

  - The assignment of task IDs

The following paragraphs discuss the SVCs, and describe the input/output operations and assembly language requests for activation services. Following this are examples of applications programs resident in different computers in the same network (that is, in communication with each other).

4.2 SUPERVISOR CALLS

Applications programs interface with the communications package by issuing SVCs that request the communications package to perform specific functions. SVCs are included in the operating system to perform I/O, task control, service functions, memory control, and file management functions. SVCs are explicitly used in assembly language tasks, whereas higher-level language statements are processed by the interpreter or compiler and are translated to the particular call required to perform the requested operation. If a statement to access a particular SVC is not available in the higher level language, it is possible to write an assembly language routine to access the SVC. Examples of assembly language routines that perform SVCs to the communications package in FORTRAN or Pascal programs are provided in this section.

SVCs are implemented using the Model 990 Computer assembly language extended operation (XOP) instruction. The XOP number is level 15, and the effective address of the XOP instruction is a SVC block address. Execution of the XOP instruction passes control to the operating system. The supervisor call block referenced in the XOP instruction contains the parameters necessary for the requested operation to be performed by the SVC >4D processor. For more detailed information on the XOP instruction, refer to the Model 990 Computer 990/10 and 990/12 Assembly Language Reference Manual.

Six SVCs can be accessed from applications programs for communications purposes. The functions they perform and the methods by which they are accessed are as follows:

Function	Assembly Language Access	FORTRAN or Pascal Access
Write output data	SVC >4D with an I/O opcode of 2	Call to subroutine WRIT4D
Read input data	SVC >4D with an I/O opcode of 3	Call to subroutine READ4D
Request activation services	SVC >4D with I/O opcodes of 4 through 7	Call to subroutine WKUP4D

4.2.1 Input/Output Calls

Assembly language programs can perform SVCs by setting up SVC call blocks and executing level 15 XOP instructions using those call blocks. The call blocks pass the necessary parameters to the SVC >4D processor.

To execute the communications SVCs, byte 0 of each call block must contain the opcode >4D. The I/O opcode in byte 2 of the call block specifies the operation that is to be performed. For normal read/write operations, the I/O opcodes that may be used by applications programs are 2 or 3. For requests for activation services, the I/O opcode ranges from 4 through 7.

Applications programs should normally use the input and output calls for the following purposes:

* Executing write calls that transmit data to the destination program

* If a response from the destination program is anticipated, requesting activation services and then executing a time delay SVC while waiting for the response

* When reactivated, executing a read call to retrieve any input data addressed to it

The following paragraphs define the format of an SVC block for executing input/output operations.

4.2.1.1 Input/Output Call Block.

The format of the SVC block for an input/output call is shown in Figure 4-1. The data contained in the call block originates as follows:

* Data supplied by the applications program:

    - SVC >4D

    - I/O opcode

    - I/O data buffer pointers

    - Input and output character counts

    - Destination ID (DID), for write operations

    - Source ID (SID) when not set to zero by applications program

- Data returned by the SVC processor:
  - Status/error code
  - System flags
  - DID, for read operations
  - SID, when set to zero by the applications program

BYTE 0	>4D		STATUS CODE	
BYTE 2	I/O OPCODE		RESERVED	
BYTE 4	SYSTEM FLAGS		USER FLAGS	
BYTE 6	I/O DATA BUFFER POINTERS			
BYTE 8	INPUT CHARACTER COUNT			
BYTE 10	OUTPUT CHARACTER COUNT			
BYTE 12	RESERVED			
BYTE 14	DID	DID	DID	DID
BYTE 16	SID	SID	SID	SID

2280121

Figure 4-1 Supervisor Call Block for Input/Output Calls

The contents of the SVC block shown in Figure 4-1 are as follows:

Byte 0 contains the opcode >4D for the SVC.

Byte 1 contains the status/error code returned by the SVC. The applications program should check the code and take appropriate action. The codes and their meanings are as follows:

Code	Meaning
00	Operation requested has been accepted by the communications package.
01	I/O opcode function specified in byte 2 has not been implemented.

02	Illegal I/O opcode in byte 2, that is, not within the range of 2 through 7.
03	Invalid source identification (SID) in bytes 16 and 17.
04	When executing a read call, no input data has been received for the program to process.
05	When executing a write call either: * no buffer is available in the system to accept the output data, or * no buffer is currently available to accept the request for activation services.
06	The communications package is not active.
07	An applications program attempted to execute a privileged operation by requesting an I/O opcode greater than 7.
08	Invalid destination identification (DID) in bytes 14 and 15.
09	When executing a write call, output buffer length is too large. When executing a read call, input buffer length is too small.
0A	The DID specified is inoperative, that is, the station is not active.
0B	When executing a write call, no buffer is available to accept the output data for the station.

Byte 2 contains the I/O opcode supplied by the applications program. For normal read or write operations, the I/O opcodes are 2 and 3; for requests from activation services, I/O opcodes range from 4 through 7. I/O opcodes greater than 7 are privileged codes reserved for use by the communications package software.

The I/O opcodes available to applications programs and the functions performed are as follows:

Code	Function Performed
00	Reserved - no call. Applications should not use this code.
01	Reserved - no call. Applications should not use this code.
02	Write/send data. Causes data pointed to by bytes 6 and 7 to be moved to the communications output buffer and sent to the DID identified in bytes 14 and 15.
03	Read input data. If input data is found, the data is moved from the communications input buffer and stored beginning at the address pointed to by bytes 6 and 7.
04 - 07	Activation services functions. (For activation services calls, the SVC block consists only of bytes 0 through 3; the remainder of the call block is not required for these calls. See paragraph 4.2.2.1)

Byte 3 is reserved.

Byte 4 contains system flags that are defined as follows:

Bit 0	Not used.
Bit 1	0= No errors occurred. 1= Error occurred on the call.
Bits 2-7	Not used.

Byte 5 contains user flags that are reserved for future use.

Bytes 6 and 7 contain the starting address of the applications program's data buffer. For read operations, this is the starting address where input data is stored.

Bytes 8 and 9 contain the logical record length of the applications program's input buffer for read operations. These bytes specify the maximum number of input characters (each character is one byte) that may be stored in the input buffer pointed to by bytes 6 and 7.

Bytes 10 and 11 contain the applications program's output character count for write operations. This is the number of bytes (characters) to be transferred from the application program's data buffer to the communications buffer. For read operations, the number of bytes sent to the application program's data buffer is returned to this location.

Bytes 12 and 13 are reserved for the communications package.

Bytes 14 and 15 contain the DID of the station or program for write calls to which output data is addressed. This word is formatted in four half-byte binary coded decimal (BCD) values. Applications programs that send data to secondary programs or remote devices must furnish the DID for the data to be routed to the proper location. On read calls, the SVC returns the network ID of the remote sender in these bytes. This word is therefore always the network ID of the remote station or program.

Bytes 16 and 17 contain the SID of the calling station or program. This word is formatted in four half-byte BCD values. The SVC supplies the SID if the applications program makes the call with zero (0) in these bytes. These bytes always identify the caller's SID, whether the call is a read or write/send call.

Note that available network IDs in bytes 14 through 17 are expressed as BCD numbers in the range of 0100 through 9999. However, they must be supplied in applications programs in hexadecimal format. For example, a network ID of 0431 would be entered in the program as >0431. The characters A through F are invalid for all network IDs. The IDs 0000 to 0099 are reserved for the communications package software.

The following paragraphs describe each of the input/output operations.

4.2.1.2 Write/Send Data - Opcode 2.

An SVC >4D, I/O opcode 2, transfers one block of data in the applications program at the source from its output data buffer to the communications output buffer for subsequent transmission to the DID. The DID may be an applications program or a remote device. Control returns to the applications program when the data transfer to the communications output buffer is complete.

No provisions are made for automatic response from the DID. Applications programs may establish their own method for validating received and/or transmitted data. For example, a DID may send a message back to the SID to validate received data. The DID word always identifies a remote program or device.

The output communications buffers are queued chronologically within the communications system and for the addressed secondary station.

4.2.1.3 Read Input Data - Opcode 3.

An SVC >4D, I/O opcode 3, transfers one block of data from a communications input buffer to the applications program´s input data buffer. The applications program may request activation services to activate it when input data is addressed to it, then request the operating system to suspend it (either by a time delay, suspension, or termination until further notice). When reactivated, the program may retrieve the input data with a read call. One read call retrieves one block of input data; retrieving multiple blocks of input data requires multiple calls.

The DID word always identifies the sender of the data.

The input data blocks are returned to the caller from a chronologically ordered queue of input messages. The caller receives the oldest input message on that queue when the read call is made.

4.2.2 Activation Services Calls

Activation services are for use by applications programs that expect to receive input data from other programs or remote devices. The applications program that expects to receive the data utilizes activation services in the following sequence:

1. The applications program requests the desired type of service from activation services.

2. The applications program executes a call to the operating system to terminate, to suspend, or to begin a time delay.

3. After the operating system deactivates the program, activation services begins a search of the input buffers for data addressed to the program.

4. When input data addressed to the applications program is located, activation services reactivates the program.

5. The applications program executes a read call (I/O opcode 3) to retrieve the data addressed to it. If a time delay has been requested, an error on the read call could indicate that activation services did not activate the program, but that the operating system did at the expiration of the time delay.

To request activation services, the applications program must execute the appropriate SVC. Byte 0 of the call block must contain the opcode >4D. The I/O opcode in byte 2 of the call block specifies the operation that activation services is to perform. The I/O opcodes for requesting activation services range from 4 through 7.

After the activation services request is made, the applications program should make the appropriate inactive state request by executing one of the following SVCs:

* Time Delay SVC, if the request for activation services is to be activated from a time delay. A delay request of 5 to 15 seconds (depending on the number of attached secondary stations and the amount of traffic flow in the network) should be made.

* Suspend SVC, if the request for activation services is to be unsuspended.

* Terminate SVC, if the request for activation services is to be bid.

4.2.2.1 Activation Services Call Block.

For activation services calls, the SVC block consists only of bytes 0 through 3; the remainder of the call block is not required for these calls. The format for the call block is shown in Figure 4-2. The data contained in the call block originates as follows:

* Data supplied by the applications program:
  - SVC >4D
  - I/O opcode
* Data returned by the SVC processor:
  - Status/error code

BYTE 0	>4D	STATUS CODE
BYTE 2	I/O OPCODE	RESERVED

2280122

Figure 4-2 Supervisor Call Block for Requesting Activation Services

The contents of the SVC block shown in Figure 4-2 are as follows:

Byte 0 contains the opcode >4D for the SVC.

Byte 1 contains the status code returned by the SVC. The applications program should check the status code returned and take appropriate action. The codes and their meanings are as follows:

Code	Meaning
00	Activation services has accepted the call.
01	Illegal I/O opcode.
02	Not used.
03	Invalid task ID. The task requesting activation services must have a network ID assigned to it by the network generator program NETGEN.
04	Not used.
05	System buffers are full; unable to accept the request.
06	Communications system inactive. In the primary station, the request is not accepted. However, in a TX5 secondary this code is ignored and the request is accepted.

Byte 2 contains the I/O opcode supplied by the applications program. Explanations of I/O opcodes used to make requests of activation services are as follows:

Code	Function
04	Request is made for the applications program to be activated from an indefinite suspension when input data from any station is addressed to it. DX10 applications programs that request this service can be installed on any program file.
05	Request is made for the applications program to be activated from a time delay when input data from any station is addressed to it. Applications programs that request this service can be installed on any program file in the DX10 system.
06	Request is made for the applications program to be bid when input data from any station is addressed to it. Applications programs that request activation from termination, in that they request to be bid, must be installed in a DX10 program file named .INDSCOMM.USERPROG and global LUNO >40 must be assigned to the file. The bid task service in TX5 is supported for resident tasks only.
07	Request is made to remove the applications program from the activation services list.

Byte 3 is reserved for use by the communications package.

The applications programs can use the activation services calls as described in the following paragraphs.

4.2.2.2 Request Activation from Suspension - Opcode 4.

When input data addressed to the applications program is anticipated on a regular yet unscheduled basis, the applications program should execute an SVC with opcode >4D, I/O opcode 4. After requesting this of activation services, the program should request the operating system to suspend it indefinitely. When any input data is addressed to it, activation services unsuspends the program. After being unsuspended, control returns to the program and it must execute a read call to retrieve the input data addressed to it.

4.2.2.3 Request Activation from a Time Delay - Opcode 5.

When an applications program anticipates a response from a destination program that validates the receipt of previously sent data, the applications program should execute an SVC with opcode >4D, I/O opcode 5. After requesting this of activation services, the program executes a call to the operating system for a time delay. When reactivated, the program must execute a read call to retrieve any input data addressed to it. If no data is retrieved, the program knows the time delay has expired and takes appropriate action. When input data addressed to the program is received, activation services activates the program before the time delay expires.

4.2.2.4 Request Activation from Termination - Opcode 6.

When input data addressed to an applications program is anticipated on an unscheduled basis, the applications program should execute an SVC with opcode >4D, I/O opcode 6. After requesting this of activation services, the program executes a call to the operating system for termination (in that it requests to be bid). When any input data is addressed to the program, activation services bids the program. After being activated, control returns to the program and it must execute a read call to retrieve the input data addressed to it.

4.2.2.5 Request Removal from Activation Services - Opcode 7.

When an applications program has been reactivated from a time delay as a result of time expiration (in that no input data has been addressed to it), the program should execute an SVC with opcode >4D, I/O opcode 7. This call removes the program from the activation services list.

Activation services replaces the latest request made to it with one that may have been made previously. For example, if an applications program requested activation from a time delay and the time delay expired, as is indicated when no data is available when the read call is made, the program assumes that the activation services request is still active. If an alternate action includes termination, the program can make a second request (to be activated from termination) without requesting removal from the activation services list.

4.3 PROGRAMMING EXAMPLES AND APPLICATIONS NOTES

The following example illustrates how two applications programs communicate with each other using the communications SVCs. Figure 4-3 illustrates the flow of data between applications programs in a primary station and a secondary station when using the communications package. Applications program B executes in the secondary station, while applications program A executes in the primary station. The communications between the two programs are outlined in steps 1 through 16 as follows:

1. Applications program B is activated in the secondary station at warm start.
2. Program B requests activation from suspension by calling activation services with an I/O opcode 4.
3. Program B executes a call to the operating system for indefinite suspension. (Activation services begins searching the communications input buffers for data addressed to program B.)
4. Applications program A at the primary station is activated.
5. Program A processes and prepares the data for subsequent use by program B.
6. Program A executes a call to write/send data to program B with an I/O opcode 2.
7. Program A requests activation from a time delay by calling activation services with an I/O opcode 5.
8. Program A executes a call to the operating system for a 20-second time delay. (Activation services begins searching the communications input buffers for data addressed to program A.)
9. When the data addressed to program B arrives at the secondary station, activation services activates program B.
10. Program B executes a call to read the input data addressed to it with an I/O opcode 3.
11. Program B processes the data it receives and prepares to send a response to program A acknowledging the receipt of data.

12. Program B executes a call to write/send data to program A with an I/O opcode 2.

13. Program B begins routine processing.

14. When the data addressed to program A arrives at the primary station, activation services activates the program from the time delay.

15. Program A executes a call to read the input data addressed to it with an I/O opcode 3.

16. Program A processes the data received and verifies that program B received all the data addressed to it.

If program A had executed the read call in step 15 and found no input data addressed to it, program A would know that the time delay had expired. It would then execute a call to activation services with an I/O opcode 7 in order to remove itself from the activation services list. The program might then return to step 2 to attempt to communicate with program B a second time, or it might take some alternate action.

If after step 16, program A anticipated other, unscheduled communications with program B, it would execute a call to activation services to activate it from termination (that is, it would request to be bid). The program would then request the operating system to terminate it. (Program A must be installed in the program file named INDSCOMM.USERPROG and assigned to global LUNO >40.) When input data addressed to program A arrives at the primary station, activation services bids program A for further processing.

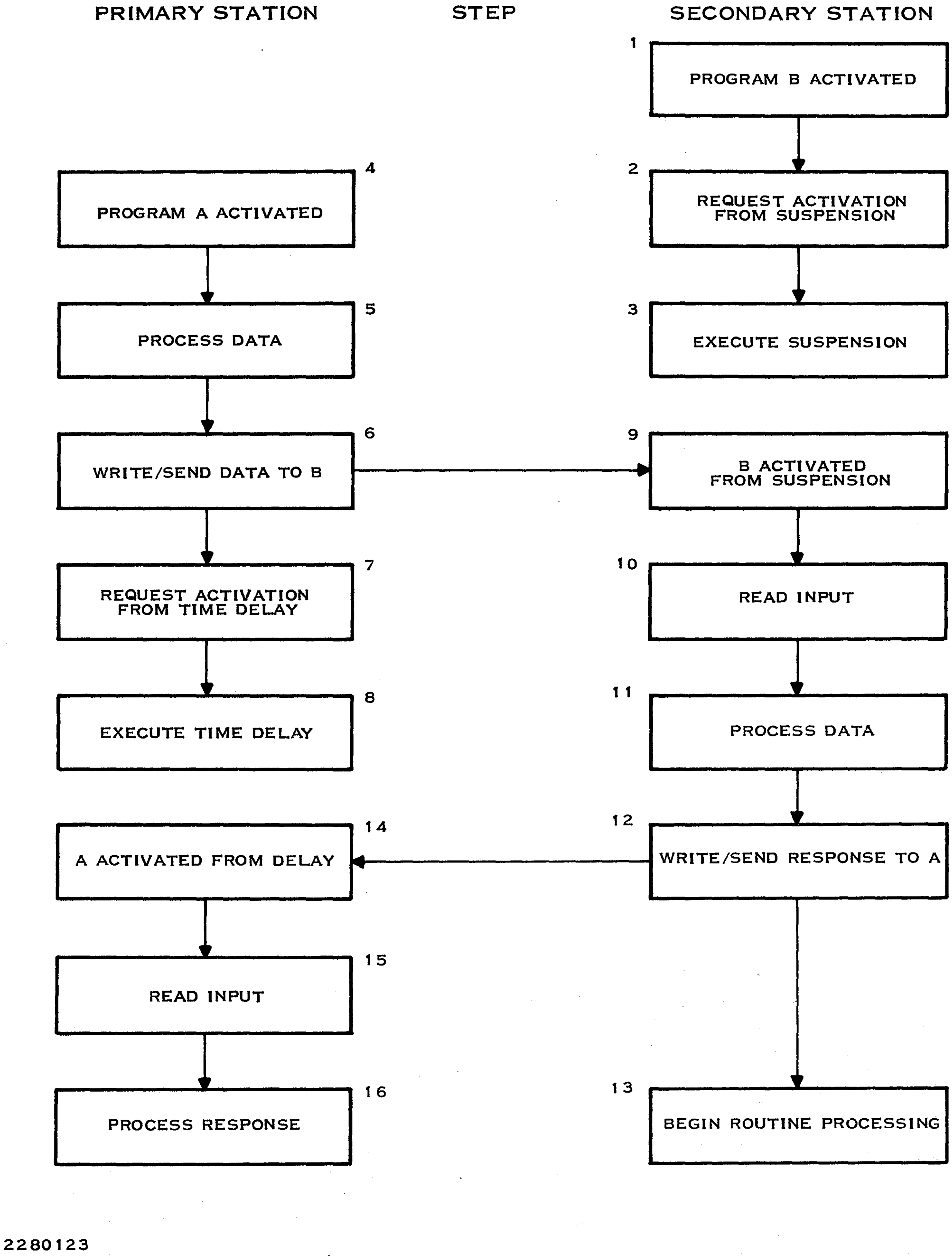
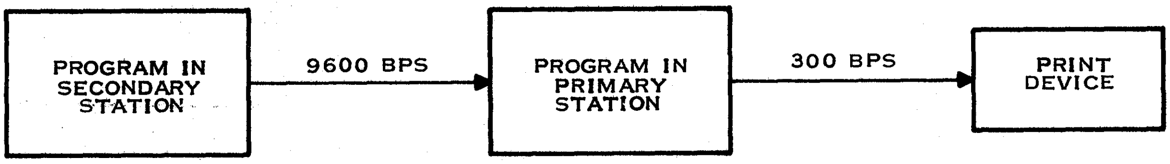


Figure 4-3 Data Flow within the Communications Package

4.4 MANAGING THE FLOW OF COMMUNICATIONS

When designing applications programs that use the network, careful consideration should be given to managing the flow of communications between programs. Figure 4-4 shows an example of a program in a secondary station that is sending data to a program in the primary station. The data is to be formatted and printed at the primary station. The communications line that connects the stations operates at 9600 bits per second (bps). A rapid succession of data sent from the secondary station arrives at the primary station at 9600 bps (about 1200 characters per second), and is formatted and printed at 300 bps (about 37 characters per second). Traffic is regulated on the communications line; therefore, the communications package soon quits accepting the data by returning an error code to the sender. If this traffic were not regulated, the primary station would quickly become congested with input data because of the difference in the line capacity of the communications line and the printer.



2280124

Figure 4-4 Managing the Flow of Communications

One possible solution to this problem is for the applications programmer to put some constraints on the program sending the data. Another possible solution is to have the receiving program write input data to a disk file as it arrives and interleave write operations to the print device with read operations to the communications package.

This example illustrates that applications programmers must be familiar with the configuration of the network, as well as the environment in which the communications package operates. The flow of traffic within the network is a function of several conditions, many of which are dependent upon external factors that influence the operation of the communications package. It is suggested that careful consideration be given to ways of avoiding congestion when designing applications programs.

4.5 ASSEMBLY LANGUAGE PROGRAMMING EXAMPLES

Figure 4-5 and Figure 4-6 list assembly language programs in DX10 and TX5 that communicate with each other. The program listed in Figure 4-5 is intended for a DX10 primary station, and the program listed in Figure 4-6 is intended for a secondary station.

```
       IDT  'PRIMPROG'
       TITL 'PRIMARY TO SECONDARY MESSAGE'
*
       DATA WP,PC,0
*
WP     BSS  32                 PROGRAM WORKSPACE
       DXOP SVC,15
*
***** SVC >4D WRITE PRB *****
*
WRITE  BYTE >4D           SVC OPCODE
WERR   BYTE 0             STATUS CODE
WOPCOD BYTE 2             I/O WRITE CALL
       BYTE 0             RESERVED BYTE
WSYSFG BYTE 0             SYSTEM FLAGS
WUSRFG BYTE 0             USER FLAGS
WBUFR  DATA OUTBUF        OUTPUT BUFFER POINTER
       DATA 0             LOGICAL RECORD LENGTH FOR READ
WCOUNT DATA OUTLEN        OUTPUT BUFFER LENGTH
       DATA 0             RESERVED
WDID   DATA >0200         DID. * NOTE HEX FORMAT *
WSID   DATA 0             SID. RETURNED AFTER CALL
*
***** SVC >4D READ PRB *****
*
READ   BYTE >4D           SVC OPCODE
RERR   BYTE 0             STATUS CODE
READCD BYTE 3             I/O READ CALL
       BYTE 0             RESERVED BYTE
RSYSFG BYTE 0             SYSTEM FLAGS
RUSRFG BYTE 0             USER FLAGS
RBUFR  DATA INBUF         READ BUFFER POINTER
LRL    DATA INLEN         MAX INPUT BUFFER LENGTH
RCOUNT DATA 0             # BYTES IN INPUT BUFFER SET BY COMM.
       DATA 0             RESERVED
RDID   DATA 0             RETURNED DID OF SENDING TASK
RSID   DATA 0             SID. MAY BE SET TO ZERO.
```

Figure 4-5 Primary Station Assembly Language Program (Sheet 1 of 3)

```
*
***** SVC >4D ACTIVATION SERVICES REQUEST *****
*
ACTSVC BYTE >4D          SVC OPCODE
ASERR  BYTE 0            STATUS CODE
ASCODE BYTE 0,0          ACT. SERVICES CODE; RESERVED BYTE
*
*** ACTIVATION SERVICES OPTION CODES TABLE BELOW ***
*
UNSUSP BYTE 4            REQUEST UNSUSPENSION
TIMEDY BYTE 5            REQUEST ACT. FROM TIME DELAY
BIDME  BYTE 6            REQUEST TO BE BID
REMOVE BYTE 7            REMOVE MY ID FROM ACT. SVCS LIST
*
*** DX10 SVC CALL BLOCKS FOR STATES ASSUMED ***
*
DELAY  DATA >0200,400    REQUEST 10 SECOND TIME DELAY
SUSPND DATA >0600        REQUEST TO BE SUSPENDED
TERMIN DATA >0400        REQUEST TO BE TERMINATED
*
LOGOUT BYTE OUTLEN
OUTBUF TEXT 'THIS MESSAGE IS SENT TO THE SECONDARY'
OUTLEN EQU  $-LOGOUT-1
       EVEN
*
LOGIN  BYTE 0            SET FOR SYSLOG WRITE
INBUF  BSS  256          INPUT MESSAGE BUFFER
INLEN  EQU  256          TX MAX BUFFER LENGTH
*
ERRMSG BYTE ERRLEN
       TEXT 'ERROR ON READ FROM SECONDARY-NO RESPONSE'
ERRLEN EQU  $-ERRMSG-1
*
SYSLOG DATA >2100,0      SYSTEM LOG PRB
MSGPTR DATA 0,0          POINTER TO OUTPUT OR INPUT MESG
*
```

Figure 4-5 Primary Station Assembly Language Program (Sheet 2 of 3)

```
* THIS SAMPLE PROGRAM USES THE >4D SVCs WITHOUT
* TESTING FOR ERROR RETURNS.  HOWEVER, THE USER IS REMINDED
* THAT ERROR TESTING SHOULD ALWAYS BE DONE TO ENSURE THAT
* THE CALL WAS SUCCESSFULLY MADE.  THE PROGRAM WRITES THE
* OUTPUT MESSAGE AND THE RESPONSE (INPUT) MESSAGE TO THE
* SYSTEM LOG.
*
*
PC      SVC   @WRITE               SEND MESSAGE TO SECONDARY TASK,
*                                    ID IS >0200  (BCD)
        LI    R1,LOGOUT            POINTER TO OUTPUT MESG
        MOV   R1,@MSGPTR           *
        SVC   @SYSLOG              WRITE OUTPUT MESG TO SYSLOG
        MOVB  @TIMEDY,@ASCODE      REQ. ACT SERVICES FROM T/DELAY
        SVC   @ACTSVC              *
        SVC   @DELAY               SYSTEM TIME DELAY, 10 SECS;
*                                  AM WAITING REPLY FROM ID >0200
*
** AM ACTIVE NOW.  MUST FIRST DO READ CALL
*
        SVC   @READ                READ REPLY FROM ID >0200
        MOVB  @READ+1,R0           CHECK FOR NO REPLY
        JEQ   NOERRS               REPLY RECEIVED
        LI    R1,ERRMSG            NO REPLY, WRITE MESG AND QUIT
        JMP   WRYTER               *
*       .
*       .                          PROCESS THE DATA HERE AS REQUIRED,
*       .                          THEN CLOSE AND TERMINATE.
NOERRS  MOVB  @RCOUNT+1,@LOGIN     PUT INPUT MESG LENGTH SYSLOG
        LI    R1,LOGIN             POINTER TO INPUT MESG
WRYTER  MOV   R1,@MSGPTR           FOR SYSLOG WRITE
        SVC   @SYSLOG              WRITE MESG TO SYSLOG
        MOVB  @BIDME,@ASCODE       REQUEST ACT SERVICES TO BID ME
*                                  IF ANY SECONDARY TASK SENDS
*                                  ME A MESSAGE.
        SVC   @ACTSVC              *
        SVC   @TERMIN              TERMINATE THE TASK
*
*               THE TASK CAN NOW BE ACTIVATED FROM A TERMINAL
*               OR BY ACTIVATION SERVICES IF AN INPUT MESSAGE
*               FOR THE TASK ARRIVES FROM A SECONDARY TASK.
        END
```

Figure 4-5 Primary Station Assembly Language Program (Sheet 3 of 3)

```
        IDT  ´SECPROG´
        TITL ´SECONDARY TO PRIMARY RESPONSE´
*
        DATA WP,PC,0
*
WP      BSS  32                   PROGRAM WORKSPACE
        DXOP SVC,15
*
***** SVC >4D READ PRB *****
*
READ    BYTE >4D             SVC OPCODE
RERR    BYTE 0               STATUS CODE
READCD  BYTE 3               I/O READ CALL
        BYTE 0               RESERVED BYTE
RSYSFG  BYTE 0               SYSTEM FLAGS
RUSRFG  BYTE 0               USER FLAGS
RBUFR   DATA INBUF           READ BUFFER POINTER
LRL     DATA INLEN           MAX INPUT BUFFER LENGTH
RCOUNT  DATA 0               # BYTES IN INPUT BUFFER SET BY COMM.
        DATA 0               RESERVED
RDID    DATA 0               SENDING TASK DID RETURNED HERE.
RSID    DATA 0               SID. MAY BE SET TO ZERO.
*
***** SVC >4D WRITE PRB *****
*
WRITE   BYTE >4D             SVC OPCODE
WERR    BYTE 0               STATUS CODE
WOPCOD  BYTE 2               I/O WRITE CALL
        BYTE 0               RESERVED BYTE
WSYSFG  BYTE 0               SYSTEM FLAGS
WUSRFG  BYTE 0               USER FLAGS
WBUFR   DATA OUTBUF          OUTPUT BUFFER
        DATA 0               LOGICAL RECORD LENGTH FOR READ
WCOUNT  DATA OUTLEN          OUTPUT BUFFER LENGTH
        DATA 0               RESERVED
WDID    DATA 0               DID. FROM READ CALL.
WSID    DATA 0               SID. MAY BE SET TO ZERO.
*
***** SVC >4D ACTIVATION SERVICES REQUEST *****
*
ACTSVC  BYTE >4D             SVC OPCODE
ASERR   BYTE 0               STATUS CODE
ASCODE  BYTE 0,0             ACT. SERVICES CODE; RESERVED BYTE
```

Figure 4-6 Secondary Station Assembly Language Program (Sheet 1 of 2)

```
*
*** ACTIVATION SERVICES OPTION CODES TABLE BELOW ***
*
UNSUSP BYTE 4              REQUEST UNSUSPENSION
TIMEDY BYTE 5              REQUEST ACT. FROM TIME DELAY
BIDME  BYTE 6              REQUEST TO BE BID
REMOVE BYTE 7              REMOVE MY ID FROM ACT. SVCS LIST
*
*** TX5 SVC CALL BLOCKS FOR SUSPEND STATE ASSUMED ***
*
SUSPND DATA >0600          REQUEST TO BE SUSPENDED
*
OUTBUF TEXT 'THIS RESPONSE MESSAGE IS SENT TO THE PRIMARY'
OUTLEN EQU  $-OUTBUF
*
INBUF  BSS  256            INPUT MESSAGE BUFFER
INLEN  EQU  256            TX MAX BUFFER LENGTH ALLOWED
*
* THIS SAMPLE PROGRAM USES THE >4D SVCs WITHOUT
* TESTING FOR ERROR RETURNS.  THE USER IS REMINDED
* THAT ERROR TESTING SHOULD ALWAYS BE DONE TO ENSURE THAT
* THE CALL WAS SUCCESSFULLY MADE.
*
PC     MOVB @UNSUSP,@ASCODE   REQUEST UNSUSPENSION FROM ACT SVCS
       SVC  @ACTSVC           *
       SVC  @SUSPND           SYSTEM SUSPEND REQUEST.
*
**   AM UNSUSPENDED HERE. DO READ CALL WHEN ACTIVE.
*
       SVC  @READ             READ MESG.
       MOV  @RDID,@WDID       SET ADDRESS FOR RESPONSE
       SVC  @WRITE            SEND RESPONSE MESSAGE TO SENDER
*      .
*      .                      PROCESS INPUT DATA HERE,
*      .                      *
       JMP  PC                START OVER.  REQ UNSUSPEND
*                             IF ANY SECONDARY TASK SENDS
*                             A MESSAGE.
       END
```

Figure 4-6 Secondary Station Assembly Language Program (Sheet 2 of 2)

4.6 FORTRAN AND PASCAL PROGRAMMING EXAMPLES

Applications programs written in FORTRAN or Pascal can interface with the communications package by calling assembly language subroutines. These subroutines build the SVC blocks, using the parameters passed to the subroutine, and then execute the SVCs.

