MUXPORT Protocol
The goal of this seminar is to provide advanced training for HDLC like protocols in general and Codex Muxport Protocol specifically. The seminar will focus on protocol definition and examination not on strapping or programming.

Upon completion of this seminar the student should be familiar with the Codex Muxport protocol and be able to do basic Muxport troubleshooting and problem determination/recognition.

The accompanying document may be used as a reference for the seminar or as a stand alone tutorial/reference on HDLC/Muxport protocol interpretation.

The following prerequisites must be met to take this seminar:

- 6000/6050 CSE Course
- 6760/6740 Launch Training
- RTS or Senior CSE

The following topics will be covered in the Muxport/BOP seminar.

1. ISO REFERENCE MODEL
2. HDLC/SDLC
3. MUXPORT PROTOCOL
4. 6740 MUXPORT/NP PROTOCOL
5. 6760 EXTENDED MUXPORT OVERVIEW
6. X.25 OVERVIEW
MUXPORT SEMINAR OUTLINE

I. INTRODUCTION
   A. Objectives
   B. Why
   C. What's Covered
   D. Training Participation

II