The following paragraphs provide an example of the assembly language subroutines required, describe calls to the subroutines, and provide examples of programs that use the subroutine calls. The examples given are for FORTRAN programs, but are similar to those for Pascal programs. For Pascal programs, the subroutines WRIT4D, READ4D, and WKUP4D must be declared as EXTERNAL FORTRAN. All Pascal parameters must be defined as variable parameters (VAR).

4.6.1 Assembly Language Subroutine for FORTRAN Interface

Figure 4-7 shows an example of the assembly language code that is required for the FORTRAN interface with the communications package. This code builds the SVC blocks that allow FORTRAN programs to execute read (READ4D), write/send (WRIT4D), and request activation services (WKUP4D). The object code for this routine is on file .INDSCOMM.OBJ.SUB4D. This file contains a module that has been partially linked with a module (F$RGMY) from the FORTRAN run-time library. Thus, Pascal users do not need the FORTRAN libraries when linking.

```
       IDT  'SUB4D'
       TITL 'FORTRAN XOP15-4D SUBROUTINES'
       REF  F$RGMY
       DEF  WRIT4D,READ4D,WKUP4D,WAITSV
       DXOP SVC,15
*********************************************************
*      LAST CHANGE DATE 07/1/80
*
*      ABSTRACT - THIS MODULE IS USED TO SUPPORT
*      XOP15-4D CALLS FROM A FORTRAN PROGRAM.  THE
*      CALLS SUPPORTED ARE READ, WRITE, AND WAKE
*      UP.
*
*      INPUTS - DATA PASSED ON A CALL FROM FORTRAN
*
*      OUTPUTS - PARAMETERS PASSED BACK FROM XOP15-4D CALLS
*
*********************************************************
*
       EVEN
BLOCK  BSS  18
       EVEN
PASSED BSS  14
*
IO4D   DATA >4D00
WRIT   DATA >200
READ   DATA >300
WAIT   DATA >600
       PAGE
*
*      THIS MODULE EXECUTES A SUSPEND TASK
*
WAITSV DATA WSP6,START6,0
WSP6   BSS  32
       DATA NAME6              NAME OF SUBROUTINE
       DATA START6             STARTING ADDRESS
       DATA 1                  1=NON REENTRANT CODE
START6 BL   @F$RGMY
       DATA 0                  NUMBER OF PARAMETERS
NAME6  TEXT 'WAITSV'
       SVC  @WAIT
       RTWP
```

Figure 4-7 Assembly Language Interface (Sheet 1 of 4)

```
*
*       THIS ROUTINE WILL TAKE INPUTS FROM A FORTRAN CALL
*       AND EXECUTE AN XOP15-4D
*
*    CALL WRIT4D(BUFFER,COUNT,DID,SID,ERROR)
*
WRIT4D DATA WSP1,START1,0
WSP1   BSS  32
       DATA NAME1                NAME OF SUBROUTINE
       DATA START1               STARTING ADDRESS
       DATA 1                    1=NON REENTRANT CODE
START1 BL   @F$RGMY
       DATA 5                    NUMBER OF PARAMETERS
       DATA PASSED               ADDRESS WHERE PARAMETERS PUT
NAME1  TEXT 'WRIT4D'             6 CHARACTER SUB NAME
       LI   R1,BLOCK             SET CALL BLOCK TO WRITE
       LI   R3,PASSED            GET ADDRESS OF ADDRESSES PASSED
       MOV  @IO4D,*R1            SET THE >4D OP CODE
       MOV  @WRIT,@2(R1)         SET OP CODE AND CLEAR RUN ID
       CLR  @4(R1)               CLEAR SYSTEM FLAGS AND USER FLAGS
       MOV  *R3,@6(R1)           GET BUFFER ADDRESS PASSED
       CLR  @8(R1)               CLEAR THE LOGICAL RECORD LENGTH
       MOV  @2(R3),R2            GET ADDRESS OF CHARACTER COUNT
       MOV  *R2,@10(R1)          SET THE OUTPUT CHARACTER COUNT
       CLR  @12(R1)              CLEAR RESEARVED WORD
       MOV  @4(R3),R2            GET DID ADDRESS
       MOV  *R2,@14(R1)          SET DID IN CALL BLOCK
       MOV  @6(R3),R2            GET SID ADDRESS
       MOV  *R2,@16(R1)          SET SID IN CALL BLOCK
       SVC  *R1                  EXECUTE XOP15->4D CALL
       MOV  @8(R3),R2            GET ADDRESS OF ERROR
       MOVB @1(R1),R4            GET ERROR RETURN
       SRA  R4,8
       MOV  R4,*R2               RETURN ERROR CODE
       MOV  @4(R3),R2            RETURN DID
       MOV  @14(R1),*R2
       MOV  @6(R3),R2            RETURN SID
       MOV  @16(R1),*R2
       RTWP
       PAGE
```

Figure 4-7 Assembly Language Interface (Sheet 2 of 4)

```
*
*       THIS ROUTINE WILL GET DATA FORM A FORTRAN CALL
*       AND EXECUTE AN XOP15-4D.
*
*       READ4D(BUFFER,LENBUF,COUNT,DID,SID,ERROR)
*
READ4D DATA WSP3,START3,0
WSP3   BSS  32
       DATA NAME3               NAME OF SUBROUTINE
       DATA START3              STARTING ADDRESS
       DATA 1                   1=NON REENTRANT CODE
START3 BL   @F$RGMY
       DATA 6                   NUMBER OF PARAMETERS
       DATA PASSED
NAME3  TEXT 'READ4D'            6 CHARACTER SUB NAME
       LI   R1,BLOCK            GET CALL BLOCK ADDRESS
       LI   R3,PASSED           GET PARAMETER ADDRESS
       MOV  @IO4D,*R1           SET FOR >4D CALL
       MOV  @READ,@2(R1)        SET READ OP CODE AND CLEAR RUN ID
       CLR  @4(R1)              CLEAR SYSTEM AND USER FLAGS
       MOV  *R3,@6(R1)          SET BUFFER POINTER
       MOV  @2(R3),R2           GET LOGICAL RECORD LENGTH ADDRESS
       MOV  *R2,@8(R1)          SET LOGICAL RECORD LENGTH
       CLR  @10(R1)             CLEAR CHARACTER COUNT
       CLR  @12(R1)             CLEAR RESEARVED WORD
       MOV  @6(R3),R2           GET ADDRESS OF DID
       MOV  *R2,@14(R1)         SET DID IS CALL BLOCK
       MOV  @8(R3),R2           GET ADDRESS OF SID
       MOV  *R2,@16(R1)         SET SID IN CALL BLOCK
       SVC  *R1                 CALL XOP15->4D
       MOV  @6(R3),R2           RETURN DID
       MOV  @14(R1),*R2
       MOV  @8(R3),R2           RETURN SID
       MOV  @16(R1),*R2
       MOV  @4(R3),R2           GET THE COUNT ADDRESS
       MOV  @10(R1),*R2         RETURN COUNT FROM CALL BLOCK
       MOV  @10(R3),R2          GET THE ERROR ADDRESS
       MOVB @1(R1),R4
       SRA  R4,8
       MOV  R4,*R2              RETURN THE ERROR
       RTWP
       PAGE
```

Figure 4-7 Assembly Language Interface (Sheet 3 of 4)

```
*
*       THIS ROUTINE ENTRY WILL GET DATA PASSED FROM
*       A FORTRAN PROGRAM AND EXECUTE AN XOP15->4D
*
*       CALL WKUP4D(OPTION,ERROR)
*
WKUP4D DATA WSP4,START4,0
WSP4   BSS  32
       DATA NAME4                NAME OF SUBROUTINE
       DATA START4               STARTING ADDRESS
       DATA 1                    1=NON REENTRANT CODE
START4 BL   @F$RGMY
       DATA 2                    NUMBER OF PARAMETERS
       DATA PASSED
NAME4  TEXT 'WKUP4D'             6 CHARACTER SUB NAME
       LI   R1,BLOCK             GET CALL BLOCK ADDRESS
       LI   R3,PASSED            GET PARAMETER ADDRESS
       MOV  @IO4D,*R1            SET THE >4D OP CODE
       MOV  *R3,R2
       MOV  *R2,R4               GET THE OPTION CODE
       CI   R4,4
       JLT  ERROR                4 <= OPTION <= 7
       CI   R4,8
       JLT  OK
ERROR  LI   R4,1                 RETURN ERROR OF 1
       JMP  SET
*
OK     SLA  R4,8                 LEFT JUSTIFY IT
       MOV  R4,@2(R1)            SET THE OPTION CODE
       CLR  @4(R1)               CLEAR SYSTEM & USER FLAGS
       CLR  @6(R1)               CLEAR BUFFER POINTER
       CLR  @8(R1)               CLEAR LOGICAL RECORD LENGTH
       CLR  @10(R1)              CLEAR CHARACTER COUNT
       CLR  @12(R1)              CLEAR RESEARVED WORD
       CLR  @14(R1)              CLEAR DID
       CLR  @16(R1)              CLEAR SID
       SVC  *R1                  DO XOP15->4D
       MOVB @1(R1),R4            GET ERROR RETURN
       SRA  R4,8
SET    MOV  @2(R3),R2            GET ERROR ADDRESS
       MOV  R4,*R2               RETURN THE ERROR
       RTWP
       PAGE
       END
```

Figure 4-7 Assembly Language Interface (Sheet 4 of 4)

4.6.2 Subroutine Calls

The following paragraphs describe the calls, which are similiar for FORTRAN and Pascal, to the previous assembly language subroutine. Unless stated otherwise, all parameters passed in the call statements must be 16-bit integer variables (INTEGER*2 in FORTRAN and INTEGER in Pascal). In the syntax definition for each call statement, the parameters are enclosed in angle brackets (<>), indicating that they are to be replaced by your assigned variables.

Note that available network IDs are always expressed as BCD numbers in the range 0100 through 9999. However, they must be supplied in applications programs in hexadecimal format. For example, a network ID of 0431 would be entered in the program as >0431. The characters A through F are invalid for all network IDs. The IDs 0000 to 0099 are reserved for the communications package software.

The subroutine calls for read (READ4D), write (WRIT4D), and activation services (WKUP4D) are described in the following paragraphs.

4.6.2.1 READ4D Subroutine Call.

Subroutine READ4D builds an SVC block and executes an SVC with I/O opcode 3. A call to this subroutine causes one block of data addressed to the calling program to be transferred from a communications input buffer to the program's data buffer. A program can either execute a call to READ4D and test the status code to determine if any input data was found, or request activation services with subroutine WKUP4D and then retrieve the data with subroutine READ4D after it has been reactivated. One call to subroutine READ4D retrieves one block of data; retrieving multiple blocks of data requires multiple calls. The syntax for the call statement is as follows:

FORTRAN:

```
  CALL READ4D(<buffer>,<buf lng>,<char cnt>,<did>,<sid>,<st code>)
```

Pascal:

```
  READ4D(<buffer>,<buf lng>,<char cnt>,<did>,<sid>,<st code>)
```

where:

<buffer>	is the name of the calling programs´s data buffer into which the subroutine returns the data that has been found. The data buffer is an array in FORTRAN tasks and either an array or a character string in Pascal tasks.
<buf lng>	contains the length in characters (one byte per character) of the calling program´s data buffer. If this value is less than the data length (character count), the subroutine returns an error code 9 in the status code variable and it returns the length of the data that has been sent in the character count variable. The subroutine does not transfer any portion of the data when the length of the data that has been sent is too large for the program´s data buffer.
<char cnt>	is the variable to which the subroutine returns the character count. This is the number of characters (one byte per character) in the data block that the subroutine transferred from a communications input buffer to the calling program´s data buffer.
<did>	contains, after successful completion of the read call, the network ID of the program or remote device that sent the data. This variable always contains the network ID of the remote program or device. If the calling program executes in a secondary station, the remote program would be in the primary station.
<sid>	contains, after successful completion of the read call, the network ID of the calling program. This is true whether the calling program (the program that issued the read call) executes in the primary station or a secondary station.
<st code>	is the variable to which the subroutine returns the status code. The calling program should check the status code that is returned and take appropriate action. The codes and their meanings are as follows:

Code	Meaning
00	Read operation has been successfully completed by the communications package.
01	Not used.
02	Not used.
03	Invalid SID specified.
04	No input data has been received for the program to read.
05	Not used.
06	The communications package is not active.
07	Not used.
08	Invalid DID specified.
09	Buffer length specified is too small to read input data.
0A	Not used.
0B	Not used.

4.6.2.2 WRIT4D Subroutine Call.

Subroutine WRIT4D builds an SVC block and executes an SVC with I/O opcode 2. A call to this subroutine causes one block of data in the calling program to be transferred from the program's data buffer to a communications output buffer for subsequent transmission to a DID in another station. Control is returned to the calling program immediately upon completion of the data being transferred to the communications output buffer.

No provisions are made for automatic response from the DID. Applications programs can establish their own methods for validating data sent and/or received (for example, DID can send a message back to the SID to validate data received).

The syntax for the call statement is as follows:

FORTRAN:

```
 CALL WRIT4D(<buffer>,<char cnt>,<did>,<sid>,<st code>)
```

Pascal:

WRIT4D(<buffer>,<char cnt>,<did>,<sid>,<st code>)

where:

<buffer>	is the name of the calling program's data buffer that contains the data to be sent. The data buffer is an array in FORTRAN tasks and either an array or a character string in Pascal tasks.
<char cnt>	contains the character count. This is the number of characters (one byte per character) in the data that the subroutine is to transfer from the calling program's data buffer to a communications output buffer.
<did>	contains the network ID of the destination program or remote device. If a calling program in a TX5 secondary provides a DID that is incorrect, the data is transmitted and then is deleted at the primary station and an error message is written to the system log at the primary station. Validation of DIDs is not performed in TX5 secondary stations, as all data is sent to the primary station. If a calling program in a primary station provides a DID that is incorrect, an error code of 8 is returned in the status code variable.
<sid>	contains either a zero or the network ID of the calling program. If this parameter contains a zero, the subroutine supplies the calling program's network ID to the variable.
<st code>	is the variable to which the subroutine returns the status code. The applications program should check the status code that is returned and take appropriate action. The codes and their meanings are as follows:

Code	Meaning
00	Data specified has been transferred to the communications package output buffer.
01	Not used.
02	Not used.
03	Invalid SID specified.
04	Not used.
05	No buffer is currently available in the operating system to accept the output data. If this code is returned, the program should execute a time delay of at least 100 milliseconds before executing a second call.
06	The communications package is not active.
07	Not used.
08	Invalid DID specified.
09	Character count exceeds maximum output buffer length.
0A	Destination specified in DID is inoperative, in that the station is down.
0B	No buffer is available in the station to accept the output data.

4.6.2.3 WKUP4D Subroutine Call.

For requests of activation services, subroutine WKUP4D builds an SVC block and executes an SVC with the I/O opcode specified by the calling program. A call to subroutine WKUP4D places the network ID of the calling program into the activation services list or removes the calling program's network ID from that list. A program that has requested activation services is activated when an input data addressed to it is found in the communications input buffers.

The syntax for the call statement is as follows:

FORTRAN:

```
CALL WKUP4D (<option>,<st code>)
```

Pascal:

```
WKUP4D (<option>,<st code>)
```

where:

<option> contains the option code. Valid option codes used to make requests of activation services are as follows:

Code	Function
04	Request is made for the applications program to be activated from an indefinite suspension when input data from any station is addressed to it. Applications programs that request this service may be installed on any program file in the DX10 system.
05	Request is made for the applications program to be activated from a time delay when input data from any station is addressed to it. Applications programs that request this service may be installed on any program file in the DX10 system.

06 Request is made for the applications program to be bid when input data from any station is addressed to it. Applications programs that request activation from termination, in that they request to be bid, must be installed in a DX10 program file named .INDSCOMM.USERPROG and global LUNO >40 must be assigned to the file. The bid task service in TX5 is supported for resident tasks only.

07 Request is made to remove the applications program from the activation services list.

<st code> is the variable to which the subroutine returns the status code. The calling program should check the status code returned and take appropriate action. The codes and their meanings are as follows:

Code	Meaning
00	Activation services has accepted the call.
01	Illegal option code specified.
02	Not used.
03	Invalid task ID. The program requesting activation services must have a network ID assigned to it by the network generator program NETGEN.
04	Not used.
05	System buffers are full; unable to accept the request.
06	Communications package inactive. In the primary station, this request is not accepted. In TX5 secondary stations, the request is accepted in spite of this condition.

After the activation services request is made, the applications program should make the appropriate inactive state request by executing one of the following SVCs:

* Time Delay SVC, if request of activation services was to be activated from a time delay. A minimum delay request of 5 to 20 seconds (depending on the number of attached secondary stations and traffic flow in the network) should be made.

* Suspend SVC, if request of activation services was to be unsuspended.

* Terminate SVC, if request of activation services was to be bid.

4.6.3 FORTRAN Programming Examples

Figure 4-8 and Figure 4-9 list FORTRAN programs that communicate with each other. The program listed in Figure 4-8 is intended for a primary station and the program listed in Figure 4-9 is intended for a secondary station.

```
C THIS TASK EXECUTES IN A PRIMARY SYSTEM AND SENDS A MESSAGE TO
C A SECONDARY WITH A NETWORK ID OF 0503. AFTER SENDING THE
C MESSAGE THE TASK REQUESTS ACTIVATIONS SERVICES TO AWAKEN IT IF
C ANY MESSAGES ARRIVE DESTINED FOR IT. AFTER REACTIVATION, THE
C TASK READS ANY RESPONSE FROM THE SECONDARY (DID)
C
  INTEGER*2 DID,SID,BUFFER(2),COUNT,STATUS,OPTION,BUFLEN
  DATA      DID/>503/,SID/0/,COUNT/4/,BUFFER(1),BUFFER(2)/'HI'/
  DATA      OPTION/2/,BUFLEN/4/
C
C WRITE THE MESSAGE TO THE SECONDARY
C
    CALL WRIT4D(BUFFER,COUNT,DID,SID,STATUS)
    WRITE(6,101) STATUS,BUFFER
C
C CALL ACTIVATION SERVICES WITH THE OPTION SET TO INDICATE THAT
C THIS TASK WILL BE IN A TIME DELAY WHEN/IF A MESSAGE COMES IN
C
    CALL WKUP4D(OPTION,STATUS)
    WRITE(6,102)
C
C DELAY FOR FIVE SECONDS
C
    CALL WAIT(5,2,STATUS)
C
C REACTIVATED HERE EITHER BY ACTIVATION SERVICES OR BY THE
C TIME DELAY EXPIRING
C READ ANY MESSAGES DESTINED FOR THIS TASK (I.E., THIS SID)
C
    CALL READ4D(BUFFER,BUFLEN,COUNT,DID,SID,STATUS)
    WRITE(6,103) STATUS
    WRITE(6,104) BUFFER,SID,DID
C
C TERMINATE
C
    STOP
101 FORMAT('WRITE4D CALL STATUS=',Z4,'MESSAGE SENT="',A2,A2,'"')
102 FORMAT('WAITING 5 SECONDS FOR RESPONSE')
103 FORMAT('READ4D CALL STATUS=',Z4)
104 FORMAT('MESSAGE RETURNED="',A2,A2,'"','SID=',Z4,'DID=',Z4)
    END
```

Figure 4-8 FORTRAN Program in Primary Station

```
C THIS TASK WAITS ON MESSAGES FROM A PRIMARY TASK. WHEN A
C MESSAGE IS RECEIVED THE TASK RESPONDS WITH A MESSAGE OF
C ITS OWN OR SENDS THE RECEIVED MESSAGE BACK DEPENDING ON
C THE INPUT CHARACTER COUNT
C
  INTEGER*2 DID,SID,STATUS,COUNT,BUFFER(2),OPTION,BUFLEN,BACK(2)
  DATA      DID/0/,SID/0/,OPTION/2/,BUFLEN/4/
  DATA      BACK(1),BACK(2)/'BYE '/
C
C CALL ACTIVATION SERVICES TO REQUEST ACTIVATION FROM A TIME
C DELAY IF A MESSAGE COMES IN FOR MY SID
C
4    CALL WKUP4D(OPTION,STATUS)
C
C DO A TIME DELAY
C
     CALL WAIT(5,2,STATUS)
C
C SEE IF ANY MESSAGES HAVE COME IN
C
     CALL READ4D(BUFFER,BUFLEN,COUNT,DID,SID,STATUS)
     IF (STATUS .NE. 0) GO TO 4
     IF (COUNT .NE. 4) GO TO 5
C
C WRITE REPLY MESSAGE IF INPUT CHARACTER COUNT IS AS EXPECTED
C
     CALL WRIT4D(BACK,BUFLEN,DID,SID,STATUS)
     GO TO 4
C
C RETURN INPUT BUFFER IF INPUT CHARACTER NOT AS EXPECTED
C
5    CALL WRIT4D(BUFFER,COUNT,DID,SID,STATUS)
     GO TO 4
     END
```

Figure 4-9 FORTRAN Program in Secondary Station

Section 5

Applications Utilities

5.1 GENERAL

Applications utilities supported in the DX10 HDLC Communications Package provide assistance in all phases of program generation and execution. The utilities are as follows:

* A download preprocessor utility for converting a standard DX10 object file into a relocated binary object file preparatory to downloading (transmitting) to a specified secondary station.

* A download task utility for transferring the binary object file to the specified secondary station via the network.

* Remote System Command Interpreter (SCI) services for interacting with a TX5 secondary station from a terminal at the primary station. Remote SCI is analogous to the functions of the SCI under DX10 and performs similar interactive operations.

* Activate and deactivate utilities for placing secondary stations online or offline.

* Statistics utility for providing data to efficiently manage the network.

The application and operation of these utilities are described in detail in this section. All utilities are interactive and require a Model 911 Video Display Terminal (VDT) for operation.

Prerequisites for the successful operation of these utilities include the following:

* A knowledge of the network configuration and the characteristics of each station in the network

* A knowledge of the DX10 operating system

5.2 DOWNLOAD PREPROCESSOR UTILITY

The download preprocessor utility converts a standard DX10 object file into a downloadable memory image for a specified secondary station. This binary formatted image is stored in a download directory. A maximum of eight file entries (called options) can be maintained for each secondary. These files may contain operating systems or stand-alone programs (such as diagnostics) for downloading to that station.

The download preprocessor utility performs the following:

* Alias functions, including:
  - Assignment of an alias, that is, assignment of an alternate name, to a secondary station.
  - Deletion of an alias previously assigned
  - Modification of an alias previously assigned
* Preprocessed file functions, including:
  - File creation
  - File deletion
  - File contents dumping
  - File patching or modification
* Download file listing functions, including:
  - Listing by station identity number
  - Listing by station aliases
  - Listing of optional download files by station
  - The relocation value of the binary object file
  - The size of each record in each file assigned

The utility supports a variable length record in order to provide downloadable modules suitable for a wide range of secondary station requirements. A special DX10 HDLC Communications ROM Loader must be installed in Model 990/5 Computers to accept preprocessed object records. This ROM Loader interfaces to a local-line module (LLM) board to load preprocessed data into 990/5 computer memory. The LLM must be at CRU address >20. Each preprocessed record is formatted as follows:

```
           +------------------------------------------+
Bytes 0,1  |           Load Address of Data           |
           +------------------------------------------+
      2,3  | Length of Load Record (number of words)  |
           +------------------------------------------+
      4,5  |   Load Record (data to be transmitted)   |
           +------------------------------------------+
```

5.2.1 Preprocessor Background Information

The download preprocessor utility creates and maintains a download directory in the primary station. Although the utility is interactive and designed for minimum reliance on this manual, the following background information should enhance your initial comprehension of the utility:

* Download directories and files. Files are stored in the .DOWNLOAD directory as .DOWNLOAD.S00nnnn.OPm

where:

00nnnn	is a six decimal digit station number. The first two digits are always zero; the last four digits are the station address.
m	is an option number in the range of 0 through 7. The options are files available for downloading to the specified station. A maximum of eight optional files are available per station.

A station may be defined in the directory whether or not it physically exists or is actually connected to the network.

Other files contained in the directory .DOWNLOAD are:

- .DOWNLOAD.ALIAS. Contains a list of all aliases associated with the communications package.
- .DOWNLOAD.ALIASIN. Each entry in this file contains a secondary station ID, the option flags, and the number of aliases defined for that station
- .DOWNLOAD.DUMP. Used to build the dump image (see paragraph 5.2.3.2).

In the normal course of operation, there is no requirement for the user to access these files by name. However, precautions should be taken to protect these files from being unintentionally deleted.

- Download records. The download record has a minimum size of 32 bytes and a maximum of 512 bytes, including the four-byte header block. The size is dependent on the type of secondary station and is a function of such things as the size of the communications buffers at the primary and secondary stations.

5.2.2 Preprocessor Utility Execution

The following paragraphs contain information on the use of the preprocessor utility. The initial menu of commands available with the preprocessor is also described.

- Preprocessor activation. The preprocessor is activated by the SCI command Preprocessor (PP), and causes the following initial menu to be displayed on the terminal:

```
[] PP

    PREPROCESSOR - AA, AM, AD, DB, DD, DL, DM, DU
                ENTER COMMAND:
```

In response to the ENTER COMMAND prompt enter the mnemonic for the desired function and press the RETURN key. A new set of prompts is then presented on the screen. Enter the required information. When the function is complete, the preprocessor terminates and is reactivated only when the PP command is reentered.

Alternatively, PRExx (where xx is the two-character mnemonic of the desired function) can be entered in response to the SCI prompt, as in the following example:

```
[] PREAA
```

This entry selects the AA function without displaying the initial preprocessor menu.

The mnemonics displayed on the initial menu have the following meanings:

Alias commands:

AA	Add (assign) an alias for a secondary station.
AD	Delete an alias assigned to a secondary station.
AM	Modify an alias assignment.

File operation commands:

DB	Build a download file for the .DOWNLOAD directory.
DD	Delete a download file from the .DOWNLOAD directory.
DU	Dump a download file, that is, display the contents of a preprocessed file.
DL	List download files and aliases.
DM	Modify a preprocessed download file.

* Preprocessor termination. Once the utility is activated, it may be terminated by one of the following:
  - Pressing the CMD key in response to any prompt.
  - Allowing the utility to terminate when the selected function completes.

5.2.3 Preprocessor Command Prompts and Responses

The download preprocessor utility guides you step by step through the required operations. Starting with the menu of preprocessor commands, select the two-character mnemonic required and type it on the keyboard. The resulting display is similar to other DX10 command prompts. The number and type of characters to be entered are specified along with the error messages that are displayed when a response fails to meet the entry requirements.

Responses to preprocessor prompts must generally conform to the following standards:

* Station number. When a number is used instead of an alias, the four-digit station number must be entered. The number is always interpreted as a binary coded decimal (BCD). An example of a station number is 0214.

* Station alias. An alphanumeric string not exceeding 14 characters may be used. Only the letters A through Z, numerals 0 through 9, and the $ sign are allowed. The first character must always be a letter. An example of an alias is PIPELINE$STA2.

- Memory location. This can be expressed in either decimal or hexadecimal. If it is expressed in hexadecimal, as elsewhere in DX10 operation, it must be preceded by a leading 0 (zero) or greater than (>) sign. If decimal notation is used, no prefixes are required.

- Option number. This must be a number in the range 0 through 7.

5.2.3.1 Preprocessor Alias Commands.

Aliases (alternate names) can be added, deleted, or modified using the following commands:

Add (Assign) an Alias.

```
ENTER COMMAND:  AA

 ASSIGN AN ALIAS TO A STATION
       STATION NUMBER: 0200          (example
        STATION ALIAS: BOILERTEST$    responses)
```

Responses must be entered as follows:

- The station number must be a valid station number and be four decimal digits.

- The station alias cannot exceed 14 characters in length and must begin with an alphabetic character. Numerals and the $ sign may be included in the character string. The response to the STATION ALIAS prompt shown above is an example of a valid alias. An alias can be assigned to a nonexistent station.

An invalid alias string causes one of the following messages to appear:

```
Only alphanumerics, A-Z, 0-9, and $ allowed

Alias already assigned
```

After an alias is assigned to a station, the station can be referred to by that alias whenever the preprocessor or downloader is used.

The maximum number of aliases that can be assigned to one station is 20.

Modify a Secondary Station Alias.

```
ENTER COMMAND:  AM

          MODIFY A STATION ALIAS
               STATION NUMBER: (decimal number)
                STATION ALIAS:
```

This command can be used for the following purposes:

- To reassign an alias from one station to another
- To add a new alias for an existing station
- To assign an alias for a proposed secondary station

When using this command, be aware of the following:

- To reassign an alias, you must enter the station number to which it is to be reassigned. This deletes the previous assignment and creates the new one.
- If the alias does not already exist, it is added to the aliases for the specified secondary station.
- If the station number does not exist, it is added to the .DOWNLOAD directory and the alias is assigned to it.

An invalid alias string causes the following error message to be displayed on the terminal:

```
Only alphanumerics, A-Z, 0-9, and $ allowed
```

Delete a Secondary Station Alias.

```
ENTER COMMAND:  AD

          DELETE A STATION ALIAS
               STATION ALIAS:
               ARE YOU SURE?:
```

A response must be entered to the ARE YOU SURE? prompt. If the response is Y (yes), the alias entered is deleted. If the response is N (no), it is not deleted and the preprocessor terminates with the following error message:

```
Alias not deleted
```

If the alias is invalid, the following error message appears:

```
This alias name is not defined
```

5.2.3.2 Preprocessor File Operations.

This paragraph describes the individual file commands (Build, Delete, Dump, Modify, and List). Examples are given where appropriate. Default conditions, error conditions, and special requirements are included.

Build a Preprocessed Download File.

```
ENTER COMMAND: DB

BUILD a PREPROCESSED DOWNLOAD FILE
      FILE TO PREPROCESS: (pathname or synonym)
        RELOCATION VALUE: >A0
    DOWNLOAD RECORD SIZE: 256
 STATION NUMBER OR ALIAS: (decimal number or alias)
  DOWNLOAD OPTION NUMBER: 0
               REPLACE ?: NO
```

This function reformats a specified object file into a memory image file suitable for downloading to the secondary station and enters it in the .DOWNLOAD directory. Note the following:

* The file to be preprocessed must be a DX10 assembled or noncompressed linked object file. The entire pathname or a synonym assigned to the file must be entered.

* The relocation value is the absolute memory location in the secondary station where program loading is to start; for TX5 systems, this is location >00A0.

* The record size must be from 32 through 512 bytes and cannot be greater than the maximum message size for the station. For TX5 secondary stations, the maximum is 256 bytes.

* The station number must be a valid number or an alias that has already been assigned.

* The option number must be from 0 through 7 (0 is the default option file). If the file already exists and the default response of NO to the REPLACE? prompt is taken, the existing file and option for that station are not erased and the following error message appears on the screen:

  ```
  Option already exists.
  ```

  This message indicates that the option file was previously created. To replace an existing file, respond Y to the prompt.

Delete a Preprocessed Download File.

```
ENTER COMMAND: DD

 DELETE A DOWNLOAD FILE
   STATION NUMBER OR ALIAS: (decimal number or alias)
    DOWNLOAD OPTION NUMBER: (0-7)
              ARE YOU SURE?: (yes or no)
```

Note the following:

- The station number or alias is entered to identify the directory name and the option number (0 through 7) is entered to identify the file to be deleted. The following error messages may appear:

  ```
  This alias name is not defined

  No such station exists

  This download file does not exist, check inputs
  ```

- Deleting a file does not delete any aliases that may be assigned to the station, nor does a single execution of this command delete all option number files.

- If the response to ARE YOU SURE? is Y or YES, the specified file is deleted. If the response is N or NO, the file is not deleted and the preprocessor terminates with the following error message:

  ```
  Download file not deleted
  ```

  Pressing the RETURN key does not default through this prompt.

List Download Files and Aliases.

```
    ENTER COMMAND: DL

     LIST DOWNLOAD FILES AND ALIASES
         LISTING ACCESS NAME:
```

The listing access name is the name of an output device or file to which the listing is written. The default value of this prompt is the terminal. If a file is specified that does not exist, it is created in the execution of the function.

An example of the format for this listing is given below and includes examples of station IDs and aliases.

```
                    Download Information
 Station        Station                      Relocation  Record
   ID           Aliases          Options        Value      Size
===============================================================
  0101      CHILLERTEST01    |      0           >00A0       256
            CHILLERTEST02    |      1           >00A0       256
            OS0100           |
---------------------------------------------------------------
  0200      BOILERTEST$$     |      0           >2400       128
            OS0200           |
            BOILERTEST14     |      1           >2400       128
            DIAGNOSTIC1      |      2           >2400       128
---------------------------------------------------------------
  0300      GASPIPE20        |      0           >00A0       256
---------------------------------------------------------------
```

Modify (Patch) a Preprocessed File.

```
ENTER COMMAND: DM

MODIFY A PREPROCESSED FILE          (example
  STATION NUMBER OR ALIAS: 0650     responses)
   DOWNLOAD OPTION NUMBER: 1
      PATCH FROM LOCATION: >A0
         THROUGH LOCATION: >B0
```

The station number or alias must exist and the patch locations must be valid. The largest number of bytes that can be displayed and modified is 144 (>90). An invalid option number results in the following error message:

```
Station option must be in range 0 - 7
```

An invalid entry for either location value (for example, >A0+2) results in the following error message:

```
Data not of appropriate type
```

After all responses are entered, the specified locations are displayed on the screen. The display in response to the above example input might be as follows:

```
        MODIFY:  STATION  0650,  DOWNLOAD OPTION  1

LOCATION
   00A0  00A6  00D6  0011  0022  0033  0044  0055  0066
         <>____
   00B0  0077  0088  0099  00AA  00BB  00CC  00DD  00EE


Press CMD key when done
```

The cursor is positioned below the contents of the first location (in this case location 00A0) and the message is displayed at the bottom of the screen. Valid entries for new values are in the range 0 through 32767 (decimal) or 0 through FFFF (hexadecimal) (preceded by 0 or >). An invalid entry causes the following message to be displayed:

```
Invalid numeric string
```

After the first location is changed the following options become available:

- Press the TAB SKIP key to move to the next location.
- Pressing the RETURN key to go to the next line and bypassing remaining locations on the same line.
- Press the CMD key if all modifications are entered. This causes the following message to appear:

  ```
  Save these modifications (Yes or No)? <>
  ```

  A Y or YES response results in the following message:

  ```
  Making changes to download file
  ```

  An N or NO response terminates the preprocessor without making any changes to the file.

If the return key is pressed after reading the end of the last line displayed on the screen the cursor moves to the beginning of a blank line. In the above example, this would be addresses >00C0 and above. Any attempt at this time to modify a location results in the following error message:

```
No data to modify, gone to next line
```

This indicates that an attempt was made to modify a location out of the range of the initial request.

Dump the Contents of a Preprocessed File.

```
ENTER COMMAND: DU

      DUMP A PREPROCESSED FILE
        STATION NUMBER OR ALIAS: 0420   (example
         DOWNLOAD OPTION NUMBER: 0       responses)
          DUMP FROM LOCATION: >A0
             THROUGH LOCATION: >D0
         LISTING ACCESS NAME:
```

Perform the dump function as follows:

- Enter valid values for the station number (or alias) and option number. If invalid responses are made, one or more of the following messages appear:

  This alias is not defined

  This station is not defined

  No such option exists for this station

- Specify the contiguous memory area desired for output by entering the first and last locations of the area. Entering invalid location values produces a display of header information only.

- Respond to the prompt for a listing access name with a device name or filename. If the file does not exist, it is created during execution of the command. If the default is taken, the output is displayed on the screen of the terminal.

An example of a file dump for station 0420, option 0 is as follows:

```
        DUMP : STATION 0420, DOWNLOAD OPTION 0

   LOCATION                 MEMORY IMAGE
    00A0      00A6 00D6 0000 0000 0000 0000 0000 0000
    00B0      0000 0000 0000 0000 0000 0000 0000 0000
    00C0      0000 0000 0000 0700 0000 4000 0100 0007
    00D0      0900 0500 000B 04C3 0300 0000 0208 0000
```

The first column lists the location of every 16th byte. The contents of the 16 byte locations starting at 00A0 are listed in order on the first line, the second 16 are listed on the second line, and so on, with each two bytes formatted as one 16-bit word.

5.3 DOWNLOADER UTILITY

The downloader utility accesses files built by the download preprocessor utility and transfers records from these files to a specified secondary station via the network. The utility consists of two parts:

- A user interface which prompts for parameters that identify the secondary station and the file to be downloaded
- A download task which downloads (sends/transmits) the file via the network to the specified secondary station

5.3.1 Downloader Utility Background Information

The downloader utility transfers files from a primary system to a specified secondary station. The files to be transmitted must be in the directory .DOWNLOAD located on the system disk. These files must have been built by the download preprocessor as described in paragraph 5.2. These files may contain operating systems or stand-alone programs for the secondary stations.

NOTE

> The download files contain relocated binary object code that is not properly formatted for TX5 operating system program loaders. The HDLC Communications ROM loader must be used.

The following background information may be helpful to the users of this utility:

- The files accessed by the utility are as follows:
  - .DOWNLOAD.S00nnnn.OPm (see paragraph 5.2.1)
- The default option (download file) for a secondary station is OP0, that is, .DOWNLOAD.S00nnnn.OP0.
- The default option, 0 (zero), is always transmitted by the primary station upon the receipt of a download request from a secondary station.

- Download start and complete messages are written to the system log at the beginning and end of the download process.
- If the downloader is aborted (due to an error from which recovery is not possible), an abort message is written to the system log.

5.3.2 Downloader Utility Execution

This paragraph contains instructions for the activation and termination of the downloader utility,as follows:

- Downloader activation. Downloading can be activated in either of two ways:
  - When a secondary station transmits a request to be downloaded by the primary station
  - When the user has completed the process of identifying the secondary station and download file via the prompts displayed on the screen
- Downloader operation. The download task transfers an object file via the communications line to a secondary station. It uses the secondary ID or alias and option numbers to build the filename, assign a logical unit number (LUNO),and open the file. The task then sends a download command to the selected secondary station to set it to the download state, (that is, ready to receive a transmission from the primary station). The file is then transmitted to the secondary station. When the download operation is complete, the download task closes the file, releases the LUNO, and removes the secondary from the download queue.

As many as 32 secondaries can be downloaded simultaneously.

5.3.3 Downloader Prompts and Responses

The downloader utility presents only two prompts when the SCI command DOWNLOAD is entered, as follows:

```
[]   DOWNLOAD

     DOWNLOAD SECONDARY
         SECONDARY ID OR ALIAS: (nnnn) or (alias)
                 OPTION NUMBER: (m)
```

In response to the first prompt, a number from 0100 through 9999 or an alphanumeric alias name of up to 14 characters is entered. In response to the second prompt, a number from zero through seven is entered, representing a file in .DOWNLOAD.S00nnnn that requires transmission to the previously specified secondary station.

Pressing the CMD key returns control to DX10 SCI.

Several error situations can arise during the interactive process. The messages that may appear and possible remedial actions are described in the following paragraphs.

5.3.3.1 Downloader Error and Procedural Messages.

Two classes of error messages are generated by the downloader utility:

- Messages displayed on the terminal. These relate to operator errors during the interactive process and to failure of entries to the download queue.
- Messages written to the system log. These relate to initialization failures of various types and to secondary station response failures.

The texts for downloader utility messages are listed in Table 5-1. Their meanings and any actions required are also described.

5.3.3.2 Downloader Error Recovery.

Certain remedial actions are recommended when an error is encountered so that the type of problem can be determined. It must be ascertained whether the problem was caused by a hardware failure, an operatior error or a soft error (that is, a transient error or one from which recovery is possible by retry).

Table 5-1 lists and explains the error messages and specifies remedial action where appropriate. When a download operation is aborted, the station is taken offline (no longer polled). The only way to reactivate the station is by executing another DOWNLOAD command.

Table 5-1 Download Messages and Error Recovery

Message Text	Meaning and Probable Cause	Recovery
SECONDARY ID NOT FOUND	The secondary station identity or alias was found to be undefined. This is probably an input error by the operator.	List the download files by using the preprocessor Download (DL) command Refer to paragraph 5.2.3 for details.
OPTION NOT FOUND	Although the secondary station was found, the option file was not. This is probably an input error by the operator.	Same as above.
DOWNLOAD REQUEST CALL FAILED STATUS = xx	An attempt to enter the secondary ID and option in the download queue failed. The download queue was full.	Wait for a download completed message or a download aborted message to appear and then try again. Check the system log.
BID TASK FAILURE	Communications system error. Unable to bid the download task.	Make sure LUNO >90 is assigned to .INDSCOMM.PROGRAM.
UNABLE TO ASSIGN LUNO TO DOWNLOAD FILE	The file required for downloading could not be found or already had a LUNO assigned.	Try again.

Table 5-1 Download Messages and Error Recovery (Continued)

Message Text	Meaning and Probable Cause	Recovery
UNABLE TO OPEN DOWNLOAD FILE	The file required for downloading was found to be already open.	Try again.
COMM SUBSYSTEM ERROR STATUS = (xx) STATION (nnnn)	The secondary station failed to respond in the correct manner. xx is XOP >4D status code.	Check the status of the secondary station. Check the size of the output buffer.
DOWNLOAD STARTED FOR STATION (nnnn) OPTION (m)	Procedural message.	Not applicable.
DOWNLOAD ABORTED FOR STATION (nnnn) OPTION (m)	An irrecoverable error situation was encountered.	Check for messages related to station to determine cause for abort.
DOWNLOAD RESTARTED FOR STATION (nnnn) OPTION (m)	A download request from a secondary station was received while a download operation was in progress.	No action required.
DOWNLOAD COMPLETE FOR STATION (nnnn) OPTION (m)	Procedural message.	Not applicable.
DOWNLOAD TASK COMPLETE	Procedural message. Downloading is complete for all stations.	Not applicable.

5.4 REMOTE SCI UTILITIES

Remote SCI provides a means of interacting with a TX5 secondary station from a terminal at the primary station. Remote SCI is modeled after DX10 SCI and makes it unnecessary for the operator to adjust to TX5 operator communication package (OCP) when debugging secondary station software. The remote SCI tasks consist of two software modules as follows:

* DX10 disk-resident remote SCI tasks which provide the following functions:

  - An interface to an interactive terminal at a DX10-supported primary station where the remote SCI feature has been included

  - An interactive process between the user and the remote SCI facility to acquire the necessary data for execution of the requested commands

  - Interpretation and formatting of the responses before transmission to a secondary station remote SCI task

  - Interpretation of the response from the secondary station

  - Conversion of the response data into a form suitable for output to the specified device

  - Writing the response to the device

  - Writing any error messages to the terminal

* A memory-resident TX5 remote SCI task linked with the appropriate operating system and providing the following functions:

  - Interpretation of the command received from the DX10 remote SCI task

  - Execution of the requested function

  - Return of the results to DX10

  - Return of appropriate error codes to DX10

  - Return to DX10 of TX5 start task and diagnostic task messages and any other messages written to the TX5 log

In using the remote SCI utility, the following should be observed:

- While the remote SCI utility is operative, any DX10 SCI command other than those listed in the remote SCI menu can be entered to initiate a DX10 process (that is, when remote SCI is active, the full set of DX10 commands is supported; but the remote SCI set applies exclusively to TX secondary stations). For example, when the modify memory (MM) command is invoked under remote SCI, the operator is modifying memory in a remote secondary station. When remote SCI is inactive, the operator uses the same command to modify memory in the DX10 primary system.
- Remote SCI can only be used with TX5 secondary stations.
- Remote SCI and OCP functions can both be included in TX systems if desired.
- Remote SCI must be generated for each TX5 secondary station before the secondary station can be linked (see Section 3).
- TX secondary station system log messages are written to the primary station system log when the remote SCI log task and dummy device service routine (DSR) are included in the secondary station system generation (see Appendix C).
- The same remote SCI command cannot be entered simultaneously from two terminals; however, different remote SCI commands can be simultaneously executed from two or more terminals.

5.4.1 Remote SCI Utility Activation and Termination

The remote SCI utility is activated and terminated as follows:

- Remote SCI activation. Remote SCI is activated by entering the SCI command REMSCI and causes the initial command menu to be displayed. This is detailed in the following paragraphs.
- Remote SCI termination. Once the utility is activated it may be terminated at any time by returning to the command mode by entering the Quit (Q) command.

5.4.2 Remote SCI Command Menus

The following paragraphs describe the various menus that appear when remote SCI is requested and the individual commands associated with each menu.

The initial menu that appears when the utility is activated by the REMSCI command is as follows:

```
[] REMSCI

                         TEXAS INSTRUMENTS
                    REMOTE COMMAND INTERPRETER

        SELECT A MENU FROM BELOW:
                     /DEBUG     - DEBUG COMMANDS
                     /TASK      - TASK COMMANDS
                     /LUNO      - LUNO COMMANDS
                     /MISC      - MISCELLANEOUS COMMANDS

<>
```

Each menu selected from the above set displays a different list of available commands. The following menus appear when the above commands are entered. Note that when remote SCI is activated, the <> prompt appears.

5.4.2.1 Debug Command Menu.

When the /DEBUG command is entered, the following menu appears:

```
<> /DEBUG

   REMOTE DEBUG COMMANDS

    AB   - ASSIGN BREAKPOINT      DB   - DELETE BREAKPOINTS
    LB   - LIST BREAKPOINTS       TR   - TRACE LOCATION
    DW   - DUMP WORK SPACE        SV   - SHOW VALUE
    MM   - MODIFY MEMORY          LM   - LIST MEMORY
    MSM  - MODIFY SYSTEM MEMORY   LSM  - LIST SYSTEM MEMORY
    RCRU- READ CRU                WCRU- WRITE CRU

<>
```

The functions of each remote debug command are briefly described below:

* AB. Assigns a breakpoint in the specified secondary station. A maximum of four breakpoints can be assigned in any secondary station.

* DB. Deletes a breakpoint in a remote secondary station or, optionally, deletes all breakpoints in that secondary station.

* DW. Dumps workspace,that is, gets the workspace for a specified secondary task and displays it on the terminal.

* LB. Lists all breakpoints by reading the breakpoint table in the secondary station and displaying it on the terminal.

* LM. Lists memory from specified secondary task locations, using relative addressing.

* LSM. Lists system memory for specified locations, using absolute addressing.

* MM. Modifies memory at specified secondary task locations, using relative addressing.

* MSM. Modifies system memory at specified secondary station locations, using absolute addressing.

* RCRU. Reads a specified CRU location in the secondary station and displays it on the terminal.

* SV. Returns the results of an expression entered. The show value function is actually performed at the DX10 and is included to provide a procedure for performing mathematical functions.

* TR. Displays the contents of a specified location on the front panel indicators at the secondary station.

* WCRU. Writes a set of data bits to a specified CRU location in the secondary station.

5.4.2.2 Task Command Menu.

When the /TASK command is entered, the following menu appears:

```
    <> /TASK

       REMOTE TASK COMMANDS

              XT  - EXECUTE TASK           KT - KILL TASK

              STS - SHOW TASK STATUS       IT - INSTALL TASK

              NID - NETWORK ID/TASK ID

    <>
```

The task commands shown are briefly described below:

- IT. Installs a task in the secondary station by sending it a DX10-resident object file. The object file is transmitted one record at a time until complete. The object code can be in either compressed or ASCII format. The object file is not preprocessed by the IT command.
- KT. Kills a task in the secondary station.
- STS. Shows task status, that is, either displays the status of the specified task or, optionally, all tasks in the specified secondary station.
- XT. Initiates the execution of a task in the specified secondary station.
- NID. Shows the relationship between TX5 task IDs and network IDs.

5.4.2.3 LUNO Command Menu.

When the /LUNO command is entered, the following menu appears:

```
    <> /LUNO

        REMOTE LUNO COMMANDS

              AL  - ASSIGN LUNO           RL - RELEASE LUNO
              SIS - SHOW I/O STATUS

    <>
```

The functions of the remote LUNO commands are briefly described below:

- AL. Assigns a LUNO to a device or file in the specified secondary station.
- RL. Releases a specified LUNO in the secondary station.
- SIS. Shows the input/output status of a specified secondary station and a specified LUNO, or alternatively, all LUNOs in that secondary station.

5.4.2.4 Miscellaneous Command Menu.

When the /MISC command is entered, the following menu appears:

```
<>/MISC

REMOTE MISCELLANEOUS COMMANDS

IDT -INITIALIZE DATE AND TIME    ERRS -COMMUNICATIONS ERROR LIST

SDT -SHOW DATE AND TIME          Q    -QUIT REMOTE SCI

<>
```

The functions of the miscellaneous commands available with remote SCI are briefly described below:

- IDT. Initializes date and time in the specified secondary station. The primary station date and time is transmitted to the secondary station.
- Q. Quits (terminates) execution of the remote SCI utility.
- SDT. Shows the date and time of the secondary station clock.

- * ERRS. Provides file pathnames for the display of communication error codes and information.

5.4.3 Remote SCI Commands

Any command listed in the preceding paragraphs may be executed after entering the SCI command REMSCI at a DX10 interactive terminal. To execute a particular command, enter the command name and press the RETURN key. Each command has a set of prompts to which you must respond; these are summarized in Table 5-2. Further details of the responses to remote SCI command prompts are provided below.

NOTE

> Unless otherwise stated, all numbers entered in response to a remote SCI prompt can be entered as a decimal value or a hexadecimal value. Hexadecimal values must be preceded by a greater than sign (>) or a zero (0).

- * SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station. The network ID entered for this prompt is the network ID assigned when the program is executed.
- * TASK ID. Enter the hexadecimal number that was assigned to the task in the TX5 system generation.
- * ADDRESS. Enter the address of the word desired.
- * LUNO. Enter a valid LUNO assignment.
- * ACCESS NAME. Enter a valid filename or device name.
- * OBJECT FILENAME. Enter the pathname or synonym of a DX10 object file.
- * PRIORITY. Enter a number from 0 through 4.
- * PRIVILEGED. Enter Y (yes) or N (no) to indicate to TX5 whether the task is required to execute as a privileged or nonprivileged task.
- * NUMBER OF BYTES. Enter the number of bytes to be accessed or displayed.

- LISTING ACCESS NAME. Enter a character string representing a valid device or file. Accepting the default causes the listing to appear at the terminal where the request was made.

- CRU BASE. Enter a valid CRU base address for the secondary station.

- DATA. Enter an integer value.

- NUMBER OF BITS. Enter a number from 0 through 15 (>F).

- PARM1. Enter a number from 0 through 65535.

- PARM2. Enter a number from 0 through 65535.

If an invalid response is made to any prompt, an error message appears on the screen. For example, entering a task ID of >FFF causes the response to be retained and the error message INVALID TASK ID to appear. Similarly, entering a value of 99999 in response to a PARM1 or PARM2 prompt causes the message INVALID PARAMETER to appear.

The following paragraphs provide a detailed description of each remote SCI command. They include example responses and point to appropriate error reporting and recovery information. The remote SCI commands are listed in alphabetical order (that is, independent of the order in which they appear in the menus). Error messages associated with the commands are described in Table 5-3.

Table 5-2 Summary of Remote SCI Commands and Prompts

Command Mnemonic	Prompt Number				
	1	2	3	4	5
AB	↑	TASK ID (d)	ADDRESS	(none)	
AL	SECONDARY ID	LUNO	NAME	(none)	
DB		TASK ID (d)	ADDRESS (d)	(none)	
DW	↓	TASK ID	(none)		
ERRS	ERROR TYPE (d)	(none)			
IDT	↑	(none)			
IT		OBJECT FILENAME	TASK ID (d)	PRIORITY (d)	PRIVILEGED (d)
KT		TASK ID	(none)		
LB		TASK ID (d)	(none)		
LM		TASK ID (d)	ADDRESS	# BYTES (d)	LISTING NAME (d)
LSM		ADDRESS	# BYTES (d)	LISTING NAME (d)	(none)
MM	SECONDARY ID	TASK ID (d)	ADDRESS	(none)	
MSM		ADDRESS	(none)		
NID		LISTING NAME (d)	(none)		
RCRU		CRU BASE	(none)		
RL		LUNO	(none)		
SDT		(none)			
SIS		LUNO (d)	(none)		
STS	↓	TASK ID (d)	LISTING NAME (d)	(none)	
SV	EXPRESSION	(none)			
TR	↑	TASK ID (d)	ADDRESS (d)	(none)	
WCRU	SECONDARY ID	CRU BASE	DATA	# BITS	(none)
XT	↓	TASK ID	PARM1 (d)	PARM2 (d)	

Note: (d) = default may be taken

5.4.3.1 Assign Breakpoint (AB) Command.

The AB command assigns a breakpoint in a specified secondary station task. The prompts and example responses for the command are as follows:

```
<>    AB

      REMOTE ASSIGN BREAKPOINT
            SECONDARY ID:  1002       (example
                 TASK ID:  >3A         responses)
                 ADDRESS:  >01B5
```

The responses must conform to the following requirements:

* SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.

* TASK ID. Enter the hexadecimal number assigned to the task at the time of TX5 system generation. If the default is taken (by pressing the RETURN key), or an S is entered, then the address entered is considered to be an absolute address in the secondary station.

NOTE

> Assigning a breakpoint to a task with a priority of 0 or 1, or to a system module (such as a DSR) inhibits communications at the secondary station.

* ADDRESS. If a task ID was entered, the number entered is considered relative to the starting address of the task in the secondary station. If the default was taken for the task ID or an S was entered, the address must be an absolute address in the secondary station.

5.4.3.2 Assign LUNO (AL) Command.

The AL command assigns a LUNO to a specified device or file at the secondary station. The prompts and example responses for the command are as follows:

```
   <>    AL

         REMOTE ASSIGN LUNO
                 SECONDARY ID: 3124      (example
                        LUNO: >17         responses)
                  ACCESS NAME: LP01
```

The responses must conform to the following requirements:

- SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.
- LUNO. Enter the logical unit number.
- ACCESS NAME. This is the access name of the file or device in the secondary station to which the LUNO is to be assigned.

NOTE

The TX5 system must include file management and file utility modules to support this command. Refer to Appendix C for further instructions.

5.4.3.3 Delete Breakpoint (DB) Command.

The DB command deletes one or all breakpoints in a specified secondary station or in a specified task. The prompts and example responses for the command are as follows:

```
    <>    DB

          REMOTE DELETE BREAKPOINT
                SECONDARY ID: 1021        (example
                     TASK ID: >4E          responses)
                     ADDRESS: (default)
```

The responses must conform to the following requirements:

* SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary.

* TASK ID. Enter the hexadecimal number assigned to the task at the time of TX5 system generation. If the default is taken by pressing the RETURN key, or S is entered, any breakpoint address entered is considered absolute.

* ADDRESS. Enter the address of the breakpoint. If the default is taken, all breakpoints assigned to the specified task, or to the system (depending on the response to the TASK ID prompt) are deleted.

5.4.3.4 Dump Workspace (DW) Command.

The DW command displays the workspace for a specified task resident in a specified secondary station. The prompts and example responses for the command are as follows:

```
    <>    DW

          REMOTE DUMP WORKSPACE
                 SECONDARY ID: 1099       (example
                      TASK ID: >1A         responses)
```

The responses must conform to the following requirements:

* SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.

* TASK ID. Enter the hexadecimal number assigned to the task at the time of TX5 system generation.

5.4.3.5 Display Error Codes (ERRS)Command..

The ERRS command displays the error codes used by the HDLC software package. When the command is entered while remote SCI is active, the following prompt appears:

```
    <> ERRS

       DISPLAY ERROR CODES FOR TYPES: SVC, UL, AS, WT, PK, LC
            ERROR TYPES: SVC    (example response)
```

The response to the ERROR TYPES prompt must be one of the options displayed. This results in the display of a file containing error code information for one of the following error types:

* SVC. Error codes returned by the SVC >4D processor when an application task makes read, write, or activation services calls

* UL. Upper-level protocol error codes

* AS. Activation services error codes

* WT. Warm-start/timer error codes

* PK. Packet-level error codes

* LC. Link control error codes

Refer to Section 4 for an explanation of the SVC error codes; refer to Appendix A for explanations of the others.

5.4.3.6 Initialize Date and Time (IDT) Command.

The IDT command initializes the date and time in the specified secondary station by transmitting the date and time in the DX10. The prompts and example responses for the command are:

```
    <>  IDT

          REMOTE INITIALIZE DATE AND TIME
                SECONDARY ID: 1002      (example response)
```

Enter the network ID assigned to the remote SCI task in the desired secondary station in response to the SECONDARY ID prompt.

5.4.3.7 Install Task (IT) Command.

The IT command installs a specified task in a specified secondary station by downloading a specified DX10-resident object file. The prompts and example responses for the command are as follows:

```
<>    IT

        REMOTE INSTALL TASK
                SECONDARY ID: 5099
            OBJECT FILENAME: .SEC99.GETDATA    (example
                    TASK ID: >10                responses)
                   PRIORITY: 4
                 PRIVILEGED: N
```

The responses must conform to the following requirements:

- SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.
- OBJECT FILENAME. The response to this prompt is a character string which is the name of a DX10 object file. The object code can be in either ASCII or compressed format. It is suggested that compressed format be used for faster loading.
- TASK ID. The response to this prompt must be >10.
- PRIORITY. This is a number from 0 through 4. Priorities 0 and 1 should be used with due regard to other DX10 system requirements. The default value is 4.
- PRIVILEGED. The response to this must be either Y (yes) or N (no) and indicates to TX5 whether the task is required to execute as a privileged or nonprivileged task. The default is NO.

The error messages associated with this command are described in Table 5-3.

5.4.3.8 Kill Task (KT) Command.

The KT command terminates the execution of a specified task in a specified secondary station. The prompts and example responses for the command are as follows:

```
<>    KT

      REMOTE KILL TASK
             SECONDARY ID: 1012        (example
                  TASK ID: >2B          responses)
```

The responses must conform to the following requirements:

* SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.

* TASK ID. Enter the hexadecimal number assigned to the task at the time of TX5 system generation.

5.4.3.9 List Breakpoints (LB) Command.

The LB command causes all breakpoints assigned in a specified secondary task to be displayed on the interactive terminal where the command was entered. The prompts and example responses for the command are as follows:

```
<>    LB

      REMOTE LIST BREAKPOINTS
           SECONDARY ID: 1021          (example
                TASK ID: >3D            responses)
```

The responses must conform to the following requirements:

* SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary.

* TASK ID. Enter the hexadecimal number assigned to the task at the time of TX5 system generation. Alternatively, the default can be taken. If the default is taken, all breakpoints in the secondary station are displayed. If S is entered, only the system breakpoints appear.

5.4.3.10 List Memory (LM) Command.

The LM command lists the contents of specified memory addresses within a specified secondary task. The command uses relative addressing. The prompts and example responses for the command are as follows:

```
   <>   LM

        REMOTE LIST MEMORY
                 SECONDARY ID: 1002
                      TASK ID: >3D          (example
             STARTING ADDRESS: >10F          responses)
              NUMBER OF BYTES: (default)
          LISTING ACCESS NAME: LP03
```

The responses must conform to the following requirements:

* SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.

* TASK ID. Enter the hexadecimal number assigned to the task at the time of TX5 system generation. If S or a RETURN is entered for the task ID, the starting address is relative to address >0 in the secondary station (that is, the command is then the same as the List System Memory (LSM) command).

* STARTING ADDRESS. This is the address relative to the starting address of the task in the secondary station. If S or a RETURN was entered in reponse to the TASK ID prompt, the address is interpreted as absolute.

* NUMBER OF BYTES. Enter the number of bytes to be listed, or take the default. If the default is taken (by pressing the RETURN key), then 16 bytes of memory are displayed.

* LISTING ACCESS NAME. This is a listing access name and must be a character string representing a primary station device or a filename. The default can be taken by pressing the RETURN key. This causes the listing to be written to the display where the request was made.

5.4.3.11 List System Memory (LSM) Command.

The LSM command displays the contents of specified memory locations in a specified secondary station. The command uses absolute addressing. The prompts and example responses for the command are as follows:

```
<>    LSM

      REMOTE LIST SYSTEM MEMORY
                SECONDARY ID: 1002
           STARTING ADDRESS: >00A0        (example
            NUMBER OF BYTES: 8             responses)
        LISTING ACCESS NAME: (default)
```

The responses must conform to the following requirements:

* SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.

* STARTING ADDRESS. This is an absolute address in the secondary station.

* NUMBER OF BYTES. Enter the number of bytes to be listed.

* LISTING ACCESS NAME. This is a listing access name and must be a character string. The default can be taken by pressing the RETURN key. This causes the listing to be written to the display where the request was made.

5.4.3.12 Modify Memory (MM) Command.

The MM command allows the modification of specified memory locations within a specified secondary task. The command uses relative addressing and displays eight locations and their contents. When a new value and/or a RETURN is entered, the next address and its contents are displayed on the terminal. The prompts and example responses for the command are as follows:

```
<>    MM

      REMOTE MODIFY MEMORY
           SECONDARY ID: 1002        (example
                TASK ID: >2C          responses)
                ADDRESS: >1FE

      01FE        0430             (press RETURN)
      0200        0C61             (press RETURN)
      0202        0101
       :           :
      020A        031C
      020C        0010             (press CMD)
```

The responses must conform to the following requirements:

- SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.
- TASK ID. Enter the hexadecimal number assigned to the task at the time of TX5 system generation. A response of S or a RETURN signifies that the address entered in response to the ADDRESS prompt is absolute.
- ADDRESS. This is the address of the location to be displayed and is relative to the starting address of the task in the secondary station. It is the first address displayed. If S or a RETURN was entered in response to the TASK ID prompt, the value entered is interpreted as absolute.

The modification is made by positioning the cursor at the desired location on the screen. Pressing RETURN moves the cursor to the next location. Pressing CMD causes the values on the screen to be returned to the secondary station and written to memory.

5.4.3.13 Modify System Memory (MSM) Command.

The MSM command differs from the MM command only in the respect that absolute addressing is used. A specified secondary address and its contents is displayed on the interactive terminal where the request was entered. If a change is required, the new value is entered followed by a RETURN (to move to the next location) or a CMD (to exit from the command). The prompts and example responses for the command are as follows:

```
    <>   MSM

         REMOTE MODIFY SYSTEM MEMORY
              SECONDARY ID: 1002        (example
                   ADDRESS: >01FE        responses)

         01FE       5001                (press RETURN)
           :          :
           :          :
         020C       0CD3                (press CMD)
```

The responses must conform to the following requirements:

* SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.

* ADDRESS. This is the address of the location that is to be changed and is absolute, that is, referenced to address 0.

5.4.3.14 Associate Network ID/Task ID (NID) Command.

The NID command provides a list of network IDs (NIDs) and their associated task IDs for a specified secondary station. The prompts and example responses for this command are as follows:

```
<> NID

    GET NETWORK ID / TX5 TASK ID
               SECONDARY ID: 2100
        LISTING ACCESS NAME:
```

The responses must conform to the following requirements:

- * SECONDARY ID. Enter the network ID assigned to the remote SCI task in the selected secondary station.
- * LISTING ACCESS NAME. Enter the synonym or name of the file or device that is to receive the data returned from the secondary station.

An example display is as follows:

```
                SECONDARY ID = 2100

         TASK ID                NETWORK ID

           >15                    2100
           >16                    2101
           >10                    2102
           >45                    2103
```

The task IDs are listed in the same sequence as they were entered during network generation.

5.4.3.15 Read CRU (RCRU) Command.

The RCRU command reads 16 bits of data from a specified secondary CRU register address and displays it on the interactive terminal where the request was entered. It is displayed as four hexadecimal digits. The prompts and example responses for the command are as follows:

```
<>    RCRU

      REMOTE READ CRU
            SECONDARY ID: 1002      (example
                CRU BASE: >80        responses)


       CRU REGISTER VALUE: >FF00     (displayed after read)
```

The responses must conform to the following requirements:

- SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.
- CRU BASE. This is the CRU address offset from which to read 16 bits of data.

5.4.3.16 Release LUNO (RL) Command.

This command releases a specified secondary LUNO. The prompts and example responses for the command are as follows:

```
<>    RL

      REMOTE RELEASE LUNO
         SECONDARY ID: 1012      (example
                LUNO: >2C         responses)
```

The responses must conform to the following requirements:

- SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.
- LUNO. Enter the logical unit number.

NOTE

> The TX5 system must include file management and file utility modules. Refer to Appendix C for instructions.

5.4.3.17 Show Date and Time (SDT) Command.

The SDT command displays (on the interactive terminal where the request was entered) the specified secondary clock. The prompts and example responses for the command are as follows:

```
    <>    SDT

          REMOTE SHOW DATE AND TIME
                 SECONDARY ID: 1012   (example response)


       1980/07/30 09:06:03
```

Enter the network ID assigned to the remote SCI task in the desired secondary station in response to the SECOND SECONDARY ID prompt.

5.4.3.18 Show Input/Output Status (SIS) Command.

The SIS command displays (on the interactive terminal where the request was entered) the input/output status of the specified device LUNO. Alternatively, the input/output status of all LUNOs can be displayed. The prompts and example responses for the command are as follows:

```
    <>    SIS

          REMOTE SHOW I/O STATUS
                SECONDARY ID: 1002     (example
                        LUNO: >3F       responses)
```

The responses must conform to the following requirements:

- SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.
- LUNO. Enter the logical unit number, or take the default by pressing the RETURN key. If the default is taken, the status of all device LUNOs is displayed.

5.4.3.19 Show Task Status (STS) Command.

The STS command displays the status of one or all specified secondary tasks. The prompts and example responses for the command are as follows:

```
    <>    STS

          REMOTE SHOW TASK STATUS
                        SECONDARY ID: 1011       (example
                             TASK ID: >5A         responses)
               LISTING ACCESS NAME: (default)
```

The responses must conform to the following requirements:

- SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.
- TASK ID. Enter the hexadecimal number assigned to the task at the time of TX5 system generation or take the default (by pressing the RETURN key). If the default is taken, the status of all tasks in the specified secondary station is displayed.
- LISTING ACCESS NAME. This must be a device name, synonym, or filename. The default can be taken by pressing the RETURN key. This causes the listing to be written to the terminal at which the request was made.

5.4.3.20 Show Value (SV) Command.

The SV command evaluates a specified expression and displays the results on the DX10 interactive terminal where the request was entered. The hexadecimal, decimal, and (if applicable) ASCII values of the input expression are displayed. The prompts and example responses for the command are as follows:

```
   <>    SV

         SHOW VALUE
                                             (example
                  EXPRESSION: >FF+>A5           responses)


         HEX: >000001A4    DECIMAL: 420    ASCII´....´
```

In response to the EXPRESSION prompt, enter the expression requiring evaluation. If the expression is a hexadecimal number, it must be preceded by a > sign. If it is an ASCII expression, it must be delimited by apostrophes, for example, ´expression´.

5.4.3.21 Trace (TR) Command.

The TR command displays (on the 990/5 front panel indicators at a specified secondary station) the contents of a specified address. If the task is the TX5 operating system, the address is interpreted as an absolute address (that is, referenced to address 0 in the secondary station). The prompts and example responses for the command are as follows:

```
<>   TR

     REMOTE TRACE
        SECONDARY ID: 1002        (example
             TASK ID: >5F          responses)
             ADDRESS: >100
```

The responses must conform to the following requirements:

* SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.

* TASK ID. Enter the hexadecimal number assigned to the task at the time of TX5 system generation, or take the default (by pressing the RETURN key). If S is entered or the default is taken, the specified address is interpreted as absolute.

* ADDRESS. Enter the address to be displayed. If no address is entered, the trace is removed.

5.4.3.22 Write CRU (WCRU) Command.

The WCRU command writes as many as 16 data bits to a specified CRU register address. The prompts and example responses for the command are as follows:

```
<>    WCRU

      REMOTE WRITE CRU
             SECONDARY ID: 1002
                 CRU BASE: >200         (example
        VALUE TO BE WRITTEN: >A5A5        responses)
     NUMBER OF BITS (0..>F): (default)
```

The responses must conform to the following requirements:

- SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.
- CRU BASE. Enter the CRU address to which the data is to be written.
- VALUE TO BE WRITTEN. Enter the integer value required to be written to the CRU address specified.
- NUMBER OF BITS (0..>F). Enter the number of bits to be written. This is a number from 0 to 15 (>F). The default is 0, which indicates 16 bits.

5.4.3.23 Execute Task (XT) Command.

The XT command executes a specified secondary task and may pass up to two parameters to the program. The prompts and example responses for the command are as follows:

```
<>   XT

     REMOTE EXECUTE TASK
            SECONDARY ID: 1002
                 TASK ID: 010          (example
                   PARM1: 0             responses)
                   PARM2: 0
```

The responses must conform to the following requirements:

- SECONDARY ID. Enter the network ID assigned to the remote SCI task in the desired secondary station.
- TASK ID. Enter the hexadecimal number assigned to the task at the time of TX5 system generation. The default is >10.
- PARM1 and PARM2. The entries must be in the range of 0 through 65535 and represent values to be passed to the program. One or both of them may be defaulted to zero by pressing the RETURN key.

5.4.4 Remote SCI Error Reporting and Recovery

Error messages associated with the remote SCI utility occur as a result of operator input error or an unexpected communications or operating system condition. The following situations result in the generation of error messages:

* Entering a command that has not been linked with the specified secondary operating system

* Entering incorrect numerical data

* Encountering an error in a response

* Encountering an incompatability when execution is attempted

* Encountering an error that occurred while the specified operation was being executed

* Communications errors

* SVC errors in a specified secondary station

* Invalid key strokes by the operator

Table 5-3 lists the errors by command group and specifies recommended remedial action.

Table 5-3 Remote SCI Messages and Error Recovery

Function and Message Text	Meaning and Probable Cause	Recovery
Breakpoint Functions (AB/DB):		
NO TSB FOR THAT TASK FOUND IN THE SECONDARY	No task status block (TSB) was found; the task specified is not installed in the specified secondary station.	Ensure that the task ID and secondary ID are entered correctly.
BP FUNCTION NOT INCLUDED IN THE SECONDARY	Breakpoint functions were not included when remote SCI was built for this secondary station.	Rebuild remote SCI for this secondary station.
BREAKPOINT TABLE FULL IN SECONDARY	At least four breakpoints are already assigned in this secondary station.	Delete at least one breakpoint in this secondary station.
BREAKPOINT ALREADY SET	Request was made to assign a breakpoint where one is already set.	None required.
UNKNOWN BREAKPOINT ADDRESS	The breakpoint address requested is invalid.	Reenter breakpoint request.

Table 5-3 Remote SCI Messages and Error Recovery (Continued)

Function and Message Text	Meaning and Probable Cause	Recovery
List Memory Functions (LM/LSM):		
LM AND LSM FUNCTIONS NOT INCLUDED IN SECONDARY		
	List memory functions were not included when remote SCI was built for this secondary station.	Rebuild remote SCI for this secondary station.
NO TSB FOR THAT TASK FOUND IN THE SECONDARY		
	The task specified is not installed in the specified secondary station.	Ensure that the task ID and secondary ID are entered correctly.
Task Control Functions (XT/KT):		
XT AND KT FUNCTIONS NOT INCLUDED IN SECONDARY		
	The task control functions were not included when SCI was built for this secondary station.	Rebuild remote SCI for this secondary station.
XT OR KT SVC FAILURE IN THE SECONDARY		
	An error occurred when an SVC call was made in the secondary station to perform the requested function.	Check all entries for validity and repeat the operation.
NO TSB FOR THAT TASK FOUND IN THE SECONDARY		
	The task specified is not installed in the specified secondary station.	Ensure that the task ID and secondary ID are entered correctly.

Table 5-3 Remote SCI Messages and Error Recovery (Continued)

Function and Message Text	Meaning and Probable Cause	Recovery
Dump Workspace Function (DW):		
NO TSB FOR THAT TASK FOUND IN THE SECONDARY	The task specified is not installed in the specified secondary station.	Ensure that the task ID and secondary ID are entered correctly.
DW FUNCTION NOT INCLUDED IN THE SECONDARY	The specified function was not included when SCI was built for this secondary station.	Rebuild remote SCI for this secondary station.
List Breakpoints Function (LB):		
LB FUNCTION NOT INCLUDED IN THE SECONDARY	The specified function was not included when SCI was built for this secondary station.	Rebuild remote SCI for this secondary station.
NO BP FOR THAT TASK IN SECONDARY	There are no breakpoints assigned to the task ID entered in the LB command.	Ensure that the task ID and secondary ID are entered correctly.
NO TSB FOR THAT TASK FOUND IN THE SECONDARY	The task specified is not installed in the specified secondary station.	Ensure that the task ID and secondary ID are entered correctly.
NO BREAKPOINTS ASSIGNED IN THE SECONDARY	There are no breakpoints assigned in the specified secondary station.	Check the secondary ID and try again.

Table 5-3 Remote SCI Messages and Error Recovery (Continued)

Function and Message Text	Meaning and Probable Cause	Recovery
Trace Memory Location (TR):		
NO TSB FOR THAT TASK FOUND IN THE SECONDARY	The task specified is not installed in the specified secondary station.	Ensure that the task ID and secondary ID are entered correctly.
TR FUNCTION NOT INCLUDED IN THE SECONDARY	The specified function was not included when SCI was built for this secondary station.	Rebuild remote SCI for this secondary station.
Show Status of Tasks (STS):		
NO TSB FOR THAT TASK FOUND IN THE SECONDARY	The task specified is not installed in the specified secondary station.	Ensure that the task ID and secondary ID are entered correctly.
STS FUNCTION NOT INCLUDED IN THE SECONDARY	The specified function was not included when SCI was built for this secondary station.	Rebuild remote SCI for this secondary station.

Table 5-3 Remote SCI Messages and Error Recovery (Continued)

Function and Message Text	Meaning and Probable Cause	Recovery
LUNO Functions (AL/RL):		
ASSIGN LUNO SVC ERROR IN SECONDARY	An error occurred when an attempt was made to assign the requested LUNO.	Enter the AL command again after checking the LUNO.
RELEASE LUNO SVC ERROR IN SECONDARY	An error ocurred when an attempt was made to release the requested LUNO.	Enter the RL command again after checking the LUNO.
LUNO FUNCTIONS NOT INCLUDED IN THE SECONDARY	The specified function was not included when SCI was built for this secondary station.	Rebuild remote SCI for this secondary station.
Initialize Date and Time (IDT):		
SVC ERROR IN SECONDARY	An error occurred when an attempt was made to set the date and time at the secondary station.	Check the secondary ID and try the IDT command again.
IDT FUNCTION NOT INCLUDED IN THE SECONDARY	The specified function was not included when SCI was built for this secondary station.	Rebuild remote SCI for this secondary station.

Table 5-3 Remote SCI Messages and Error Recovery (Continued)

Function and Message Text	Meaning and Probable Cause	Recovery
Install Task Function (IT):		
ASSIGN LUNO ERROR		
	An error occurred when an attempt was made to assign a LUNO to the specified DX10 file.	Enter the IT command again after checking that the filename is correct and that it is an object file.
OPEN LUNO ERROR		
	An error occurred when an attempt was made to open the DX10 file specified by the user.	Enter the IT command again after checking that the filename is correct.
READ FILE ERROR		
	An error occurred when an attempt was made to read the DX10 file specified by the user.	Enter the IT command again after checking that the filename is correct.
INSTALL TASK FUNCTION NOT INCLUDED IN THE SECONDARY		
	The specified function was not included when SCI was built for this secondary station.	Rebuild remote SCI for this secondary station.
INSTALL TASK ALREADY IN PROGRESS IN SECONDARY		
	The remote SCI function is in use at some other DX10 terminal.	Try again.
BAD OBJECT FORMAT		
	The object file specified is not in the expected format.	Enter the IT command again after checking that the file is in DX10 object format.

Table 5-3 Remote SCI Messages and Error Recovery (Continued)

Function and Message Text	Meaning and Probable Cause	Recovery
NO TSB FOUND FOR THE TASK ID REQUESTED	No TSB exists to associate with the task ID specified for installation. Only one dynamic task can be installed remotely.	Insure task >10 is included in the TX5 system generation.
SECONDARY MEMORY INSUFFICIENT TO INSTALL TASK	The model 990/5 computer memory is insufficient to load the requested program.	Increase the size of the memory or reduce the program size.
CHECKSUM ERROR IN TRANSMITTED OBJECT RECORD	A redundancy checksum error occurred in the secondary station during the processing of an object record.	Try again.
SECONDARY TASK IS ACTIVE	The task ID specified in IT command is still active.	Kill the task and retry the IT function.
UNSPECIFIED ERROR - TRY AGAIN	An unidentifiable error occurred.	The responses to the prompts are possibly incorrect, that is, the secondary ID or task ID is valid but not correct for the IT function. Check the entries and retry. If the problem persists, rebuild the TX5 and download it.

Table 5-3 Remote SCI Messages and Error Recovery (Continued)

Function and Message Text	Meaning and Probable Cause	Recovery
Modify Memory Functions (MM/MSM):		
MM OR MSM FUNCTION NOT INCLUDED IN THE SECONDARY	The specified function was not included when SCI was built for this secondary station.	Rebuild remote SCI for for this secondary station.
NO TSB FOR THAT TASK FOUND IN THE SECONDARY	The task specified is not installed in the specified secondary station.	Ensure that the task ID and secondary ID are entered correctly.
WHAT DID YOU ENTER?	The key depressed at the interactive terminal cannot be interpreted by the MM/MSM command processor.	Make the entry again. If the problem recurs, do not reenter the function or command at this terminal.
INVALID VALUE INPUT	An invalid hexadecimal or decimal value was entered while attempting to modify a memory location.	Retry again using a valid value.
CANNOT ASSIGN LUNO TO DEVICE	An error occurred during an attempt to assign a LUNO to the interactive device in use.	Try again. If the error recurs, retry at another interactive terminal.
CANNOT OPEN LUNO TO DEVICE	An error occurred during an attempt to open the interactive device in use.	Try again. If the error recurs, retry at another interactive terminal.

Table 5-3 Remote SCI Messages and Error Recovery (Continued)

Function and Message Text	Meaning and Probable Cause	Recovery
DEVICE STATUS ERROR	An error occurred during a read device status call by the modify memory program.	Try again. If the error recurs, retry at another interactive terminal.
DEVICE CLEAR ERROR	An error occurred during a clear screen call by the modify memory program.	Try again. If the error recurs, retry at another interactive terminal.
DEVICE INPUT/OUTPUT ERROR (WRITE)	An error occurred during a call to write data to the interactive device in use.	Try again. If the error recurs, retry at another interactive terminal.
DEVICE INPUT/OUTPUT ERROR (READ)	An error occurred during a call to read data from the interactive device in use.	Try again. If the error recurs retry at another interactive terminal.

Table 5-3 Remote SCI Messages and Error Recovery (Continued)

Function and Message Text	Meaning and Probable Cause	Recovery
Read/Write CRU Functions (RCRU/WCRU):		
WCRU OR RCRU FUNCTION NOT INCLUDED IN THE SECONDARY	The specified function was not included when SCI was built for this secondary station.	Rebuild remote SCI for this secondary station.
WRITE CRU ERROR	An error occurred during an attempt to write the specified value to the CRU location in the specified secondary station.	Check that the value, the CRU address, and the secondary ID are correct and retry.
READ CRU ERROR	An error occurred during an attempt to read a value from the CRU location in the specified secondary station.	Check that the value, the CRU address, and the secondary ID are correct and retry.
CRU REGISTER DATA IS (nnnn)	This is a normal return when CRU read is requested.	Interpret the value (nnnn) as hexadecimal.

Table 5-3 Remote SCI Messages and Error Recovery (Continued)

Function and Message Text	Meaning and Probable Cause	Recovery
Show Date and Time Function (SDT):		
SVC ERROR IN SECONDARY	An error occurred in the specified secondary when the Get Date and Time SVC call was made.	Check the TX5 link map to ensure that the TX5 linking is correct. If required, regenerate the TX5 system and relink.
SDT FUNCTION NOT INCLUDED IN THE SECONDARY	The specified function was not included when SCI was built for this secondary station.	Rebuild remote SCI for this secondary station.
Show Input/Output Status (SIS):		
SIS FUNCTION NOT INCLUDED IN THE SECONDARY	The specified function was not included when SCI was built for this secondary station.	Rebuild remote SCI for this secondary station.
LUNO NOT ASSIGNED IN THE SECONDARY	The LUNO specified in the request is not assigned in the specified secondary station.	Ensure that the LUNO entered was correct and retry.

Table 5-3 Remote SCI Messages and Error Recovery (Continued)

Function and Message Text	Meaning and Probable Cause	Recovery
Network ID / Task ID Association:		
NID FUNCTION NOT INCLUDED IN THE SECONDARY	The NID function was not included when SCI was built for this secondary station.	Rebuild remote SCI for this secondary station.
NO NETWORK TABLES FOUND IN SECONDARY	The remote SCI task is not able to access the network tables.	Rebuild the TX5 system and download.
NID DATA FROM SECONDARY REMOTE SCI IS INCORRECT	The remote SCI task is returning erroneous data from the secondary.	Retry. If the problem persists rebuild the TX5 and download.

Table 5-3 Remote SCI Messages and Error Recovery (Continued)

Function and Message Text	Meaning and Probable Cause	Recovery
Communications Interface Errors:		
INVALID SECONDARY ID	The user entered a secondary ID number that is not in the network.	Reenter the secondary ID and retry.
HDLC COMMUNICATIONS WRITE ERROR =	This message is followed by a status code normally returned in byte 1 of the SVC block when a write call is made. (See status byte codes below.)	Ensure that the secondary ID value was entered correctly and retry.
HDLC COMMUNICATIONS READ ERROR =	This message is followed by a status code normally returned in byte 1 of the SVC block when a read call is made. (See status byte codes below.)	Ensure that all information was entered correctly and retry. If the secondary station failed during the remote SCI session, this error error will eventually recur.
HDLC COMMUNICATIONS WAKE-UP ERROR =	This message is followed by a status code normally returned in byte 1 of the SVC block when a request for activation services call is made. (See status byte codes below.)	Ensure that all information was entered correctly and retry. If the problem persists, the activation services queue may be full. The communications system must be rebuilt to enlarge the size of the queue.

Table 5-3 Remote SCI Messages and Error Recovery (Continued)

Status Byte Codes Returned by Communications Interface Errors:

1 = Function not implemented.
2 = Illegal I/O opcode.
3 = Invalid source ID.
4 = No input message queued for caller (read calls).
5 = No buffers available in the system to queue the caller´s output message, activation services requests, or download requests.
6 = Communications system not active.
7 = (Not returned).
8 = Undefined destination ID.
9 = Output buffer length too large.
A = Destination specified by destination ID is inoperative. This error code is also returned if a download is in progress to the secondary addressed by a caller task.
B = No buffers available in the channel to queue the caller´s output message.

The status byte codes listed above are SVC >4D errors. They can also be displayed under remote SCI by entering the command ERRS (see paragraph 5.4.3.5).

5.5 STATISTICS PACKAGE

The statistics utility collects data and allows the user to view the condition of the communications system. Certain memory areas in the FCCC and the mainframe are used to store statistical data. The data stored is reset at communications system warm-start time or at any time desired by the operator. It is possible to accumulate statistical information for any period of time desired. The purpose of the package is to provide data for efficient network management, such as load distribution. The ease of operation of the software makes it simple to determine whether any changes in the network configuration have affected network performance positively or negatively.

The statistical data accumulated includes the following items:

- Total messages, input and output
- Message lengths, input and output
- Buffer data:
  - Buffer count
  - Buffers available and/or allocated
- Queue data:
  - Queue entries available
  - Queue entries used
  - Queue full counts
- Reset and retransmit counts:
  - Global (all stations)
  - By station
- Not ready-to-receive state counts:
  - Global (all stations)
  - By station

- Link-level statistics:
  - Message counts, input and output
  - Number of polls sent
  - Number of responses received
  - Number of line errors
  - Number of time-outs
  - Number of retransmits
  - Number of times offline
  - Number of message sequence errors
  - Number of resets
- Message throughput information
- Warm-start time

The statistics package functions are activated via SCI using the command STATS. The menu includes a number of two-character commands that can be entered to select the statistics function desired. When the statistics function is activated, it allows selection of the destination of the statistical data, that is, a file or the VDT. If a file is selected, it can be viewed using the Show File (SF) command. Four major types of operation are available via the STATS command:

- Network statistics accumulation
- Station statistics accumulation
- Line statistics accumulation
- Network testing using artificial data

The statistics menu is as follows:

```
STATISTICS AVAILABLE:

NETWORK:                               STATION:

     NETWORK STATUS          -NS         STATUS OF A STATION -SS
     NETWORK TRAFFIC         -NT         MESSAGE TOTALS      -SM
     STATUS OF BUFFERS       -NB         ERRORS REPORTED     -SE
     MESSAGE TOTALS          -NM         STATION RESET       -SR
     STATUS OF QUEUES        -NQ
     ERRORS REPORTED         -NE
     NETWORK RESET           -NR

LINE:                                  ARTIFICIAL DATA:

     LINE STATUS             -LL         ECHO MESSAGE        -AE
     STATION STATUS          -LS         REPLY TO MESSAGE    -AR
     LINE RESET              -LR         DROP MESSAGE        -AD

ENTER Q TO RETURN TO SCI

ENTER REQUEST:__ (Enter two-character command from above list)
```

The desired statistics function is selected by entering the appropriate two-character mnemonic after the ENTER REQUEST prompt. The following paragraphs describe each statistics command.

5.5.1 Network Statistics

5.5.1.1 Network Status (NS) Command.

The NS command can be used to show the following information:

* The number of lines enabled in the network
* The number of secondary stations attached to each line
* The status of each secondary station
* The line address and destination (DID) of each secondary station
* The current time
* The time the communications package was initialized or reset

To display the status of the network, enter NS in response to the ENTER REQUEST prompt. The statistics package responds by requesting a listing access name, as follows:

```
NETWORK STATUS
     LISTING ACCESS NAME: ST01
```

ST01 is an example response to this prompt. The statistics report is then written to the listing access name in the following format:

```
          ===  HDLC NETWORK STATUS ===

NUMBER OF LINES = nn
NUMBER OF SECONDARY STATIONS = nnn

                        LINE/STATION STATUS
                        -------------------
              # OF ATTACHED     LINE                  SECONDARY
DEVICE NAME    SECONDARIES      ADDRESS/DID            STATUS
-----------   -------------     -----------           ---------
    CMxx           15            >nn / nnnn              ON
                                     / nnnn
                                     / nnnn
                                     / nnnn
                                     / nnnn

                                 >nn / nnnn              OFF
                                     / nnnn
                                         ..
                                         ..
    CMyy           20            >nn / nnnn              ON
                                     / nnnnn
                                         ..
                                         ..

CLOCK TIMES: WARM START: MM/DD/HH:MM:SS   CURRENT: MM/DD/HH:MM:SS
```

The report may extend over more than one screen. The VDT scroll functions (F1 and F2) can be used to view the entire report.

5.5.1.2 Network Traffic (NT) Command.

The NT command can be used to display the following information:

- The total number of buffers available to the communications package
- The number of buffers in use
- The percentage of the total buffers still available for use
- The number of received messages queued at each software level
- The number of messages for transmission queued at each software level
- The number of messages received and transmitted as a function of message size
- The number of times each queue was full and an attempt was made to process an additional message (overflow)
- The current time
- The time the communications package was initialized or reset

To display traffic status, enter NT in response to the ENTER REQUEST prompt. The statistics package responds with the following display:

```
STATUS OF TRAFFIC, NETWORK
   LISTING ACCESS NAME:  ST01
```

ST01 is an example response to the prompt. The traffic report is then written to the VDT in the following format:

```
     ===  HDLC NETWORK TRAFFIC INFORMATION ===

BUFFERS:  TOTAL = ***     ALLOCATED = ***     % AVAILABLE = **.**

QUEUE LENS: OUTPUT          INPUT      OVERFLOW: OUTPUT        INPUT
            ------          -----                ------        -----
      L4 =            L2 =                  L4 =          L2 =
      L3 =            L3 =                  L3 =          L3 =
      L2 =           A/S =                  L2 =         A/S =

MESSAGES:                             OUTPUT            INPUT
           1 - 16 BYTES          =                 =
          17 - 32                =                 =
          33 - 64                =                 =
          65 - 128               =                 =
         129 - 256               =                 =
        >256                     =                 =
        -------------          ------------      ----------
         SUM                     =                 =

       TOTAL RESETS = *****    TOTAL RETRANSMISSIONS = ****

CLOCK TIMES: WARM START: MM/DD/HH:MM:SS  CURRENT: MM/DD/HH:MM:SS
```

5.5.1.3 Network Buffers (NB) Command.

The NB command can be used to display the following information:

- The number of buffers available
- The percentage of buffers available
- The number of buffers allocated
- The percentage of buffers allocated
- The maximum buffer size
- The current time
- The time the communications package was initialized or reset

To display the status of the network buffers, enter NB in response to the ENTER REQUEST prompt. The statistics package responds with the following display:

```
STATUS OF BUFFERS, NETWORK
     LISTING ACCESS NAME:  ST01
```

ST01 is an example response to the prompt. The buffer status is then written to the VDT in the following format:

```
        ===  HDLC NETWORK BUFFERS STATUS ===

            TOTAL BUFFERS = ****
        BUFFERS ALLOCATED = ****
        BUFFERS AVAILABLE = ****
      % BUFFERS ALLOCATED = **.**
      % BUFFERS AVAILABLE = **.**
      MAXIMUM BUFFER SIZE = *** BYTES

CLOCK TIMES: WARM START:MM/DD/HH:MM:SS   CURRENT:MM/DD/HH:MM:SS
```

5.5.1.4 Network Queues (NQ) Command.

The NQ command can be used to display the following information:

- The length of each output queue
- The length of each input queue
- The number of times each queue was full and an attempt was made to process another message
- The current time
- The time the communications package was initialized or reset

To display queue data, enter NQ in response to the ENTER REQUEST prompt. The statistics package responds as follows:

```
    STATUS OF QUEUES, NETWORK
         LISTING ACCESS NAME:  ST01
```

ST01 is an example response to the prompt. The queue status is then written to the VDT in the following format:

```
                 ===  HDLC NETWORK QUEUES STATUS ===

QUEUE LENS: OUTPUT        INPUT         OVERFLOW:    OUTPUT      INPUT
            ------        -----                      ------      -----

   L4 =            L2  =                      L4  =          L2  =
   L3 =            L3  =                      L3  =          L3  =
   L2 =           A/S  =                      L2  =         A/S  =

CLOCK TIMES: WARM START: MM/DD/HH:MM:SS   CURRENT: MM/DD/HH:MM:SS
```

5.5.1.5 Network Errors (NE) Command.

The NE command can be used to display the following information:

- The total number of reset commands that have been issued because of secondary station error conditions
- The total number of times that messages were queued for transmission more than once
- The current time
- The time the communications package was initialized or reset

To display error information, enter NE in response to the ENTER REQUEST prompt. The statistics package responds with the following display:

```
ERRORS REPORTED, NETWORK
    LISTING ACCESS NAME:  ST01
```

ST01 is an example response to the prompt. The error statistics are then written to the VDT in the following format:

```
              ===  HDLC NETWORK ERROR INFORMATION ===

  TOTAL RESETS = ****              TOTAL RETRANSMISSIONS = ****

 CLOCK TIMES: WARM START:MM/DD/HH:MM:SS   CURRENT:MM/DD/HH:MM:SS
```

5.5.1.6 Number of Network Messages (NM) Command.

The NM command can be used to display the following information:

- The number of output messages, grouped by size
- The number of input messages, grouped by size
- The total numbers of input and output messages
- The current time
- The time the communications package was initialized or reset

To display network message statistics, enter NM in response to the ENTER REQUEST prompt. The statistics package responds with the following display:

```
NUMBER OF MESSAGES, NETWORK
    LISTING ACCESS NAME:  ST01
```

ST01 is an example response to the prompt. The message statistics are then written to the VDT in the following format:

```
        ===  HDLC NETWORK MESSAGE INFORMATION ===

MESSAGES:                           OUTPUT            INPUT
            1  -   16 BYTES      =                =
            17 -   32 BYTES      =                =
            33 -   64 BYTES      =                =
            65 -  128 BYTES      =                =
            129-  256 BYTES      =                =
            >256      BYTES      =                =
            --------          ------------    ----------------
            SUM                  =                =

CLOCK TIMES: WARM START:MM/DD/HH:MM:SS   CURRENT:MM/DD/HH:MM:SS
```

5.5.1.7 Network Restart (NR) Command.

The NR command can be used to reset all network, station and line stastistical counters to zero. Warm-start time is reset to the current time and statistical counting is restarted. To activate the network restart function, enter NR in response to the ENTER REQUEST command.

5.5.2 Station Statistics

5.5.2.1 Status of a Station (SS) Command.

The SS command can be used to display the following station information:

- Current station status
- The total number of input messages, grouped by size
- The total number of output messages, grouped by size
- The total number of reset commands issued
- The total number of retransmissions attempted
- The total number of times the station or channel was not ready to receive
- The current time
- The time the communications package was initialized or reset

To display station status, enter SS in response to the ENTER REQUEST prompt. The statistics package responds with the following display:

```
STATUS OF A SECONDARY STATION
     STATION ID(0100-9999): 0100
     LISTING ACCESS NAME: .STATISTICS.STA0100
```

The responses to the prompts are examples of a station ID and a listing filename, respectively. Station status statistics are then written to the file in the following format:

```
              ***  HDLC STATION STATUS ***

                       STATION ID =  NNNN

   CURRENT STATE OF STATION (ON/OFF) =

   MESSAGES:                          OUTPUT          INPUT
            1  -  16 BYTES       =              =
            17 -  32 BYTES       =              =
            33 -  64 BYTES       =              =
            65 - 128 BYTES       =              =
            129- 256 BYTES       =              =
            >256     BYTES       =              =
            --------              ----------     ------------
             SUM                 =              =

        TOTAL RESETS = ****      TOTAL RETRANSMISSIONS = ****

    TIMES STATION, OR CHANNEL, WAS NOT READY TO RECEIVE = ****

  CLOCK TIMES: WARM START:MM/DD/HH:MM:SS   CURRENT:MM/DD/HH:MM:SS
```

5.5.2.2 Station Errors Reported (SE) Command.

The SE command can be used to display the following information:

- The total number of times the station was reset
- The total number of retransmission attempts
- The total number of times the station, or channel, was not ready to receive
- The current time
- The time the communications package was initialized or reset

To display the station errors report, enter SE in response to the ENTER COMMAND prompt. The statistics package responds with the following display:

```
    ERRORS REPORTED, STATION

         STATION ID(0100-9999): 0200
         LISTING ACCESS NAME: ST02
```

The example responses to the prompts show a station ID and the VDT designated to receive the error report. The error report is displayed in the following format:

```
            ***  HDLC STATION ERRORS ***

                     STATION ID =  NNNN

       TOTAL RESETS = ****       TOTAL RETRANSMISSIONS = ****

  TIMES STATION, OR CHANNEL, WAS NOT READY TO RECEIVE = ****

CLOCK TIMES: WARM START:MM/DD/HH:MM:SS   CURRENT:MM/DD/HH:MM:SS
```

5.5.2.3 Station Message Totals (SM) Command.

The SM command can be used to display the following information:

- Station ID
- The number of input messages, grouped by size
- The number of output messages, grouped by size
- The current time
- The time the communications package was initialized or reset

To display station message statistics, enter SM in response to the ENTER REQUEST prompt. The statistics package responds with the following display:

```
MESSAGE TOTALS, STATION
     STATION ID(0100-9999): 0340
     LISTING ACCESS NAME: ST11
```

The example responses to the prompts show a station ID and the VDT designated to receive the report. The statistics are then displayed in the following format:

```
   ***  HDLC STATION MESSAGE INFORMATION ***

                 STATION ID =  NNNN

MESSAGES:                        OUTPUT           INPUT
       1  -  16 BYTES      =              =
       17 -  32 BYTES      =              =
       33 -  64 BYTES      =              =
       65 - 128 BYTES      =              =
       129- 256 BYTES      =              =
       >256     BYTES      =              =
       --------             ------------   ----------------
        SUM                =              =


CLOCK TIMES: WARM START:MM/DD/HH:MM:SS   CURRENT:MM/DD/HH:MM:SS
```

5.5.2.4 Station Reset (SR) Command.

The SR command can be used to reset all statistics counters associated with a specified station. Counting resumes from zero. To reset a station in this way, enter SR in response to the ENTER REQUEST prompt. The statistics package responds with the following display:

```
RESET STATION
          STATION ID (0100-9999): 0100
```

The response to the prompt is an example of a station ID. The station counters are then reset as indicated above.

5.5.3 Line Statistics

5.5.3.1 Line Status (LL) Command.

The LL command can be used to display the following information:

* The number of received messages
* The number of transmitted messages
* The number of polls
* The number of responses to those polls
* The number of line errors by category
* The number of stations attached to the line
* The current time
* The time the communications package was initialized or reset

To display line status, enter LL in response to the ENTER REQUEST prompt. The statistics package responds with the following display:

```
STATUS OF A LINE
  FCCC DEVICE NAME (CMXX): CM01
      LISTING ACCESS NAME: ST01
```

The example responses to the prompts show a communications device name and the VDT designated to receive the report. The line status is displayed in the following format:

```
           === HDLC LINE INFORMATION ===

                   FCCC DEVICE NAME = CM01

  MESSAGES: OUT = ****              IN = ****
     POLLS:     = ****       RESPONSES = ****

  LINE ERRORS:
  ------------
                TIME OUTS          = *****
                RETRANSMISSIONS    = *****
                RESETS             = *****
       MESSAGE SEQUENCE ERRORS     = *****
                OFF LINE COUNT     = *****
                INPUT ERRORS       = *****
                OUTPUT ERRORS      = *****

  NUMBER OF MULTIPOINT STATIONS ATTACHED = **

CLOCK TIMES: WARM START:MM/DD/HH:MM:SS   CURRENT:MM/DD/HH:MM:SS
```

5.5.3.2 Status of a Station at Line Level (LS) Command.

The LS command can be used to display the following station information:

- The total number of messages received from that station
- The total number of messages transmitted to that station
- The total number of polls transmitted to that station
- The total number of responses to those polls received from that station
- The total number of errors detected by category
- The time delay between polls specified for that station
- The current time
- The time the communications package was initialized or reset

To diplay station status at line level, enter LS in response to the ENTER REQUEST prompt. The statistics package responds with the following display:

```
    STATUS OF A STATION, LINE LEVEL INFORMATION
           STATION ID (0100-9999): 0100
              LISTING ACCESS NAME: ST01
```

The example responses to the prompts show a station ID and the VDT designated to receive the report. The report is displayed in the follow format:

```
           === HDLC LINE INFORMATION ===

FCCC DEVICE NAME = CM01      STATION ID = NNNN     DROP ADDRESS = NN

  MESSAGES: OUT = ****                 IN = ****
     POLLS:     = ****          RESPONSES = ****

  LINE ERRORS:
  ------------
                 TIME OUTS          = *****
                 RETRANSMISSIONS    = *****
                 RESETS             = *****
         MESSAGE SEQUENCE ERRORS    = *****
                 OFF LINE COUNT     = *****
                 INPUT ERRORS       = *****
                 OUTPUT ERRORS      = *****

  TIME DELAY BETWEEN POLLS:       = ***.***  SECONDS

CLOCK TIMES: WARM START:MM/DD/HH:MM:SS   CURRENT:MM/DD/HH:MM:SS
```

5.5.3.3 Line/Station Reset (LR) Command.

The LR command can be used to reset the statistical counters pertaining to a station or to all stations on a line. To use this function, enter LR in response to the ENTER REQUEST prompt. The statistics package responds with the following display:

```
  RESET LINE
        FCCC DEVICE NAME (CMXX): 0
         STATION ID (0100-9999): 0100
```

The responses to the prompts are examples of a secondary station ID and the corresponding FCCC channel. To reset the counters for a single station, enter its ID in response to the ENTER STATION ID prompt. To reset the counters for all stations on the line specified in response to the first prompt, default through the second prompt. Resetting any or all of the line counts has no effect at all on the network or station statistics.

5.5.4 Artificial Data

The HDLC package includes the ability to specify and generate messages for testing the network in conjunction with the statistics-gathering functions. This facility is termed artifical data and is available only for use with TX5 stations.

To use the artificial data task, implement the following:

1. Include remote SCI in the secondary station during system generation and network generation
2. Include the install task (IT) function as part of the remote SCI
3. Include the kill task (KT) function as part of the remote SCI
4. Assign a network ID to task >10 in the TX5 during network generation
5. Install the artificial data program in the TX5 using remote SCI IT and KT functions
6. Execute task >10 after it is installed

To load the program into the TX5 secondary station, use the following procedure:

1. Enter REMSCI at a DX10 primary terminal.
2. Enter IT after the <> prompt. This results in the following display:

```
<> IT

   REMOTE INSTALL TASK

      SECONDARY ID: 0200
   OBJECT FILENAME: .INDSCOMM.TXOBJ.ARTDAT      (example
           TASK ID: 010                          responses)
          PRIORITY: 3
        PRIVELEGED: N
```

Enter responses to the prompts as follows:

a. SECONDARY ID. The network ID assigned to the secondary station during network generation.

b. OBJECT FILENAME. The file containing the artificial data program. For artificial data this file is .INDSCOMM.OBJECT.ARTDAT.

c. TASK ID. This must be >10. A network ID also must have been assigned during network generation.

d. PRIORITY. This value must be in the range of 1 through 3. A higher value increases throughput.

e. PRIVILEGED. Answer NO, this task is not privileged.

3. After the task is installed, it must be executed using the remote SCI command and XT as follows:

```
<> XT

   REMOTE EXECUTE TASK
     SECONDARY ID: 0200
          TASK ID: >10        (example responses)
           PARM 1: 0
           PARM 2: 0
```

Enter responses to the above prompts as follows:

a. SECONDARY ID. Enter the network ID of the secondary station.

b. TASK ID. Enter the ID of the task just installed.

c. PARM 1 and PARM 2. Take the defaults for these prompts.

When the task is installed and executed in the secondary station, the artificial data program is ready to process data.

To send a message to the secondary station enter AE, AR, or AD in response to the ENTER REQUEST prompt (on the statistics menu). The following display appears in response to any of the three commands:

```
   STATION ID (0100-9999): 0220
      TASK ID (0100-9999): 0225   (example
  MESSAGE COUNT (20 - 50): 50      responses)
 MESSAGE LENGTH (1 - 256): 100
            TASK PRIORITY: 1
 USE ACTIVATION SERVICES?: NO
      LISTING ACCESS NAME: LP02
```

Enter responses to the above prompts as follows:

- STATION ID. Enter the network ID of the TX5 station.

* TASK ID. Enter the network ID of the TX5 task >10.

* MESSAGE COUNT. Enter the number of messages to be sent. This must be in the range of 20 through 50.

* MESSAGE LENGTH. Enter the length of the message in bytes (excluding overhead).

* TASK PRIORITY. Enter 1 through 3, depending on the priority required. 3 is the DX10 default.

* USE ACTIVATION SERVICES. Enter YES or NO to indicate whether activation services are required at the secondary station.

* LISTING ACCESS NAME. Enter the device or file that is to receive the results of the artificial data test.

If the command AE (Echo Message) was entered, the specified number of messages, each of the specified length, is sent to the secondary station and echoed back to the primary station. If the command AR (Reply to Message) was entered, the specified number of messages, each of the specified length, is sent to the secondary station. The secondary station then sends back a one-byte acknowledgement. If the command AD (Drop Message) was entered, th specified number of messages, each of the specified length, is sent. No reply is sent back to the primary station.

Upon completion of the test a report in the following format is sent to the device or file specified as the listing access name:

```
     === HDLC MESSAGE TRANSFER INFORMATION ===

 SECONDARY STATION ID = NNNN      TEST DURATION = **** SECONDS

 SEND COUNT/LENGTH    = ***/***     REPLY COUNT/LENGTH = ***/***

 MESSAGE OUTPUT RATE  = **.** MESSAGES/SECOND

MESSAGE RESPONSE TIMES, IN SECONDS, AS MEASURED AT THE SENDING TASK:

  MEAN    = **.**             MODAL   = **.**

  MAXIMUM = **.**             MINIMUM = **.**

  LINE SERVICE TIME/MESSAGE          = **.**  SECONDS

 LINE UTILIZATION TIME               = ***.** SECONDS

  WAIT TIME/MESSAGE                  = **.**  SECONDS

CLOCK TIMES - START: MM/DD/HH:MM:SS    FINISH: MM/DD/HH:MM:SS
```

The following parameters appear in the report:

- * SECONDARY STATION ID. This is the station ID entered at the start of the test.
- * TEST DURATION. This is the number of seconds that elapsed from the start to the end of the transmission.
- * SEND COUNT/LENGTH. This is the count of the number of messages sent and the length of those messages. For correct station operation, these numbers should be the same as those entered in response to the MESSAGE COUNT and MESSAGE LENGTH prompts when the artificial data test was activated.
- * REPLY COUNT/LENGTH. This is the number of replies received from the secondary station and the length of those replies. For correct operation, these should be as follows:
  - AE command. The REPLY COUNT/LENGTH is the same as the SEND COUNT/LENGTH.
  - AR command. The reply length is one. The reply count is the same as the send count.
  - AD command. Both the reply length and the reply count are zero.
- * MESSAGE OUTPUT RATE. This is the average number of mesages sent each second by the primary station.
- * MEAN. This is the average response time for all messages.
- * MODAL. This is the most frequently occurring response time during the test. If any single response exceeds 20 seconds, the test terminates and an error message is displayed.
- * MAXIMUM and MINIMUM. These are the longest and the shortest response times observed during the test, taking into account that the system clock has an accuracy of only ±1 second.
- * LINE SERVICE TIME/MESSAGE. This is the line time required to transmit a single message, including the overhead of 15 bytes.
- * LINE UTILIZATION TIME. This is the total amount of time required to transmit the specified number of messages.
- * WAIT TIME/MESSAGE. This is a measure of the average time a message spends in the system queues awaiting

transmission. In the DX10, it is the time spent in moving data from the user task buffer to the communications buffer, servicing the message at each software level, and queuing it down to the line level. It is dependent on the existing DX10 workload.

- * CLOCK TIMES. These are the times that the test completed (FINISH) and the time the communication package was initialized or reset (WARM-START).

5.6 ACTIVATE/DEACTIVATE UTILITIES

Activate and deactivate utilities are available to take a secondary station offline or bring one online. To activate a station, enter the SCI command ACT at a primary station terminal and identify the station to be activated by its network ID, as shown in the following example:

```
[ ] ACT
   ACTIVATE A STATION
          STATION NUMBER: <network ID>
```

After the station has been identified, it is polled during the next polling sequence. If the station is operational, it responds to the polls sent to it by the primary station. If it is not operational, it cannot respond to the polls sent. In this case, the primary station polling time-out expires and an error message is written to the system log indicating that the station is down. Activating a station is a primary station process only. If the specified secondary station is down after the ACT command is entered, it may indicate that the station requires downloading.

To deactivate a station, enter the SCI command DEACT at a primary station terminal and identify the station to be deactivated by its network ID, as shown in the following example:

```
[ ] DEACT
   DEACTIVATE A STATION
     STATION NUMBER: <network ID>
```

The deactivated station is now omitted from the primary station polling sequence. It is placed in the disconnect mode, and polling cannot resume until the station is reactivated by the ACT command. A download operation cannot be started to a station in the deactivated state. To start a download operation, execute the ACT command first and then the DOWNLOAD command (see paragraph 5.3).

Appendix A

Error Messages

A.1 GENERAL

This appendix describes the error messages written to the system log during activation and operation of the network. Error messages may be encountered at each software level in the communications package. Error messages may also be encountered when the communications warm start program is executed. Each of the error messages are described separately in the following paragraphs.

Error messages that occur when executing the applications utilities are described in Section 5. Status codes and error messages returned to applications programs are described in Section 4.

All error messages written to the system log begin with the common COMM ERROR(CC) format, where (CC) is a two-character level or function identifier. The two-character code that is printed with each message identifies the following:

Error Code	Meaning
WT	Error encountered in communications warm start program
UL	Error encountered in upper-level software
PK	Error encountered in packet-level software
LC	Error encountered in data link software
AS	Error encountered in activation services

A.1.1 Warm Start Program Errors

The warm start program reports error conditions using the following format:

```
COMM ERROR(WT) = NNNN CE.GG.CC.AD.LU

               or

COMM ERROR(WT) = NNNN 2B.BE.00.00.00
```

where:

NNNN	is the packet address of the packet affected by the error (for use by analysts only).
CE	is the error code returned on the supervisor call (SVC) made (to the FCCC) to send the logical channel indicator (LCI) assignment message.
BE	is the error code returned if the warm start task is unable to bid the download task. The value 2B indicates that this error is a bid task SVC error.
GG	is the virtual channel group number.
CC	is the logical channel number. GG and CC together are the LCI.
AD	is the physical (or switch) address of the affected secondary station.
LU	is the LUNO assigned to the secondary station´s line. This value defines the actual line (line 1, line 2, etc.) to which the secondary station is attached.

A.1.2 Upper-Level Errors

The upper-level software reports errors in the following format. Upper-level errors are those errors not included in the other error reports.

```
COMM ERROR(UL) = EE.GG.CC.DS.SR
```

where:

EE is the error code. These error codes are also returned to calling tasks when an error occurs. The error codes are as follows:

Error Code	Meaning
00	Request complete
01	Invalid destination ID (DID)
02	Queue congestion (temporary condition)
03	Session already exists for this caller
04	No session blocks available
05	No buffers available
06	Invalid user data message length
07	Unable to accept write or read calls
08	Invalid I/O opcode in caller´s physical record block (PRB)

GG is the virtual channel group number.

CC is the logical channel number. GG and CC indicate the permanent virtual channel (PVC) affected.

DS is the destination ID (DID) number.

SR is the request code, where this value has the following meanings:

Error Code	Meaning
00	Data message
07	Fast message

A.1.3 Packet-Level Errors

The packet-level software reports error conditions in the following format.

```
COMM ERROR(PK) = NNNN 0GCC.SREE.SSID.DSID
```

where:

NNNN is the data packet address of the packet affected by the error (for use by analysts only).

0G is the virtual channel group.

CC is the logical channel number. Together 0G and CC are referred to as the logical channel indicator (LCI) and identify the PVC affected by the error.

SR is the packet P(S) and P(R) counts. These values are each right justified in the left and right nibble of this byte. The maximum value of each should not exceed 7.

EE is the error code. It defines the type of error condition encountered at run time:

Error Code	Meaning
01	Invalid LCI
02	Invalid packet type
03	Invalid packet length (too small)
04	Invalid state in send receive ready
05	Invalid state in send receive not ready
06	Invalid state on packet time-out
07	Invalid state on acknowledgement time-out
08	Reset received
09	Reset sent
0A	Packet retransmit failed

SSID is the source (sender) ID (SID).

DSID is the destination ID (DID).

A.1.4 Data Link Errors

The data link level software reports error conditions in the following format. If you need a more complete description of one of the following codes, refer to the Model 990 Computer Four Channel Communications Controller Installation and Operation Manual. If an error code appears that is not listed, below refer to Model 990 Computer DX10 Operating System Reference Manual Volume 6 - Error Reporting and Recovery or to the FCCC manual referred to above.

```
COMM ERROR(LC) = NNNN XX.YY.LL.ID.CC
```

where:

NNNN is the packet address of the destination packet.

XX is the SVC performed to the FCCC (that is, read, write, specials, etc).

YY is the error code returned by the SVC call. The following error codes are defined:

Error Code	Meaning
02	Illegal opcode
03	Channel not opened
05	Illegal memory address
06	Operation aborted
07	Device I/O error
0D	Insuufficient system table area
13	Time-out error
18	Disconnected error
19	Unable to accept command
21	Invalid channel number
25	No CRB buffers available
26	Download/dump illegal address
28	Frame complete not found
29	Illegal buffer address
2A	Chain size exceeded threshold
2B	Chain write construction error
2C	No slave buffer count
2D	Odd reserve memory length
2E	Reserve memory blocks exceeded

Error Code	Meaning
30	Time-out error
31	Memory parity error
32	TILINE channel 0 data compare error
33	Illegal TILINE opcode
34	Zero byte count specified
36	TILINE channel 1 compare error
38	Odd host header address
39	TILINE transfer did not complete
3F	No CRB owner
41	Receive buffer overrun
43	Write parameters ISR error
44	Open with undefined protocol
45	Channel already open
46	Transmit abort detected
4D	Character parity error
4E	Framing error
4F	Receiver overrun error
50	Invalid response address (no SCB)
51	Download in progress error
52	Station offline error (timed out)
53	Station commanded offline
55	Station restored to online status
60	Illegal buffer length requested
61	Buffer requested larger than pool
62	Buffer size requested not found
64	Return buffer overlaps
65	Returning same buffer twice
66	Return buffer above buffer pool
67	Return buffer below buffer pool
68	Allocated flag not set

LL is the LUNO assigned to the call (the channel number can be determined from the LUNO).

ID is the HDLC line (physical) address.

CC is the I/O call code.

A.1.5 Fatal Data Link Errors

The following errors cause an interruption of processing:

Error Code	Meaning
61	FCCC power failure
62	FCCC memory parity error
63	Illegal level 3 interrupt
64	Illegal level 4 interrupt
65	Illegal level 5 interrupt
66	Illegal level 6 interrupt
67	Illegal level 7 interrupt
68	Illegal level 8 interrupt
69	Illegal level 9 interrupt
6A	Illegal level A interrupt
6B	Illegal level B interrupt
6C	Illegal level C interrupt
6D	Illegal level D interrupt
6E	Illegal level E interrupt
6F	Illegal level F interrupt
70	Illegal XOP encountered
71	FCCC TMS 9900 self-test error
72	FCCC ROM CRC error during ROM test
73	FCCC RAM data test error
74	FCCC TMS 9901 clock test failure
75	FCCC channel test failure
77	Memory parity error not set by reset
78	No interrupt from memory parity error
79	Cannot clear memory parity error
7A	TILINE time-out error not set by reset
7B	Cannot clear TILINE error
7C	RAM refresh error

A.1.6 Activation Services Errors

The activation services reports errors in the following format:

```
COMM ERROR(AS) = NNNN XX.YY.BB.RR
```

where

NNNN is the packet address of the undeliverable packet.

XX is the SVC performed by activation services:

>0E	Activate task from time delay
>07	Unsuspend task
>2B	Execute (bid) task

YY is the state code returned by the SVC call. It defines the state of the task when the activation attempt was made.

BB is the bid task ID of the task activation services attempted to activate.

RR is the run task ID of the task activation services attempted to activate.

Appendix B

Throughput

B.1 POLLING METHODS

The two processes involved in the control of a multipoint polled line are polling and addressing. Polling is used by the master, or primary, station to solicit input messages from the secondary stations. Addressing is used to send messages to secondary stations. In the polling process, the primary station invites the secondary stations one by one to send any messages that are waiting to be transmitted to the primary station. A secondary with a message sends it as a reply to the poll. If a secondary has no message to send, the reply is a negative poll indicating that the secondary has no data to send. The primary station then polls the next station. This process is referred to as roll call polling.

The primary invites each secondary to send by working through a polling list until all secondary stations in the list have been polled. The primary then goes back to the top of the polling list and begins the roll call again. This polling list is a fixed-format data structure that is generally stored on a disk file and loaded into memory prior to the start of communications activity. It should be modifiable so that the list of stations can be changed as the network grows (or decreases) in size.

There are many variations of roll call polling, but none of them eliminate the roll calling aspect of the process. In general, any scheme that causes each secondary station to be polled as the primary works its way through a polling list can be termed roll call polling. Some variations of the roll call process are as follows:

* Prioritized roll call - Each secondary station is assigned a priority value so that higher priority stations are polled more frequently than lower priority stations.

* Time-delayed roll call - Each secondary station is assigned a time value that defines the time to wait between polls. The poller still works its way through a polling list but issues polls only to stations whose time value has expired.

When the primary station has a message to send to the secondary, the message can be sent in place of the poll. Alternatively, the primary can send a poll to the secondary indicating that a message is to be sent and asking if the secondary can receive the message. Many protocol variations exist, and within each protocol many possible implementations can be used to handle this case.

Another method of polling is hub polling. This method is generally used on data links that are connected by modems over long distances. The poll cycle begins when the primary polls the station that is physically the greatest distance away. That station then polls its nearest neighbor, and so on, until finally the last station issues a return poll to the primary. This method of polling requires more complex software at each secondary than the roll call method. It also requires that each secondary know the address of its neighbor (the next station to be polled) and have recovery ability in case that station does not respond to the poll. This method is intended, in part, to overcome the propagation delay incurred when transmitting over great distances.

B.2 HDLC PACKAGE POLLING METHOD

The DX10 HDLC Communications Package provides a method of polling that is designed to improve throughput to the secondary stations. The method is best described as time delayed roll call with data priority. With this method, polls to each secondary station occur after measured delays unless a message is queued for that station during the delay period. In that case, the delay is preempted and the message is transmitted. The time delay between polls is selected by the user and can range from a minimum of 0 milliseconds to a maximum of 8000 seconds (133.33 minutes) in steps of 250 milliseconds. A zero-second delay implies that the station should be polled at every poll opportunity. If all secondary stations on one multipoint line have a delay of zero, the polling reverts to pure roll call (that is, each station is polled every time the poller accesses that station´s entry in the polling list).

If the poller finds no station ready for polling during the entire scan, it starts the next scan by polling the first station in the list. It then continues through the list, omitting any station whose time delay has not expired. On the next scan, it starts by polling the second station in the list; on the third scan, it starts with the third station; and so on. Using this method, all secondary stations in the list are polled in an orderly manner during periods of no scheduled polling activity.

After a message is transmitted to a station, the poller uses the following process to ensure that a sufficient number of attempts is made to collect any response data that the secondary station needs to transmit. The poller sends a series of ready-to-receive polls in the following sequence:

1. Ten polls with no delay
2. Ten polls with a delay of at least 250 milliseconds between each poll
3. Ten polls with a delay of at least 500 milliseconds between each poll
4. Ten polls with a delay of at least 750 milliseconds between each poll

After the last poll in this cycle is sent, the poller reverts to the normal delay between polls for this station. This polling activity gives the secondary station approximately 15.5 seconds to formulate a response message and send it back to the primary station.

The usefulness of this method can be seen in the operation of the PM550 Programmable Controller. After it is loaded with a functioning program, it can operate for long periods of time without any direction from a primary station program. The time between messages to this device should be known when the network is defined; the delay between polls to this device should be specified to limit the number of polls sent to it. Ideally, the PM550 should never be polled if the time between message transmissions to it is fairly short, that is, less than one or two minutes. The device would normally be polled for status at one- or two-minute intervals; but if a message is queued for it during this interval, the delay between polls is preempted and the message is transmitted immediately. After the message is transmitted, the primary station issues the polls at the faster rate (described above) and eventually resumes polling the device at the selected time interval. Under operational circumstances such as these, it would be wasteful to poll the secondary every 250 milliseconds or so. In fact, it may be wasteful to ever poll the secondary if some primary station program will be sending it messages fairly frequently.

In contrast to the preceding example, a TX5 secondary can be configured with software that operates independently of any primary station software. For these secondary stations a maximum acceptable time delay between polls must be selected to ensure that the secondary station's data is collected as required. A time value from 500 milliseconds up to 5 or 10 seconds may be acceptable for such secondary stations, depending on the application involved.

A secondary station can always respond to a poll with a message unless the poll is negative. A negative poll is sent only as a status poll and only if the primary station cannot accept data in response. This poll type is generally sent when the primary station has no input buffers available to accept response data. Unless this situation continues for a prolonged time, the occurrence of negative polls has no effect on the application programs attempting to send messages to the secondary stations.

B.3 TIME DELAY VALUE SELECTION

The user selects the time delay between polls during network generation. The time delay selection must be based on the user´s knowledge of a secondary station´s function. The secondary´s function should be known in sufficient detail to allow the network planner to select the appropriate time delay. During network generation, valid responses to the MAXIMUM TIME BETWEEN POLLS prompt are in the range of 0 to 32,000. An entry of 0 indicates that the secondary should be polled at every poll opportunity. An entry greater than 0 indicates the desired delay time between polls. It is prudent to choose this delay based on the operational requirements of the secondary station. Each unit of time entered is equal to 250 milliseconds.

B.4 POLL AND RESPONSE COMPONENTS

Each HDLC poll and and each response consists of 48 bits. For a line data rate of 9600 bits per second (bps), the line time to poll is approximately 48/9600 or 5 milliseconds, and the line time to respond is the same. At 7200 bps, the line time to poll is 48/7200 or 6.67 milliseconds, and the line time to respond is the same. As the line data rate decreases, the line time to send a poll and get a response increases accordingly. For example, at 300 bps, the line time to poll is 160 milliseconds and the line time to respond is the same. The line data rate should be taken into consideration by applications that send messages to secondary stations and expect the addressed secondary (or task in that secondary) to send a reply message back. Obviously, the response time is greatly affected by the line data rate.

B.4.1 Turnaround Time (TAT)

The turnaround time (TAT) at a secondary station differs from one secondary to the next. TAT is the time component of a secondary system that is used to formulate, queue, and begin the transmission of the response to a poll. This TAT can be considered at both the data link level and the levels above the data link. For example, the data link level TAT is relatively small in comparison to the line transmission time of a poll with response. But at the upper levels, the TAT is much larger. For example, to send a 260-byte message to a PM550 and get a 1-byte response message back requires a TAT of about 1.92 seconds at the PM550 software levels above the data link level. This time must be considered, along with the line data rate, by an applications task at the primary that communicates with a PM550 secondary station.

Another component of TAT is the time between polls. After a message is sent to a secondary station, it must be delivered to its final destination. This delivery requires the passing of the message from queue to queue. At each queue "stop" in this movement, some processing is performed on the header information that is a part of every message. After the message arrives at its destination, a response may be formed and written to the communications system. This response is now processed back through the output queues, header information is attached as required, and, finally, the message is queued awaiting transmission. However, transmission cannot occur until the secondary receives a poll from the primary. At that time the message can be sent with the response to the poll. If another message had arrived in the queue earlier, it would have been sent first (that is, another poll must come in before the next message in the output queue can be sent).

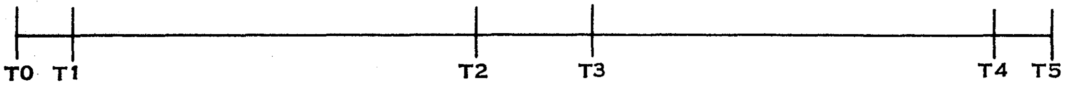
Determining the response times to traffic generated at all levels is a complex task and, generally, of little or no use to the user. The user is primarily concerned with how long it takes to send a message (or series of messages) and how long it usually takes to get a reply to a message or a series of messages sent to a secondary station. This information can be determined by using the statistical functions referred to as artificial data. Artificial data can be generated to a TX5 secondary station; the results provide the user with information on response times that can be expected in the actual operating environment. This information is the result of simulation performed by the artificial data task.

B.4.2 Polling Time for a Single Line

Assuming that all secondary stations are polled in roll call fashion, the total time to send polls and collect responses at 9600 bps on one line that has 32 secondary stations attached is 32 times 10 milliseconds, or about 320 milliseconds. Note also that with the option of selecting a time delay between polls, a delay value of 250 milliseconds between polls is not a realistic selection if that delay value is selected for every secondary on a line with more than about 25 secondary stations. With 250-millisecond delays, the time delay will expire before the 24th or 25th station is polled, or sooner if data is sent as part of a poll or a response. Once this time delay has expired for most of the secondary stations, the delay time is of no value and the polling reverts to pure roll call.

For any line data rate below 9600 bps, the polling reverts to roll call for a smaller number of secondary station. Assuming a line data rate of 300 bps, the time to send one poll with response is 48/300 + 48/300 = 320 milliseconds. If a 250-millisecond delay is selected for every secondary stations, it will expire for all secondary stations before the response to the first poll is completed.

The sum of the polling operations at the data link level is shown below. The TAT (T3 - T2) is the TAT for the data link level software and any line turnaround time required; it does not include TAT for upper-level or task-level replies.



2280616

where:

T0	is the start of the FCCC poller program scan through the polling list.
T1	is the start of the poll transmission over the line.
T2	is the end of the poll transmission.
T3	is start of the response transmission.
T4	is the end of the response transmission.
T5	is when the FCCC poller program processes the response and goes back to the function at T0 to poll the next station in the list.

T1-T0 is the FCCC table scan and processing time.

T2-T1 is the poll transmission time.

T3-T2 is the time required for the secondary to process the input and formulate a data link level response (TAT).

T4-T3 is the response transmission time.

T5-T4 is the FCCC poller program response processing time.

During time T1-T0, the FCCC performs such functions as the following:

* Chains to the next station control block and tests to see if there is time to poll. If there is not enough time, it chains to the next station control block.
* Acquires an input communications buffer and associates it with the output poll. If no buffer is available, it sets appropriate flags and prepares to send a negative poll.
* Gets HDLC sequence counts and builds the poll command.
* Sets appropriate flags and counters and performs an output call.

During time T3-T2, the secondary station performs such functions as the following:

* Processes input characters.
* Tests the input poll for validity and directive. Validity checking tests the poll type and sequence counts for validity. Directive checking tests the poll type to determine if the secondary response can include a message.
* Processes and begins transfer of response.

During time T5-T4, the poller processes the response to the poll by performing the following:

* Tests the input (response) for presence of message.
* Tests HDLC sequence counts regardless of response type.
* Performs any flag and counting operations required to finalize the poll sequence.
* Returns to the logic to begin the chaining operation through the station control blocks.

B.5 FCCC MULTILINE OPERATION

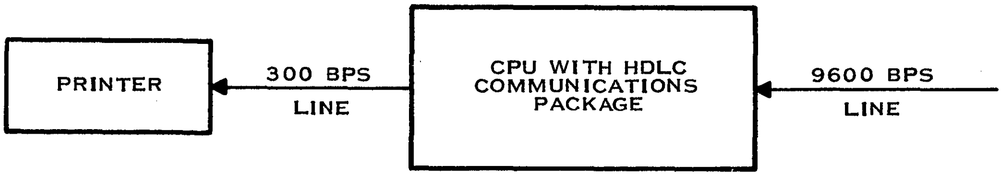
The communications package software supports operation of all four channels, or lines, of the FCCC card included in the DX10 system. Including the FCCC card in DX10 is a system generation function and including the HDLC operation of an FCCC line is a network generation function. Any FCCC line to be operated by the HDLC package is specified at network generation time. A set of software programs offloaded in FCCC RAM at COMMGO time performs the HDLC data link control, polling, and timing functions required to control traffic flow at the data link level. After the user specifies the number of FCCC lines and the characteristics of these lines during network generation, the tables are created to operate the FCCC lines as specified. This operation is transparent to the user tasks.

B.5.1 Data Rate Selection for Multiline Operation

The FCCC card has a total character-handling capability of 3800 characters per second (cps), one character equals one eight-bit byte. This limits FCCC throughput to an average of less than 9600 bits per second (bps) per channel. In the synchronous mode, a 9600-bps (9.6K-bps) line can transfer 1200 cps. Therefore, the FCCC cannot operate at 9.6K bps on all four channels simultaneously. A lesser aggregate bandwidth must be selected for the FCCC to operate correctly. Refer to Section 3 for data rate combinations that can be selected to make use of the FCCC maximum bandwidth. This chart should be consulted before beginning the network generation process and when creating documentation on the network.

B.6 AVOIDING BOTTLENECKS

Depicted below is a communications system node with a 9600-bps communications line and a 300-bps print device. Messages arriving at this node are directed to the print device. A rapid succession of input messages arriving on the communications line at 1200 cps is printed out at 37.5 cps. Under any practical conditions, the communications buffers available to that secondary station soon fill up and the communications system stops accepting input messages from that station (that is, the node rapidly becomes congested, at least for that station or line, because the communications transfer rate is significantly higher than the printer transfer rate).



2280617

One solution to this problem is to put some constraint on the program in the station sending the data. Another possible solution is to arrange the receiving software at the station to read input messages as rapidly as possible, spool the messages to a disk file, and interleave write operations to the printer with read (receive) operations. However, even with this arrangement, the total number of characters received must not be greater than the capacity of the disk file to adjust for situations where interleaving may not be possible (for example, if the printer is offline).

The DX10 primary station communications software limits the number of messages that can be sent to or received from any secondary station. This upper limit, called the window size, is set to a value based on the functionality of the communications software at the secondary station. In the DX10 HDLC Communications Package, it is based on the buffer capacity and software capability at each secondary. For TX5 secondary stations, the window size is seven and for TM990 or PM550 type secondary stations, it is one.

Allocation of a window size to each secondary station is not the ultimate solution, but it does provide some control over the flow of data. However, in a network where the the number of secondary stations times the window size is greater than the number of available buffers, deadlock could still occur. If every task at every node attempted to send as many messages as the system allowed, there generally would not be a sufficient number of buffers to accommodate all messages. Therefore, users should ensure that each task that communicates does so on an optimal basis. This approach helps to improve the system throughput and reduce the possibility of deadlock caused by traffic activity in excess of the handling ability of the communications package.

B.7 TRANSMISSION FORMAT

TRANSMISSIONS UNDER HDLC ARE IN THE FOLLOWING FORMAT:

BYTE	0	1	2	. . . . . .	n−2	n−1	n
	FLAG	ADDRESS	COMMAND	OPTIONAL DATA	CRC	CRC	FLAG

2280618

Each field is described in the following paragraphs.

B.7.1 Flag Fields

The flag byte is used at the start and end of the transmission block as a delimiter. Its value is binary 01111110 (>7E). Zero-bit insertion begins with the first byte after the start flag and ends at the second cyclic redundancy check (CRC) byte.

B.7.2 Address Field

The address field is the HDLC or drop address. Its value may lie in the range of 0 through 255; however, in the DX10 HDLC Communication Package, the upper limit is 63 if the secondary stations are any combination of TX5s, PM550s, or TM990s.

B.7.3 Command Field

The command field is the HDLC command/response byte. All valid values for this byte are given in the following list:

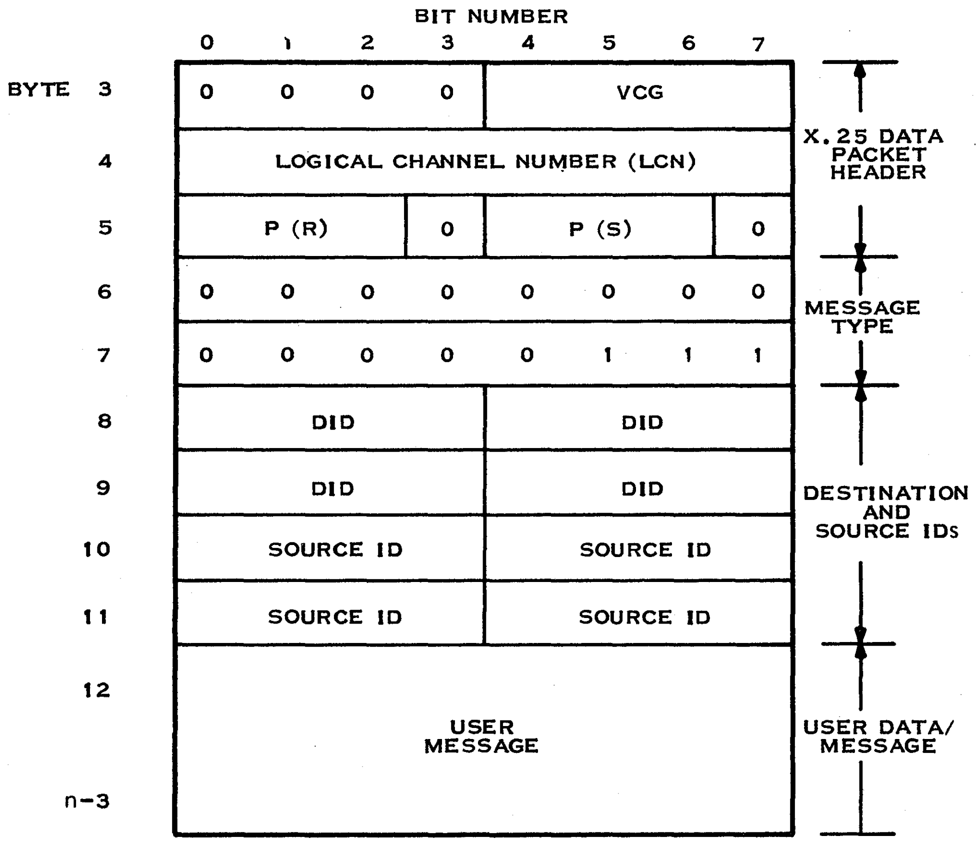
	Commands Sent by Primary	Responses Accepted by Primary	Meaning
Information (I) Frames:			
	I	I	Information transfer
Numbered Supervisory (S) Frames:			
	RR	RR	Receive ready
	RNR	RNR	Receive not ready (busy)
	REJ	REJ	Reject
	SIM		Set initialization mode (download)
		RIM	Request initialization mode (download)
Unnumbered (U) Frames:			
	UI		Unnumbered information transfer
		UA	Unnumbered acknowledge
	SNRM		Set normal response mode
	DISC		Disconnect command
		DM	I am disconnected
		CMDR	Command (frame) reject

B.7.4 Optional Data Field

The optional data field can be one of the following:

- A control message, a three byte field. This is the same as the first three bytes of the data packet shown below.
- User data with control information. Details are shown in the following diagram:

Optional Data Packet Format



	BIT NUMBER							
	0	1	2	3	4	5	6	7
BYTE 3	0	0	0	0	VCG			
4	LOGICAL CHANNEL NUMBER (LCN)							
5	P (R)			0	P (S)			0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	1	1
8	DID				DID			
9	DID				DID			
10	SOURCE ID				SOURCE ID			
11	SOURCE ID				SOURCE ID			
12	USER MESSAGE							
n-3								

2280619

The fields shown are as follows:

* Packet header, consisting of:
  - VCG, the FCCC line number to which the device is attached. This is a default value accepted during network generation.
  - LCN, the HDLC drop address. This is also a default value accepted during network generation.
  - P(R), the receive packet level counter
  - P(S), the transmit (send) packet level counter
* Message type, always set to 0007 for a data packet
* Destination and Source IDs coded in binary coded decimal and assigned during network generation
* User data/message. The maximum size is usually 256 or 260 bytes.

B.7.5 CRC Fields

The CRC fields are two-byte (16-bit) cyclic redundancy checks generated at the transmitting station. The receiving station also generates a CRC and compares it with the received value. This CRC is the CRC-CCITT polynomial.

Appendix C

Generating a TX5 Operating System

C.1 GENERAL

This appendix describes the procedures necessary to generate a TX5 operating system for use as a secondary station in the DX10 HDLC Communications Package. Secondary TX5 operating systems must be generated at the DX10 primary station by executing the appropriate GENTX5 utility. Individuals responsible for TX5 system generation should refer to the TX5 Operating System Programmer's Guide to become familiar with the process before attempting to perform the system generation required to include the communications package.

The communications register unit (CRU) address for the local-line module (LLM) in the secondary station must be >20 if the secondary operating system is to be downloaded from the primary station. Also, LUNO >C0 must be assigned to the LLM board and must not be assigned to any other device.

If the secondary station is equipped with a ROM communications loader, the following applies for the 990/5 computer:

Load Priority: 1 - MDU
2 - TILINE Diskette Drive
3 - LLM

If remote SCI is to be included in the TX5, the remote SCI generator program (LREMSCI) must be executed before the operating system is generated. LREMSCI produces a link control stream that is executed with the Link Editor. The object file specified in the link edit process is then included in the TX5 link control stream (see Section 3 of this manual).

It is possible to include both the operator communication package (OCP) and remote SCI (REMSCI) in a TX5 secondary station. If REMSCI is required but OCP is not, include the REMSCI log task and the dummy log device service routine (DSR). In this case, the TX5 warm start and diagnostic messages are directed to the DX10 system log.

NOTE

> If both REMSCI and OCP are required, do not include the REMSCI log task or the dummy log DSR. In this case, the warm start and diagnostic messages are directed to the TX5 log.

The following paragraphs give an example of TX5 operating system generation for a secondary station. The example includes only the required communications package software. If the secondary operating system is to be downloaded from the primary station, applications programs should be included in the system generation. These resident programs are then linked with the TX5 operating system and do not require installation separate from the download process. In the following example system generation, remote SCI (REMSCI) is included and OCP is not included.

C.2 EXAMPLE TX5 SYSTEM GENERATION

TX5 system generation is activated by executing the GENTX5 utility and responding to the prompts as shown in Figure C-1.

Respond YES to the INCLUDE FILE MANAGEMENT? prompt if the remote SCI functions Assign LUNO (AL) or Release LUNO (RL) are to be included in the TX5 system. This causes the tasks FMP1 and FUR (IDs >F0 and >F1) to be included in TASKDF. One or both may be required depending on the assign and release LUNO functions required. If file management is to be included in the TX5, both tasks must also be included and LUNO assignments can be made to either devices or file pathnames. If LUNO assignments are to be made only to devices, only FUR must be included along with the TX5 system modules indicated in the sample TX5 link stream below. Task IDs >23 through >28 are used by the HDLC communications package. Task IDs >F0 and >F1 may also be used if file management is required (or if the assign/release LUNO functions are required).

Upon completion of system generation, the TASKDF and TXDATA files must be assembled using the SCI command XMA as follows:

```
[ ] XMA
    EXECUTE MACRO ASSEMBLER
           SOURCE ACCESS NAME: <TASKDF or TXDATA filename>
           OBJECT ACCESS NAME:
          LISTING ACCESS NAME:
            ERROR ACCESS NAME:
                      OPTIONS:
       MACRO LIBRARY PATHNAME:
                  PRINT WIDTH: 80
                  PAGE LENGTH: 60
```

Enter the appropriate filenames after the SOURCE ACCESS NAME and OBJECT ACCESS NAME prompts and the appropriate filename or device name after the LISTING ACCESS NAME prompt. The remainder of the prompts are answered in accordance with the instructions in the DX10 Operating System Reference Manual, Volume 3, Application Programming Guide.

```
[]GENTX5
LISTING ACCESS NAME: (NONE)

  GENTX5   3.2.0       TX5 SYSGEN

MEMORY AVAILABLE: (2000)

SYSGEN FOR DS990? (NO)
LINE FREQ:  (60)
TIME SLICING?(YES)
TIME SLICE VALUE: (1)
TASK SENTRY? (NO)
COMMON SIZE: (170)
POWER-FAIL RECOVERY? (NO)
# OF EXP CHASSIS: (0)

  DEVICE NAME:  LL01             (* Suggested name for the LLM
  DEVICE TYPE:  SD
  TILINE DEVICE? (NO)
  CRU BASE ADDR:  >20            (* >20 required for the LLM
  ACCESS MODE:                   (* Enter RETURN for record
  INT LEVEL IN MAIN CHASSIS: 4   (* Suggested interrupt level
  TIME-OUT COUNT:  2000          (* Increase this value for rates
                                    less than 9600 bps
  CRU INT BIT: (15)
  LABEL OF DSR:  HDLCSR
  LABEL OF COMMON INT ROUTINE: IDL000
  LABEL OF UNSOLICITED INT ROUTINE: IDLINT
  EXTENSION DATA:  !DATA 0,0,0   (* DSR scratch space
  EXTENSION DATA:                (* Enter RETURN to continue

  DEVICE NAME:  DLOG             (* Specifies REMSCI without OCP
  DEV TYPE:  SD
  TILINE DEVICE? (NO)
  CRU BASE ADDR:  0              (* Based on system configuration
  ACCESS MODE:
  INT LEVEL IN MAIN CHASSIS: 15  (* Entry required to continue,
                                    interrupt not used
  TIME-OUT COUNT:                (* Time-out ignored by dummy DSR
  CRU INT BIT: (15)              (* Entry required to continue
  LABEL OF DSR:  DUMLOG
  LABEL OF COMMON INT ROUTINE:  DUMIDL
  LABEL OF UNSOLICITED INT ROUTINE:  DUMRT
  EXTENSION DATA:                (* No extension data required
  DEVICE  NAME:                  (* No more associated devices

SVC # -                          (* No special SVCs required
XOP -                            (* No special XOPs required
```

Figure C-1 Generating a TX5 System (Sheet 1 of 2)

```
INCLUDE FILE MANAGEMENT? (YES)   (* See paragraph C2
INCLUDE DIAGNOSTIC TASK? (YES)   (* Yes for OCP and DUMLOG DSR
INCLUDE OCP? (NO)                (* NO if OCP not desired
INCLUDE CONTROL PROGRAM? (YES) N

DEFINE USER TASK(S)? (NO)  Y
TASK ID:  >23                    (* Data link control task
  PRIORITY LEVEL: 0
  INITIAL STATE: (0) 3           (* Active at warm start
  INITIAL DATA LABEL:  L2ENT
TASK ID:  >24                    (* Packet-level task
  PRIORITY LEVEL:  1
  INITIAL STATE: (0) 3           (* Active at warm start
  INITIAL DATA LABEL:  L3ENT
TASK ID: >25                     (* UI Processor
  PRIORITY LEVEL:  1
  INITIAL STATE: (0)             (* Bid by L2ENT,
  INITIAL DATA LABEL:  UIENT         inactive at warm start
TASK ID: >26                     (* Communications timer
  PRIORITY LEVEL:  1
  INITIAL STATE: (0) 3           (* Active at warm start
  INITIAL DATA LABEL:  NETIME
TASK ID: >27                     (* Enter if remote SCI required
  PRIORITY LEVEL:  1
  INITIAL STATE: (0) 3           (* Active at warm start
  INITIAL DATA LABEL:  REMSCI
TASK ID: >28                     (* Enter if remote SCI log task
  PRIORITY LEVEL:  1                 without OCP is included
  INITIAL STATE: (0)             (* Bid by DUMLOG DSR
  INITIAL DATA LABEL:  LOGTSK
TASK ID:
MULTIPLE DYNAMIC TASK(S)? (NO)
CONSOLE DEVICE NAME:
DEFAULT PRINTER: (DUMY)
ASSIGN LUNO - >20                (* Required for LLM
  DEVICE NAME:  LL01             (* Must match installed name
ASSIGN LUNO - 0                  (* Required for remote SCI log
  DEVICE NAME:  DLOG                 task without OCP

ASSIGN LUNO -                    (* No more LUNOs to assign

# OF SPARE DEV LUNO BLKS: (6)
# OF DEFAULT BUFFERS: (0)
# OF GENERAL BUFFERS: (0)
TASKDF OUTPUT FILE NAME: (NONE)  (* Enter filename for TASKDF
TXDATA OUTPUT FILE NAME: (NONE)  (* Enter filename for TXDATA

END TX5 3.2.0 SYSGEN
```

Figure C-1 Generating a TX5 System (Sheet 2 of 2)

After the TASKDF and TXDATA files have been assembled, a link edit control file must be accessed (or created) to link the generated TX5 system. This file must be edited to include any resident tasks or system modules (XOP, DSR, etc.) that were included during system generation. The file must also include the network information tables built during network generation, remote SCI and the log task if required, and the communications tasks (shown in the example link control stream below). After the link control file is prepared, the TX5 system is linked using the XLE command as follows:

```
[ ] XLE
    EXECUTE LINKAGE EDITOR
           CONTROL ACCESS NAME: <name of the link control file>
     LINKED OUTPUT ACCESS NAME: <object of linkage process>
           LISTING ACCESS NAME: <list of results of linkage>
                   PRINT WIDTH: 80
```

The XLE command is described in the DX10 Operating System Reference Manual, Volume 2, Production Operation, and the TX5 Operating System Programmer's Guide.

C.3 SAMPLE LINK STREAM

In the sample link stream shown in Figure C-2, the DX10 HDLC Communications Package modules are on a directory that has been assigned the synonym TO (directory .INDSCOMM.TXOBJ). The TX5 system modules are on the volume TX5 under directories OFR, OFM, OTX, and OCP. Users should familiarize themselves further with the TX5 system generation and link edit procedures by referring to the TX5 Operating System Programmer's Guide.

Those modules required to support the device assign and release LUNO functions in Remote SCI are noted. To support file assign and release LUNO functions, all of the file management parts should be included.

After the TX5 operating system has been linked, the object file is ready for preprocessing and downloading. Alternatively, if the operating system is to be loaded from another peripheral, the object file can be copied to the media appropriate for that peripheral.

```
FORMAT        NON-COMPRESSED
TASK          TXCOMM
;INCLUDE TX5.OTX.DUMMY
;****************************
;
;             SECTION I           (* Required data structures
;
;****************************
INCLUDE       .TXDATA             (* User's TXDATA (sample name only)
INCLUDE       .TASKDF             (* User's TASKDF (sample name only)
INCLUDE       TX5.OTX.PATCH$      (* Required for patch area
;****************************
;
;             SECTION II          (* Device service routines
;
;****************************
;INCLUDE      TX5.ODR.DSR911A     (* 911 VDT DSR
;INCLUDE      TX5.ODR.KSRDSR      (* KSR 820 DSR
;INCLUDE      TX5.ODR.KSR9902     (* KSR 820 DSR if connected to 9902
;INCLUDE      TX5.ODR.LPDSR       (* 810 printer DSR
;INCLUDE      TX5.ODR.LP9902      (* 810 DSR if connected to 9902
;INCLUDE      TX5.ODR.DDIOSR      (* TILINE floppy disk DSR
;****************************
;
;        DX10 HDLC DSR'S          (* DSRs required by DX10
;                                    HDLC Communications
;****************************
INCLUDE       TO.NETL1            (* HDLC LLM DSR
INCLUDE       TO.DUMLOG           (* HDLC dummy log DSR
;****************************
;
;      DX10 HDLC MODULES          (* Software modules required by
;                                    DX10 HDLC Communications
;****************************
INCLUDE       TO.NETL2            (* HDLC task ID >23
INCLUDE       TO.NETL3            (* HDLC task ID >24
INCLUDE       TO.NETUL            (* Upper level protocol support
INCLUDE       TO.NETUI            (* HDLC task ID >25
INCLUDE       TO.NETIME           (* HDLC task ID >26
INCLUDE       TO.NETDAT           (* HDLC required data structures
INCLUDE       TO.NETSUB           (* HDLC required subroutines
INCLUDE       TO.NETSVC           (* HDLC SVC processor
INCLUDE       .REMSCI             (* HDLC task ID >27. Sample file
;                                    name for remote SCI object code
INCLUDE       TO.LOGTSK           (* HDLC task ID >28. Enter only if
;                                    the logger task is required
INCLUDE       .NIT0100            (* Sample file name for secondary
;                                    NIT object code
```

Figure C-2 Sample TX5 Link Stream (Sheet 1 of 4)

```
;****************************
;
;           SECTION III       (* File utility modules
;
;****************************
INCLUDE     TX5.OFR.FMPDEFS   (* Required for AL and RL in REMSCI
;INCLUDE     TX5.OFR.ALLOC
INCLUDE     TX5.OFR.ALSVC     (* Required for AL and RL in REMSCI
;INCLUDE     TX5.OFR.BLDDIR
;INCLUDE     TX5.OFR.BLDFDR
INCLUDE     TX5.OFR.BLDLDT    (* Required for AL and RL in REMSCI
;INCLUDE     TX5.OFR.CMPSCD
;INCLUDE     TX5.OFR.CNMSVC
;INCLUDE     TX5.OFR.CONVRC
;INCLUDE     TX5.OFR.CRESVC
;INCLUDE     TX5.OFR.DEALLC
;INCLUDE     TX5.OFR.DELENT
;INCLUDE     TX5.OFR.DFSVC
;INCLUDE     TX5.OFR.FLOPEN
;INCLUDE     TX5.OFR.FNDFIL
INCLUDE     TX5.OFR.FURDAT    (* Required for AL and RL in REMSCI
INCLUDE     TX5.OFR.FURPRC    (* Required for AL and RL in REMSCI
;INCLUDE     TX5.OFR.HASH
;INCLUDE     TX5.OFR.PRTSVC
;INCLUDE     TX5.OFR.RDPBM
;INCLUDE     TX5.OFR.READ
;INCLUDE     TX5.OFR.RESAU
INCLUDE     TX5.OFR.RLSVC     (* Required for AL and RL in REMSCI
;INCLUDE     TX5.OFR.SCNPBM
;INCLUDE     TX5.OFR.SERDIR
;INCLUDE     TX5.OFR.SETBMP
INCLUDE     TX5.OFR.SRCDNT    (* Required for AL and RL in REMSCI
INCLUDE     TX5.OFR.STACK     (* Required for AL and RL in REMSCI
;INCLUDE     TX5.OFR.SYNSVC
;INCLUDE     TX5.OFR.UPDDOR
INCLUDE     TX5.OFR.VOLUME    (* Required for AL and RL in REMSCI
```

Figure C-2 Sample TX5 Link Stream (Sheet 2 of 4)

```
;*****************************
;
;           SECTION IV        (* File management modules
;
;*****************************
;INCLUDE     TX5.OFM.ADVPTR
;INCLUDE     TX5.OFM.BLDFCB
;INCLUDE     TX5.OFM.BLKREC
INCLUDE     TX5.OFM.CHKACC    (* Required for AL and RL in REMSCI
;INCLUDE     TX5.OFM.CLOSE
;INCLUDE     TX5.OFM.FDRUP
;INCLUDE     TX5.OFM.CNVTBK
;INCLUDE     TX5.OFM.CNVTRC
;INCLUDE     TX5.OFM.DIRTY
INCLUDE     TX5.OFM.FMDAT1    (* Required for AL and rl in REMSCI
INCLUDE     TX5.OFM.FMPTSK    (* Required for AL and rl in REMSCI
;INCLUDE     TX5.OFM.GETBLK
;INCLUDE     TX5.OFM.GETRBK
INCLUDE     TX5.OFM.GETTSB    (* Required for AL and RL in REMSCI
;INCLUDE     TX5.OFM.OPEN
;INCLUDE     TX5.OFM.OPNEXT
;INCLUDE     TX5.OFM.RDBLK
;INCLUDE     TX5.OFM.RECLCK
;INCLUDE     TX5.OFM.RELRD
;INCLUDE     TX5.OFM.RELSVC
;INCLUDE     TX5.OFM.RELWRT
;INCLUDE     TX5.OFM.RETAU
INCLUDE     TX5.OFM.SEMAPH
;INCLUDE     TX5.OFM.SEQBKS
;INCLUDE     TX5.OFM.SEQRD
;INCLUDE     TX5.OFM.SEQSVC
;INCLUDE     TX5.OFM.SEQWRT
;INCLUDE     TX5.OFM.UNBKRC
;INCLUDE     TX5.OFM.UNLALL
;INCLUDE     TX5.OFM.UNLSVC
;INCLUDE     TX5.OFM.WRTBLK
```

Figure C-2 Sample TX5 Link Stream (Sheet 3 of 4)

```
;****************************
;
;           SECTION V           (* Operator communications
;
;****************************      package (OCP). Refer to
;INCLUDE     TX5.OCP.OCPTSK        programmers guide for further
;INCLUDE     TX5.OCP.OCPTBL        information
;INCLUDE     TX5.OCP.OCPPRC
;INCLUDE     TX5.OCP.OCPIOU     (* Single dynamic task systems
;INCLUDE     TX5.OCP.DOCPTSK    (* Multiple dynamic task systems
;INCLUDE     TX5.OCP.OCPTLD
;INCLUDE     TX5.OCP.OCPSLD
;INCLUDE     TX5.OCP.OCPTAD
;INCLUDE     TX5.OCP.OCPLRT     (* Single dynamic task systems
;INCLUDE     TX5.OCP.DOCPLRT    (* Multiple dynamic task systems
;INCLUDE     TX5.OCP.OCPEND
;****************************
;
;           SECTION VI          (* Control program.
;
;****************************
;INCLUDE    TX5.SYS.CNTRL
;****************************
;                               (* Operating system modules
;           SECTION VII            Refer to TX5 programmers
;                                  guide
;****************************
INCLUDE     TX5.OTX.TXROOT      (* 990/10 and 990/5 systems only
INCLUDE     TX5.OTX.IOSUPR
INCLUDE     TX5.OTX.TSKFUN
INCLUDE     TX5.OTX.TSKLDR      (* Single dynamic task systems only
INCLUDE     TX5.OTX.CNVRSN
INCLUDE     TX5.OTX.MEMSVC      (* Single dynamic task systems only
INCLUDE     TX5.OTX.TBUFMG
INCLUDE     TX5.OTX.DTASK
INCLUDE     TX5.OTX.EVENTK      (* 990/10 and 990/5 systems only
INCLUDE     TX5.OTX.GETEVT      (* 990/10 and 990/5 systems only
INCLUDE     TX5.OTX.TXSTRT      (* 990/10 and 990/5 systems only
INCLUDE     TX5.OTX.TXEND       (* 990/10 and 990/5 systems only
INCLUDE     TX5.OTX.STASK
END
```

Figure C-2 Sample TX5 Link Stream (Sheet 4 of 4)

Appendix D

Communications Structures

D.1 GENERAL

This appendix contains information required to program secondary stations to operate over a permanent virtual circuit (PVC) using X.25 procedures. The requirements for generating and handling the following structures are discussed:

* The logical channel identifier
* Various control packets:
  - Reset request
  - Reset confirmation
  - Receive ready
  - Receive not-ready
* The data packet

The method of addressing that provides program to program communication is also described.

D.2 LOGICAL CHANNEL IDENTIFIER (LCI) ASSIGNMENT

Proper operation of the level 3 software requires the assignment of an LCI to each secondary station. Because it is not proper for either level 3 to interprete a level 2 frame command or for level 2 to assign an LCI for level 3, it is recommended that an unnumbered-information (UI) processor be added to the secondary station software. Level 2 can call this processor when a UI command is received. In this way, if the need arises to expand UI control, only the UI processor is affected. The primary station transmits the LCI message to the secondary station during warm-start or after a secondary station is downloaded. The secondary station only interprets the LCI frame, it never sends it. It cannot accept or send data or flow control packets until the LCI is assigned.

The format of the LCI assignment frame is as follows:

BYTE	0	1	2	3	4	5	6,7	8
	FLAG	ADDR.	UI	>00	0001 GGGG	CCCC CCCC	FCS	FLAG

2280620

where:

Byte 0	is the start-of-frame flag, >7E.
Byte 1	is the switch address of the secondary station.
Byte 2	is the HDLC UI command with P-bit set to 1.
Byte 3	is the I (information) field descriptor. This is normally set to zero but may be used for UI control expansion, if required.
Byte 4	is set to 0001, the general format identifier. GGGG is the virtual channel group number of the secondary station.
Byte 5	is the logical channel number of the secondary station.
Bytes 6, 7	are the 16-bit CRC HDLC frame check.
Byte 8	is the end-of-frame flag, >7E.

When the secondary station receives this message its level 2 software activates the UI processor and responds with an unnumbered acknowledge (UA) message to the primary station The secondary station then assigns GGGG CCCC CCCC as its LCI. This LCI will be present in every packet sent on the permanent virtual circuit (PVC).

D.3 PACKET LEVEL (LEVEL 3) PROCEDURES

The following paragraphs describe the level 3 interface procedures that are used for communications between primary and secondary stations. Each packet transmission is contained in an HDLC numbered information (I) field. Only one packet is contained in each I field and only the data transfer packet is accompanied by user data. The types of packet are as follows:

- Reset request (RQ) packet
- Reset confirmation (RC) packet
- Receive ready (RR) packet
- Receive not-ready (RNR) packet
- Data packet

D.3.1 Reset Request Packet

A reset request packet is transmitted when any of the following occur:

- An invalid packet send count, P(S), is found in a data packet.
- An invalid packet receive count, P(R), is found in data packet, an RNR packet, or an RR packet.
- An incorrect packet length is detected.
- The general format identifier is not 0001.
- Multiple retransmission attempts result in failure.
- A time-out occurs after sending a reset request. Normally, the secondary station sends an RQ packet and then awaits a reset confirmation packet. If the reset confirmation packet is not detected within a predetermined time the RQ packet is retransmitted. Transmission of the RQ packet is repeated until either a reset confirmation or an RQ packet is received. In either event the reception is treated as a reset confirmation since both stations are apparently attempting to reset.

The following conditions are ignored by the packet level and do not require reset action:

- Reception of a packet with an invalid LCI.
- Reception of a packet of unknown type.

When a reset request packet is transmitted the following events should occur:

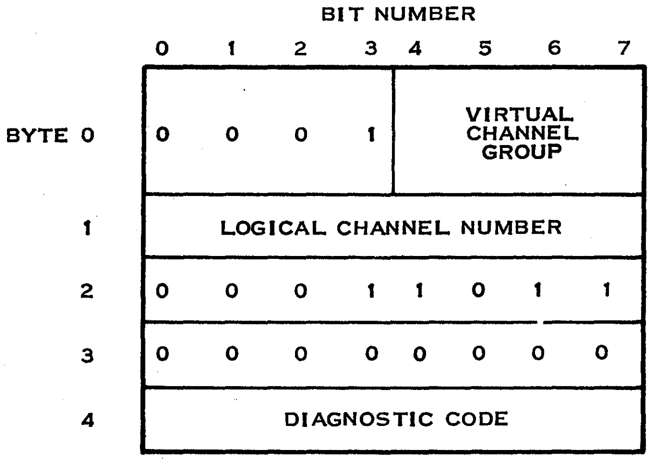
- The packet level send counter, P(S), is set to zero
- The packet level receive counter, P(R), is set to zero

- * Any packets awaiting transmission are re-submitted for packet level processing

- * A retransmission attempt is initiated - Retransmission attempts should be counted and a maximum of 3 to 5 retries is recommended for most types of secondary stations. If the attempts still fail, then the transmitting device must decide how to handle the packets still in the queue for transmission.

The reset procedure causes queued packets to be renumbered. Packets that are unacknowledged when a reset request packet is transmitted will remain unacknowledged thus raising the possibility of duplicate packet transmission. It is recommended that applications programs should be programmed accordingly and duplication of data be considered preferable to deletion.

The format of the reset request packet is as follows:



	BIT NUMBER							
	0	1	2	3	4	5	6	7
BYTE 0	0	0	0	1	VIRTUAL CHANNEL GROUP			
1	LOGICAL CHANNEL NUMBER							
2	0	0	0	1	1	0	1	1
3	0	0	0	0	0	0	0	0
4	DIAGNOSTIC CODE							

2280621

where:

Byte 0 is the general format identifier, 0001, followed by the virtual channel group (VCG) number.

Byte 1 is the logical channel number (LCN). VCG and LCN together form the logical channel identifier (LCI).

Byte 2 is the reset packet type identifier.

Byte 3 is always set to zero.

Byte 4 is the diagnostic code that points to the type of problem encountered.

Current valid diagnostic codes for byte 4 are as follows:

Bits 0,1,2	Bits 3 through 7	Meaning
000	00000	Problem undefined.
P(R)	00001	Bad P(R) returned.
P(S)	00010	Bad P(S) returned.
000	00011	Packet is incorrect length.
000	00100	General format identifier is not 0001.
000	00101	Data packet has been sent too many times.
000	00110	Station in receive-not-ready condition too long.
000	00111	Invalid packet format detected.
P(R)	01000	M data bit set in data packet header.

D.3.2 Reset Confirmation Packet

A reset confirmation packet is transmitted to acknowledge the reception of a reset request packet after the reset procedures have been performed (see the previous paragraph). However, if a reset request packet has been sent and a reset request packet received the transmission of a reset confirmation packet is not required. In this case, process the received reset request packet as a reset confirmation packet. The format of a reset confirmation packet is as follows:

	BIT NUMBER							
	0	1	2	3	4	5	6	7
BYTE 0	0	0	0	1	VIRTUAL CHANNEL GROUP			
1	LOGICAL CHANNEL NUMBER							
2	0	0	0	1	1	1	1	1

2280622

where:

Byte 0 is the general format identifier, 0001, followed by the virtual channel group (VCG) number.

Byte 1 is the logical channel number (LCN). VCG and LCN together form the logical channel identifier (LCI).

Byte 2 is the reset confirmation identifier.

D.3.3 Receive Not Ready Packet

The conditions to be considered for the RNR packet are as follows:

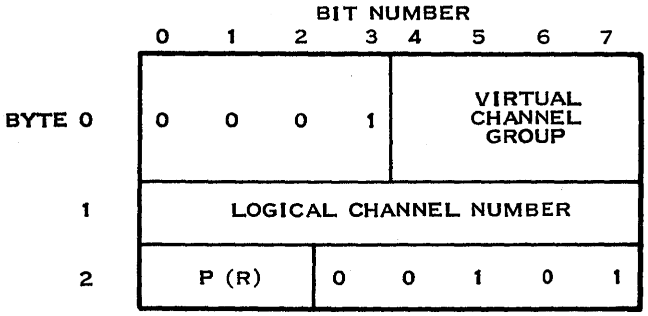
- When an RNR is transmitted, that is, when the sender is unable to receive data packets on the PVC. This condition clears when any of the following occurs:
  - The transmission of a receive ready packet after the RNR condition is corrected.
  - The reception of a reset request packet.
  - The transmission of a reset request packet.
- When an RNR is received, that is, when the receiving station is unable to send data packets on the PVC. This condition clears when any of the following occurs:
  - A receive ready packet is received
  - A reset request packet is received
  - A reset request packet is sent and a reset request or a reset confirmation packet is received in reply.

After receiving an RNR, the receiving station can continue to receive data packets on the PVC.

When the RNR condition has persisted for too long (as determined by the receiver of the RNR) a reset request packet is transmitted.

The P(R) and P(S) counters are not reset when an RNR is sent or received. If the station that sent the RNR continues to send data packets, the P(S) counter is updated with each data packet sent. The P(R) counter indicates the next expected P(S) count if the RNR condition is cleared without resetting.

The format of the RNR packet is as follows:



2280623

where:

Byte 0 is the general format identifier, 0001, followed by the virtual channel group (VCG) number.

Byte 1 is the logical channel number (LCN). VCG and LCN together form the logical channel identifier (LCI).

Byte 2 is the RNR packet identifier.

D.3.4 Receive Ready Packet

A receive ready (RR) packet is sent under the following conditions:

- After an RNR packet is transmitted and the transmitting station is again able to receive data packets, that is, the RNR condition is cleared.
- When a receive window is full, to acknowledge data packets if no data packets are queued for transmission. Refer to paragraph D.4 for further information on windows.

A receive ready packet is received under the following conditions:

- After the reception of an RNR packet and the sending station has cleared the RNR condition.
- To acknowledge receipt of a data packet.

The format of the receive ready packet is as follows:

BIT NUMBER

	0	1	2	3	4	5	6	7
BYTE 0	0	0	0	1	VIRTUAL CHANNEL GROUP			
1	LOGICAL CHANNEL NUMBER							
2	P (R)			0	0	0	0	1

2280624

where:

Byte 0 is the general format identifier, 0001, followed by the virtual channel group (VCG) number.

Byte 1 is the logical channel number (LCN). VCG and LCN together form the logical channel identifier (LCI).

Byte 2 is the receive ready packet identifier.

D.3.5 Data Packet

The data packet header is appended only to user data sent on the PVC; it is not appended to the control packets previously described. Data packets are sequentially numbered using the P(S) parameter and carry an embedded acknowledgement of the last data packet received, P(R). P(R) is the sequence number of the next expected data packet. P(R) and P(S) counters are maintained for each PVC.

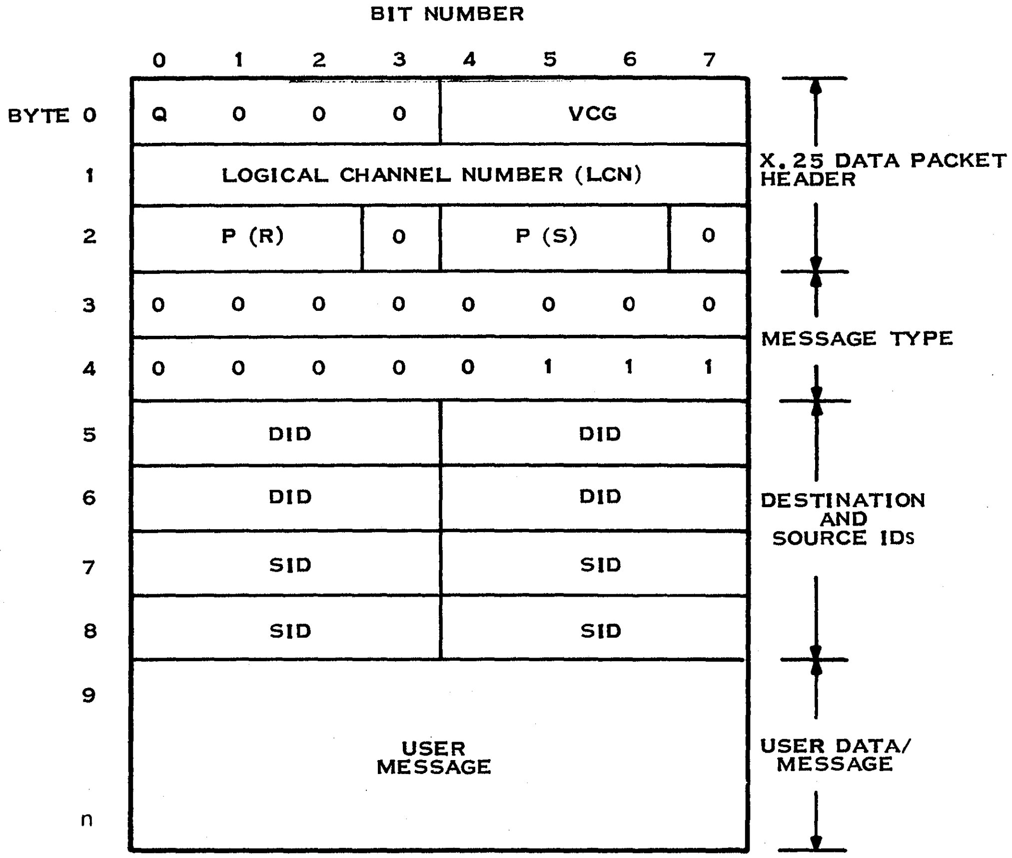
A data packet can be transmitted when the following conditions are satisfied:

* The LCI is assigned and the PVC is in the data transfer state.

* An RNR has not been received, or if one has been received, the RNR condition no longer exists.

* The send window is not full.

The following format depicts a complete data packet after the header has been added:



BIT NUMBER

	0	1	2	3	4	5	6	7	
BYTE 0	Q	0	0	0	VCG				X.25 DATA PACKET HEADER
1	LOGICAL CHANNEL NUMBER (LCN)								
2	P (R)			0	P (S)			0	
3	0	0	0	0	0	0	0	0	MESSAGE TYPE
4	0	0	0	0	0	1	1	1	
5	DID				DID				DESTINATION AND SOURCE IDs
6	DID				DID				
7	SID				SID				
8	SID				SID				
9	USER MESSAGE								USER DATA/ MESSAGE
n									

2280625

where:

Bytes 0, 1, 2 are the packet header, consisting of:

- A data qualifier bit, Q. This is always set to zero.
- VCG, the line number to which the device is attached.
- LCN, the HDLC drop address for the device.
- P(R), the receive packet level counter

- A more-data bit, M. This bit must always be set to zero.
- P(S), the transmit (send) packet level counter

Bytes 3 and 4 are the session control word. This is fixed in value at >0007 for all data packets.

Bytes 5 and 6 are destination IDs coded in binary coded decimal and assigned during network generation.

Bytes 7 and 8 are source IDs coded in binary coded decimal and assigned during network generation.

Bytes 9, etc. are user data. The maximum size is usually 256 or 260 bytes.

D.4 DATA PACKET FLOW

The orderly flow of data is enhanced by defining a window size for each permanent virtual circuit (PVC). The window size specifies the maximum number of sequentially numbered data packets that a station is authorized to transmit and have outstanding at any given time without receiving a receive ready packet as an acknowledgement between packet transmissions. With modulo 8 arithmetic a window size of 1 through 7 can be specified but the size must be the same at the primary and secondary station for a given PVC. A window size of 1 specifies that the sending station must receive an acknowledgement before sending the next data packet. A window size of 7 specifies that the sending station may not transmit more than 7 data packets or have more than 7 outstanding before receiving an acknowledgement. The window size is determined by the memory available for buffering input and output data packets, including the memory required to manage these buffers.

If the receiving station window is exceeded, a procedural error has occurred and a reset request packet should be sent as a reply.

Examples of transmitted data packets follow.

D.4.1 Packet with User Data

When a data packet is transmitted over the data link the format appears on the line as follows:

			I FIELD				
FLAG	ADDR.	I	PACKET HEADER	X	USER DATA	CRC	FLAG

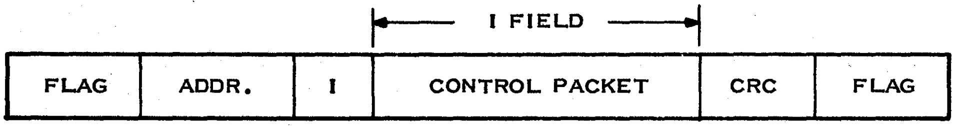
2280626

where:

Flag	is the start-of-frame and end-of-frame flag, >7E.
Addr	is the switch address of the secondary station.
I field	includes the packet header, X (see next entry), and the user data.
X	is the session control word and the destination and source IDs.
CRC	is the frame check character.

D.4.2 Packet with No User Data

When a control packet is transmitted over the data link the format appears on the line as follows:



2280627

where:

Flag	is the start-of-frame and end-of-frame flag , >7E.
Addr	is the switch address of the secondary station.
I field	includes only the control packet, that is, a receive ready, a reset request, a reset confirmation, or a receive not-ready packet.
CRC	is the frame check character.

D.5 TASK LEVEL IDENTIFIERS

The three levels of X.25 are the physical link, the data link procedures and the packet level procedures. To provide for task to task, or program to program, communications a supplimentary method of addressing is available. This method provides for addressing uniquely identified programs or devices in the distributed environment. The addresses are refered to as Destination IDs and Source IDs. There is a unique ID assigned to TM990 and PM550 secondary stations. In data packets arriving at the secondary stations the ID is refered to as the destination ID. In data packets sent from the secondary station the same ID is refered to as the source ID. The IDs must be included in every data packet entering and leaving the devices. To send a data packet to the primary station, TM990 and PM550 do the following after receiving the first input data packet:

1. Include the session control word in the output data packet.
2. Take the destination and source IDs received in the input data packet and reverse their relative positions for the output data packet. In other words, make the source ID in the received data packet the destination ID in the send data packet, and vice-versa.

Communications between primary station tasks and TM990 or PM550 should be peformed in accordance with this appendix.

D.5.1 Network Information Table (NIT)

This section describes the purpose of the destination and source IDs and how they are used by the primary station software.

A Network Information Table (NIT) in the primary station includes the information required to route messages from primary station tasks to secondary station tasks or devices. There are two table formats defined. The long form is an output format, the short form is an input format. The input format is refered to as the short form because it includes the minimum information required to route input messages to tasks at the local station. For each PVC supported in the local line network, there is one output format NIT entry. At each station that uses the the NIT formats, there is one input format NIT entry for each task that uses the communications services.

TX5 secondary stations also use the NIT input and output formats. The requirements for, and format of, NIT entries at the secondaries is essentially the same as at the primary station. For those stations that do not support multi-tasking (such as TM990 and PM550) no NIT is required.

The NETGEN program that builds the NIT tables executes a series of prompts to an interactive device. This program uses the responses to these prompts to build the NIT entries for the primary station or the TX5 secondary stations. The user of this program must know the local line network configuration in order to properly build the tables. This program is further described in Section 3.3.

The concept of 16-bit destination and source IDs is derived from the need to extend the range of data link addresses (currently limited to 64 for LLM type devices). These IDs are also used to address logical entities (tasks or programs) within a distributed computer system. For each physical address (such as LLM or PM550 switchs) and for each task ID in a TX5 secondary station, a unique network address is assigned. This address has a decimal range of 0 through 9,999, providing up to 10,000 unique IDs within a network. Use of these IDs allows duplication of physical addresses (such as LLM switch addresses) and logical addresses (such as task IDs) within the network. One ID is assigned to one device or one task, and there are no duplicate IDs within the network.

This ID concept supports the download function required by some secondary stations. The primary station download task cannot, and should not, address LLM type switch addresses in the network since these addresses may be duplicated in a large multipoint configuration (see Section 2). Assigning an ID to each LLM type switch address enables the download task to address that ID. The communications software, using the NIT, then resolves the problem of routing download messages to the appropriate multipoint line.

Assignment of an ID to each secondary station TX5 task allows the use of the same task ID in more than one TX secondary station. For example, task ID >80 in a secondary station with LLM switch address >1E on one multipoint local line can be assigned ID 1001, while the same task ID and switch address on another multipoint local line can be assigned ID 2001. This scheme requires that the sending task in the primary station specify either ID 1001 or ID 2001 depending on the station and task that is to receive the message. Section 2 contains a diagram depicting this scheme.

D.6 DOWNLOAD TRANSMISSION SEQUENCE

Secondary stations can be downloaded either automatically when a communications warm-start is initiated, or by using the DOWNLOAD command. The following paragraphs show the sequence of events in terms of the frames transmitted from the primary to secondary stations and vice versa.

In the frames shown below, the following terms are used:

Flag	is the HDLC start-of-frame and end-of-frame delimiter.
Addr	is the switch address of the receiving station.
CRC	is the frame check character.
>17	is the code for the HDLC RIM (request initialization mode) or SIM (set initialization mode) command.
>13	is the code for the HDLC UI (unnumbered information transfer) command.
>73	is the code for the HDLC UA (unnumbered acknowledge) command.
DATA	is the program being downloaded.

D.6.1 Download by Secondary Request

When the download operation is automatically accomplished the following sequence of events occurs:

1. The host polls the secondary station.
2. The secondary station sends a download request (RIM).
3. The host sends the first frame containing program data (UI).
4. The secondary station acknowledges (UA).
5. The host sends the second frame containing program data (UI).
6. The secondary station acknowledges (UA).

This continues until the download completes.

The sequence is shown in the following diagram:

2280628

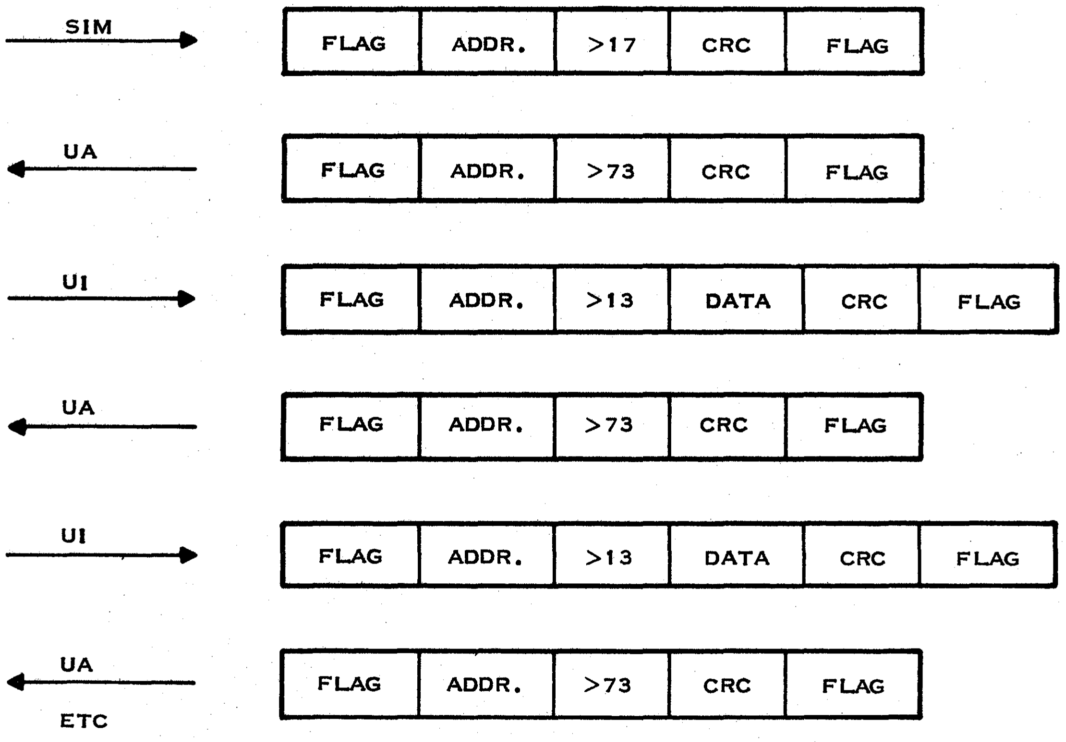
D.6.2 Download by User (SCI) Request

When the download operation is requested by the use of the SCI command DOWNLOAD the following sequence of events occurs:

1. The host sends an initialization command (SIM) to the secondary station.
2. The secondary station acknowledges (UA).
3. The host sends the first frame containing program data (UI).
4. The secondary station acknowledges (UA).
5. The host sends the second frame containing program data (UI).
6. The secondary station acknowledges (UA).

This continues until the download completes.

The sequence is shown in the following diagram:



2280629

The frames containing program data are shown in more detail in the following diagram:

FLAG	ADDR.	>13	LOAD ADDRESS	WORD COUNT	LOAD DATA	CRC	FLAG
(BYTE)	(BYTE)	(BYTE)	(WORD)	(WORD)	(WORDS)	(WORD)	(BYTE)

2280630

The word count is the number of words contained in the data field; the load address is the memory starting address for loading the data at the secondary station. The load data is binary load data which has already relocated. The data is relocated by using the preprocessor command PREDB, (see paragraph 5.2.2). The last data frame will have a zero in the word count field and the load address field will contain the program start vector. The start vector is given using the assembler END <start vector> statement. If no start vector is supplied by the program the address frame will contain a zero.

GLOSSARY

This appendix contains a glossary of the terms used in this manual. The definition of terms provided apply specifically to the DX10 HDLC Communications Package as they are used in this manual.

Activation services

Activation services is a part of the communications package software for use by applications programs that expect input data from other stations. A call to activation services is executed by a level 15 XOP with opcode >4D, where the I/O opcode ranges from 4 to 7 and specifies the function to be performed.

CCITT

An abbreviation for the Consultative Committee for International Telegraph and Telephone. The CCITT is a standards organization set up to study digital networks. The DX10 HDLC Communications Package is a subset of the X.25/HDLC developed by the CCITT.

Communications buffers

Input and output buffers allocated by the communications package to handle data transfers between stations in the network.

Communications line

A line consisting of a single shielded, twisted-pair cable that interconnects the stations in a local line network. The cable is connected from the FCCC board in the primary station to the LLM boards in the secondary stations.

Communications register unit

See CRU.

Communications tasks

Software programs in the communications package included in both primary and secondary stations that provide communications services to the network.

CRU (communications register unit)

The direct command-driven serial input/output interface of the 990 computer.

Data buffers

Storage areas allocated within an application program to hold data required by the program.

Data Link Level - (Level 2)

Level 2 software provides data link control for an unbalanced (multipoint) line operating in the HDLC "normal response" mode. A subset of the International Standard High Level Data Link Control (HDLC) protocol is specified for use with the DX10 HDLC Communications Package. Included in this control are message sequence counting and HDLC frame formatting procedures.

Device service routine

See DSR.

Destination ID

See DID.

DID (destination ID)

The network ID assigned to stations or applications programs that have data sent to it.

Download directory

A directory maintained by the communications package that contains the file names of formatted operating systems for secondary stations and the station IDs to which the operating systems are assigned.

Download preprocessor

The utility program in the primary station that converts a standard DX10 object file into a relocated binary formatted object file and stores the object file in the download directory.

Download task

This task accesses the files built by the download preprocessor and transfers records from these files to the downloading secondary station via the communications package.

Downloading

A procedure by which the operating system or stand-alone program for a secondary station is sent from the primary station over the communications line. TX secondary stations require an DX10 HDLC Communications ROM loader for downloading.

DSR (device service routine)
A program that provides the software interface by servicing interrupts and performing input and output operations between a device and the system software. The communications package includes a special DSR that provides the TILINE interface between the CPU and the FCCC card.

FCCC
The Four Channel Communications Controller board that provides the electrical interface between the primary station and the communications line.

HDLC
An abbreviation for High Level Data Link Control.

LCI
The logical channel identifier is made up of a VCG/LCN (Virtual Channel Group/Logical Channel Number) that is assigned when the NETGEN program is executed.

LLM (local line module)
A CRU device installed in the computer chassis that provides the physical interface for TX5 secondary stations to the network.

LUNO (logical unit number)
A number which specifies the device for an input/output operation.

Modem
A device which (1) converts a digital waveform to an analog waveform suitable for transmission, and (2) converts the received analog waveform to digital during reception.

Multipoint line
A single communications line that connects several secondary stations to one primary station. All secondary stations on the same multipoint line are connected to the same FCCC port in the primary station.

Network
A configuration of primary and secondary stations interconnected by cables.

Network Generator Programs - NETGEN and TXNETGEN
The programs that create the network information table (NIT) entries. NETGEN creates the NIT entries for the primary station that identify secondary stations and primary applications programs that use the communications package. TXNETGEN creates the NIT entries for the secondary stations that identify secondary task that use the communications package.

Network ID

A four-digit binary coded decimal value that ranges from 0000 to 9999. Each station and applications program that uses the communications package is assigned a unique network ID. Network IDs 0000 through 0099 are reserved for use by the communications package. The network ID becomes the SID when data is sent from a location; the network ID becomes the DID when data is addressed to a location.

NIT (Network Information Table)

The network information table is a memory resident table contained in the primary station and TX5 secondary stations. There is one long form NIT entry in the primary station for each secondary station in the network and one short form NIT entry for each applications program in the the primary station that uses the communications package. Each TX secondary station contains one long form NIT entry and one short form NIT entry for each applications program in the station that use the communications package.

Off-line

A condition pertaining to the status of a station in the network. When a station is taken off-line, it is logically disconnected and all polls normally sent to the station are inhibited. This condition may be caused by an operator action or by certain error conditions.

PVC

A permanent virtual circuit is established within the communications package that provides the logical connection between the primary station and each secondary station.

Packet Level (Level 3)

Level 3 provides the procedures that control the flow of information on the permanent virtual circuit (PVC). Each message (called a data packet) includes a data packet header and data packet sequence numbers. The sequence numbers are used to count data packets sent and received on the PVC and to limit the number of data packets that can be unacknowledged at any given time. Packets supported under the DX10 HDLC Communications Package are standardized under the CCITT X.25 recommendations. Packets without data are also sent on the PVC to indicate the receipt of other packets or the condition of the sender (i.e., busy, resetting, not busy).

PDT (physical device table)

A table that contains the device-related data required by the DSR.

Physical Link Level (Level 1)

Level 1 is the electronic interface between the stations that are controlled on the FCCC board.

Polling

An operation conducted by the polling software on the FCCC board that controls the flow of data between stations in the network. The polling operation scans the secondary stations on a nonprioritized, roll call basis to determine if any of the stations require communications services.

Preprocessing

A procedure performed on the operating system or a stand-alone program for a secondary staton to format it for downloading over the communications line (see download preprocessor).

Primary station

A Model 990/10 or /12 Computer system with a DX10 operating system (version 3.4). communications package software and hardware (FCCC and cable), and input/output devices. Control of the communications package is maintained at the primary station.

Remote SCI

Remote SCI provides a means of interacting with a TX5 secondary station from a terminal at the DX10 primary station. The Remote SCI command set is modeled after the DX10 SCI command set.

Secondary station

A Model 990/5 Computer system or a remote processing device that is connected to and communicates with the primary station via a communications line.

Session Control

The level of software in the communications package that handles all applications programs´ requests for establishing, operating, and terminating a session.

SID (source identification)

The network ID assigned to a station or applications program that sends data to another location in the network.

Source ID

See SID.

Station

A 990 computer system, including all input/output devices such as a printer or terminal that are locally-connected to the computer or any other device that can be connected to the network.

Station ID

The network ID assigned to the LLM board in the secondary station.

Switch ID

The switch selected address of the LLM board in the secondary station.

Time out

A condition that occurs when a secondary station has been sent a poll and the station does not respond within the time specified.

User task

An applications program that uses the communications package.

Warm start

A procedure that loads the station's operating system and restarts the system. The procedure consists of pressing the HALT/SIE, RESET, and LOAD switches on the computer front panel. In the primary station, an IS command to initialize the DX10 operating system and a COMMGO command to initialize the communications package must also be executed. In the secondary station, any tasks given an initial state of 3 during system generation are automatically activated at warm start.

XOP

Special extended operation (XOP) procedures included in the communications package to provide an interface between applications programs and the software within the communications package.

INDEX

The following index lists key words and concepts from the subject material of the manual together with the area(s) in the manual that supply major coverage of the listed concept. The numbers along the right side of the listing reference the following manual areas:

- Sections - References to sections of the manual appear as "Section x" with the symbol x representing any numeric quantity.

- Appendixes - References to appendixes of the manual appear as "Appendix y" with the symbol y representing any capital letter.

- Paragraphs - References to paragraphs of the manual appear as a series of alphanumeric or numeric characters punctuated with decimal points. Only the first character of the string may be a letter; all subsequent characters are numbers. The first character refers to the section or appendix of the manual in which the paragraph is found.

- Tables - References to tables in the manual are represented by the letter T followed immediately by another numeric character (representing the section of the manual containing the table). The second character is followed by a dash (-) and a number: Tx-yy

- Figures - References to figures in the manual are represented by the letter F followed immediately by another numeric character (representing the section of the supplement containing the figure). The second character is followed by a dash (-) and a number:

  Fx-yy

- Other entries in the Index - References to other entries in the index are preceded by the word "See" followed by the referenced entry.

Accumulation, Statistics 1.4.2.5
Activate 1.4.2.5
    Utilities 5.6
Activating:
    Communications 3.5
    Primary Station 3.5.1
Activation 3.3.2
    Remote SCI 5.4.1
    Request 4.2.2.2, 4.2.2.3, 4.2.2.4
Activation Services 1.4.2.4, 4.2.2
    Errors A.1.6
Add Alias 5.2.3.1
Adding:
    FCCC Board 3.3.4.1
    Primary Task 3.3.4.1
    Secondary Station 3.3.4.1
    Secondary Task 3.3.4.1
Additional:
    Lines 3.3.3.3
    Stations 3.3.3.3
Address Field B.7.2
Addresses, Station 2.2.2.2
Addressing, Network 2.2, 2.2.3, F2-1
AL Command 5.4.3.2
Alias:
    Add 5.2.3.1
    Assign 5.2.3.1
    Commands 5.2.3.1
    Delete 5.2.3.1
    Modify 5.2.3.1
Artificial Data 5.5.4
Assembling TX5 . . . . 3.4.4
Assembly Language:
    Interface F4-7
    Program F4-5, F4-6
    Subroutine 4.6.1
Assign Alias 5.2.3.1
Assign Breakpoint Command 5.4.3.1
Assign LUNO Command 5.4.3.2
Assignments, Network 2.2.2
Associate Network ID/Task ID Command 5.4.3.14

Board:
    FCCC 1.4.2.1
    LLM 1.4.2.1
Build Download File 5.2.3.2
Building:
    Primary System 3.2.1
    Remote SCI 3.4.2
    Secondary System 3.4.1

Call:
    READ4D 4.6.2.1

    WKUP4D 4.6.2.3
    WRIT4D 4.6.2.2
Calls:
    Input/Output 4.2.1
    Subroutine 4.6.2
    Supervisor 4.2
Changing an Entry 3.3.4.2
Command:
    AL 5.4.3.2
    Assign Breakpoint 5.4.3.1
    Assign LUNO 5.4.3.2
    Associate Network ID/Task ID 5.4.3.14
    DB 5.4.3.3
    Delete Breakpoint 5.4.3.3
    Display Error Codes 5.4.3.5
    Dump Workspace 5.4.3.4
    DW 5.4.3.4
    ERRS 5.4.3.5
    Execute Task 5.4.3.23
    Field B.7.3
    IDT 5.4.3.6
    Initialize Date and Time 5.4.3.6
    Install Task 5.4.3.7
    IT 5.4.3.7
    Kill Task 5.4.3.8
    KT 5.4.3.8
    LB 5.4.3.9
    Line Reset 5.5.3.3
    Line Status 5.5.3.1
    List Breakpoints 5.4.3.9
    List Memory 5.4.3.10
    List System Memory 5.4.3.11
    LL 5.5.3.1
    LM 5.4.3.10
    LR 5.5.3.3
    LS 5.5.3.2
    LSM 5.4.3.11
    MM 5.4.3.12
    Modify Memory 5.4.3.12
    Modify System Memory 5.4.3.13
    MSM 5.4.3.13
    NB 5.5.1.3
    NE 5.5.1.5
    Network Buffers 5.5.1.3
    Network Errors 5.5.1.5
    Network Messages 5.5.1.6
    Network Queues 5.5.1.4
    Network Restart 5.5.1.7
    Network Status 5.5.1.1
    Network Traffic 5.5.1.2
    NID 5.4.3.14
    NM 5.5.1.6
    NQ 5.5.1.4
    NR 5.5.1.7

NS 5.5.1.1
NT 5.5.1.2
RCRU 5.4.3.15
Read CRU 5.4.3.15
Release LUNO 5.4.3.16
RL 5.4.3.16
SDT 5.4.3.17
SE 5.5.2.2
Show Date and Time 5.4.3.17
Show Input/Output Status . . . 5.4.3.18
Show Task Status 5.4.3.19
Show Value 5.4.3.20
SIS 5.4.3.18
SM 5.5.2.3
SR 5.5.2.4
SS 5.5.2.1
Station Errors 5.5.2.2
Station Message 5.5.2.3
Station Reset 5.5.2.4, 5.5.3.3
Station Status 5.5.2.1, 5.5.3.2
STS 5.4.3.19

Command Summary, Remote SCI . . . T5-2

Command:
SV 5.4.3.20
TR 5.4.3.21
Trace 5.4.3.21
WCRU 5.4.3.22
Write CRU 5.4.3.22
XT 5.4.3.23

Commands:
Alias 5.2.3.1
Remote SCI 5.4.3

Communications:
Activating 3.5
Hardware Features 1.3.1
Software Features 1.3.2
Utilities 1.4.2.5

Configurations, Network 1.2
CRC Field B.7.5

Data Link Errors A.1.4
Data Packet D.3.5
Data Rate Selection B.5.1

Data:
Read Input 4.2.1.3
Write/Send 4.2.1.2

DB Command 5.4.3.3
Deactivate 1.4.2.5
Utilities 5.6
Debug Command Menu 5.4.2.1
Delete Alias 5.2.3.1
Delete Breakpoint Command . . . 5.4.3.3
Delete Download File 5.2.3.2
Directories and Files, HDLC . . T3-1

Display Error Codes Command 5.4.3.5
Documenting:
    Primary Station 2.3.1
    Secondary Stations 2.3.2
Download File:
    Build 5.2.3.2
    Delete 5.2.3.2
    List 5.2.3.2
Download:
    Messages T5-1
    Preprocessor 1.4.2.5, 5.2
Downloader 1.4.2.5
    Error Recovery 5.3.3.2
    Errors 5.3.3.1
    Messages 5.3.3.1
    Prompts 5.3.3
    Responses 5.3.3
    Utility 5.3.1
Downloading Preprocessed Files 3.5.2
Dump Preprocessed File 5.2.3.2
Dump Workspace Command 5.4.3.4
DW Command 5.4.3.4

End-to-End Control 1.4.1.4
Entry:
    FCCC 3.3.4.2
    Network 3.3.4.2
    Packet Time-Out 3.3.4.2
    Queue Buffer 3.3.4.2
Error Recovery:
    Downloader 5.3.3.2
    Remote SCI 5.4.4
Error Reporting, Remote SCI 5.4.4
Errors:
    Activation Services A.1.6
    Data Link A.1.4
    Downloader 5.3.3.1
    Fatal A.1.5
    Packet-Level A.1.3
    Upper-Level A.1.2
    Warm Start A.1.1
ERRS Command 5.4.3.5
Examples, FORTRAN 4.6.3
Execute Task Command 5.4.3.23

Fatal Errors A.1.5
FCCC Board 1.4.2.1
FCCC Board, Adding 3.3.4.1
FCCC:
    Entry 3.3.4.2
    Installing 3.2.2
    Multiline Operation B.5
Field:
    Address B.7.2

    Command B.7.3
    CRC B.7.5
    Flag B.7.1
    Optional Data B.7.4
Flag Field B.7.1
Format, Transmission B.7
FORTRAN:
    Examples 4.6.3
    Program F4-8, F4-9

Generation:
    Network 3.3
    TX5 . . . . FC-1
Generator, Network 1.4.2.5

Hardware Features, Communications 1.3.1
HDLC:
    Directories and Files T3-1
    Level Functions T1-1
    Levels F1-3
    Line Control 1.4.1.2, 1.4.2.2

ID Association:
    Network/Task 3.3.3.4
    Task/Network 3.3.3.4
ID Command, Associate Network ID/Task 5.4.3.14
IDs:
    Network 2.2.2.5
    Task 2.2.2.4
IDT Command 5.4.3.6
Information Tables, Network 1.4.2.4, 2.2.1
Initialize Date and Time Command 5.4.3.6
Input Data, Read 4.2.1.3
Input/Output Calls 4.2.1
Install Task Command 5.4.3.7
Installing:
    FCCC 3.2.2
    LLM 3.4.3
Interface, Assembly Language F4-7
IT Command 5.4.3.7

Kill Task Command 5.4.3.8
KT Command 5.4.3.8

LB Command 5.4.3.9
LCI 2.2.2.3, D.2
Level Functions, HDLC T1-1
Level:
    Line 1.4.1.1, 1.4.2.1
    Packet 1.4.1.3, 1.4.2.3
    Task D.5
    User Task 1.4.1.5
    Utilities 1.4.1.5
Level 1 . . . . 1.4.1.1, 1.4.2.1

Level 2 . . . . 1.4.1.2, 1.4.2.2
Level 3 . . . . 1.4.1.3, 1.4.2.3
Level 4 . . . . 1.4.1.4, 1.4.2.4
Levels, HDLC F1-3
Line Control, HDLC 1.4.1.2, 1.4.2.2
Line Level 1.4.1.1, 1.4.2.1
Line Reset Command 5.5.3.3
Line Statistics 5.5.3
Line Status Command 5.5.3.1
Lines, Additional 3.3.3.3
Link Stream, TX5 . . . . C.3, FC-2
List Breakpoints Command 5.4.3.9
List Download File 5.2.3.2
List Memory Command 5.4.3.10
List System Memory Command 5.4.3.11
LL Command 5.5.3.1
LLM:
    Board 1.4.2.1
    Installing 3.4.3
LM Command 5.4.3.10
Logical Channel Identifier 2.2.2.3, D.2
Logical Unit Numbers 2.2.2.1
LR Command 5.5.3.3
LS Command 5.5.3.2
LSM Command 5.4.3.11
LUNO Command Menu 5.4.2.3
LUNOs 2.2.2.1

Menu:
    Debug Command 5.4.2.1
    LUNO Command 5.4.2.3
    Miscellaneous Command 5.4.2.4
    Remote SCI 5.4.2
    Task Command 5.4.2.2
Messages:
    Download T5-1
    Downloader 5.3.3.1
    Remote SCI T5-3
Miscellaneous Command Menu 5.4.2.4
MM Command 5.4.3.12
Modify Alias 5.2.3.1
Modify Memory Command 5.4.3.12
Modify Preprocessed File 5.2.3.2
Modify System Memory Command 5.4.3.13
Modifying Network 3.3.4
MSM Command 5.4.3.13
Multiline Operation, FCCC B.5

NB Command 5.5.1.3
NE Command 5.5.1.5
Network:
    Addressing 2.2, 2.2.3, F2-1
    Assignments 2.2.2
Network Buffers Command 5.5.1.3

Network:
    Configurations 1.2
    Entry 3.3.4.2
Network Errors Command 5.5.1.5
Network:
    Generation 3.3
    Generator 1.4.2.5
    IDs 2.2.2.5
    Information Tables 1.4.2.4, 2.2.1
Network Messages Command 5.5.1.6
Network:
    Modifying 3.3.4
    Polling F1-6
Network Queues Command 5.5.1.4
Network Restart Command 5.5.1.7
Network Statistics 5.5.1
Network Status Command 5.5.1.1
Network Traffic Command 5.5.1.2
Network/Task ID Association 3.3.3.4
New Configuration 3.3.3
NID Command 5.4.3.14
NIDs 2.2.2.5
NITs 1.4.2.4, 2.2.1
NM Command 5.5.1.6
Non-TX5 Secondary 3.3.3.1
NQ Command 5.5.1.4
NR Command 5.5.1.7
NS Command 5.5.1.1
NT Command 5.5.1.2

Opcode 2 . . . . 4.2.1.2
Opcode 3 . . . . 4.2.1.3
Opcode 4 . . . . 4.2.2.2
Opcode 5 . . . . 4.2.2.3
Opcode 6 . . . . 4.2.2.4
Opcode 7 . . . . 4.2.2.5
Operation, System 1.4.2
Optional Data Field B.7.4

Packet:
    Data D.3.5
    Level 1.4.1.3, 1.4.2.3
    RC 1.4.2.3
    Receive Not Ready 1.4.2.3, D.3.3
    Receive Ready 1.4.2.3, D.3.4
    RESET 1.4.2.3
    Reset Confirmation 1.4.2.3, D.3.2
    Reset Request 1.4.2.3, D.3.1
    RNR 1.4.2.3
    RR 1.4.2.3
Packet Time-Out Entry 3.3.4.2
Packet-Level Errors A.1.3
Patch Preprocessed File 5.2.3.2
Planning 2.2.2

Polling B.1, B.2
    Network F1-6
Preprocessed File:
    Dump 5.2.3.2
    Modify 5.2.3.2
    Patch 5.2.3.2
Preprocessed Files, Downloading 3.5.2
Preprocessor 5.2.2
    Download 1.4.2.5, 5.2
    Prompts 5.2.3
    Responses 5.2.3
Primary Station:
    Activating 3.5.1
    Documenting 2.3.1
Primary System, Building 3.2.1
Primary Task, Adding 3.3.4.1
Program:
    Assembly Language F4-5, F4-6
    FORTRAN F4-8, F4-9
Prompts:
    Downloader 5.3.3
    Preprocessor 5.2.3

Queue Buffer Entry 3.3.4.2

RC Packet 1.4.2.3
RCRU Command 5.4.3.15
Read CRU Command 5.4.3.15
Read Input Data 4.2.1.3
READ4D Call 4.6.2.1
Receive Not Ready Packet 1.4.2.3, D.3.3
Receive Ready Packet 1.4.2.3, D.3.4
Release LUNO Command 5.4.3.16
Remote SCI 1.4.2.5
Remote SCI:
    Activation 5.4.1
    Building 3.4.2
    Command Summary T5-2
    Commands 5.4.3
    Error Recovery 5.4.4
    Error Reporting 5.4.4
    Menu 5.4.2
    Messages T5-3
    Termination 5.4.1
Removal, Request 4.2.2.5
Request:
    Activation 4.2.2.2, 4.2.2.3, 4.2.2.4
    Removal 4.2.2.5
Reset Confirmation Packet 1.4.2.3, D.3.2
RESET Packet 1.4.2.3
Reset Request Packet 1.4.2.3, D.3.1
Responses:
    Downloader 5.3.3
    Preprocessor 5.2.3

RL Command . . . 5.4.3.16
RNR Packet . . . 1.4.2.3
RR Packet . . . 1.4.2.3

SCI, Remote . . . 1.4.2.5
SDT Command . . . 5.4.3.17
SE Command . . . 5.5.2.2
Secondary, Non-TX5 . . . 3.3.3.1
Secondary Station, Adding . . . 3.3.4.1
Secondary Stations, Documenting . . . 2.3.2
Secondary System, Building . . . 3.4.1
Secondary Task, Adding . . . 3.3.4.1
Secondary, TX5 . . . 3.3.3.2, 3.4
Selection:
    Data Rate . . . B.5.1
    Time Delay . . . B.3
Services, Activation . . . 4.2.2
Show Date and Time Command . . . 5.4.3.17
Show Input/Output Status Command . . . 5.4.3.18
Show Task Status Command . . . 5.4.3.19
Show Value Command . . . 5.4.3.20
SIS Command . . . 5.4.3.18
SM Command . . . 5.5.2.3
Software Features, Communications . . . 1.3.2
SR Command . . . 5.5.2.4
SS Command . . . 5.5.2.1
Station Addresses . . . 2.2.2.2
Station Documentation . . . F2-2, F2-3
Station Errors Command . . . 5.5.2.2
Station Message Command . . . 5.5.2.3
Station Reset Command . . . 5.5.2.4, 5.5.3.3
Station Statistics . . . 5.5.2
Station Status Command . . . 5.5.2.1, 5.5.3.2
Stations, Additional . . . 3.3.3.3
Statistics:
    Accumulation . . . 1.4.2.5
    Line . . . 5.5.3
    Network . . . 5.5.1
    Station . . . 5.5.2
STS Command . . . 5.4.3.19
Subroutine:
    Assembly Language . . . 4.6.1
    Calls . . . 4.6.2
Supervisor Call Block . . . F4-1, F4-2
Supervisor Calls . . . 4.2
SV Command . . . 5.4.3.20
System Generation, TX5 . . . 3.4.5
System Operation . . . 1.4.1, 1.4.2
Systen Generation, TX5 . . . C.2

Task Command Menu . . . 5.4.2.2
Task:
    IDs . . . 2.2.2.4
    Level . . . D.5

Task/Network ID Association 3.3.3.4
TAT B.4.1
Termination 3.3.2
    Remote SCI 5.4.1
Time Delay Selection B.3
TR Command 5.4.3.21
Trace Command 5.4.3.21
Transmission Format B.7
Turnaround Time B.4.1
TX5:
    Assembling 3.4.4
    Generation FC-1
    Link Stream C.3, FC-2
    Secondary 3.3.3.2, 3.4
    System Generation 3.4.5
    Systen Generation C.2

Upper-Level Errors A.1.2
User Interface 1.4.2.4
User Task Level 1.4.1.5
Utilities:
    Activate 5.6
    Communications 1.4.2.5
    Deactivate 5.6
    Level 1.4.1.5
Utility, Downloader 5.3.1

Warm Start Errors A.1.1
WCRU Command 5.4.3.22
WKUP4D Call 4.6.2.3
Write CRU Command 5.4.3.22
Write/Send Data 4.2.1.2
WRIT4D Call 4.6.2.2

XT Command 5.4.3.23

USER'S RESPONSE SHEET

Manual Title: Model 990 Computer DX10 HDLC Communications Package User's Guide (2270526-9701)

Manual Date: 1 October 1981 **Date of This Letter:** ______

User's Name: ______ **Telephone:** ______

Company: ______ **Office/Department:** ______

Street Address: ______

City/State/Zip Code: ______

Please list any discrepancy found in this manual by page, paragraph, figure, or table number in the following space. If there are any other suggestions that you wish to make, feel free to include them. Thank you.

Location in Manual	Comment/Suggestion

NO POSTAGE NECESSARY IF MAILED IN U.S.A.
FOLD ON TWO LINES (LOCATED ON REVERSE SIDE), TAPE AND MAIL

CUT ALONG LINE

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 7284 DALLAS, TX

POSTAGE WILL BE PAID BY ADDRESSEE

TEXAS INSTRUMENTS INCORPORATED
DIGITAL SYSTEMS GROUP

ATTN: TECHNICAL PUBLICATIONS
P.O. Box 2909 M/S 2146
Austin, Texas 78769

FOLD

Texas Instruments U.S. District Sales and Service Offices

(A complete listing of U.S. offices is available from the
district office nearest your location)

California
831 S. Douglas Street
El Segundo, California 90245
(213) 973-2571

100 California Street
Suite 480
San Francisco, California 94111
(415) 781-9470

776 Palomar Avenue
P.O. Box 9064
Sunnyvale, California 94086
(408) 732-1840*

3186 Airway
Suite J
Costa Mesa, California 92626
(714) 540-7311

Colorado
9725 East Hampden Avenue
Suite 301
Denver, Colorado 80231
(303) 751-1780

Florida
1850 Lee Road
Suite 115
Winter Park, Florida 32789
(305) 644-3535

Georgia
3300 Northeast Expressway
Building 9
Atlanta, Georgia 30341
(404) 458-7791

*Service telephone number

Illinois
515 West Algonquin Road
Arlington Heights, Illinois 60005
(312) 640-2900
(800) 942-0609*

Massachusetts
504 Totten Pond Road
Waltham, Massachusetts 02154
(617) 890-7400

Michigan
24293 Telegraph Road
Southfield, Michigan 48034
(313) 353-0830
(800) 572-8740*

Minnesota
7625 Parklawn Avenue
Minneapolis, Minnesota 55435
(612) 830-1600

Missouri
2368 Schuetz
St. Louis, Missouri 63141
(314) 569-0801*

New Jersey
1245 Westfield Avenue
Clark, New Jersey 07066
(201) 574-9800

Ohio
4124 Linden Avenue
Dayton, Ohio 45432
(513) 258-3877

Pennsylvania
420 Rouser Road
Coraopolis, Pennsylvania 15108
(412) 771-8550

Texas
8001 Stemmons Expressway
P.O. Box 226080
M/S 3108
Dallas, Texas 75266
(214) 689-4460

13510 North Central Expressway
P.O. Box 225214
M/S 393
Dallas, Texas 75265
(214) 238-3881

9000 Southwest Freeway, Suite 400
Houston, Texas 77074
(713) 776-6577

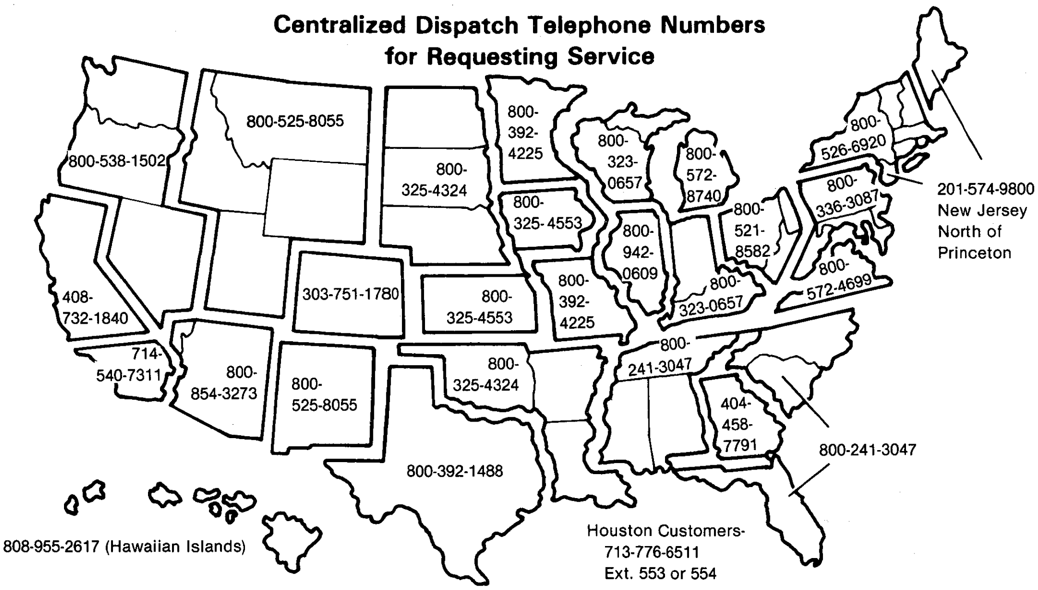
8585 Commerce Drive, Suite 518
Houston, Texas 77036
(713) 776-6531
(713) 776-6553*

Virginia
1745 Jefferson Davis Highway
Crystal Square 4, Suite 600
Arlington, Virginia 22202
(703) 553-2200

Wisconsin
205 Bishops Way
Suite 214
Brookfield, Wisconsin 53005
(414) 784-1323

TI-CARE*

Centralized Dispatch Telephone Numbers for Requesting Service



808-955-2617 (Hawaiian Islands)

Houston Customers-
713-776-6511
Ext. 553 or 554

Installation for Computer Systems
800-231-2807
713-937-1200 (Texas only, collect)

Dallas Customers-
214-238-3881

*Service mark of Texas Instruments

The **TI Customer Support Line** is available to answer our customers' complex technical questions. The extensive experience of a selected group of TI senior engineers and systems analysts is made available directly to our customers. The **TI Customer Support Line** telephone number is **(512) 250-7407.**

TEXAS INSTRUMENTS
INCORPORATED
DIGITAL SYSTEMS GROUP
POST OFFICE BOX 2909 AUSTIN, TEXAS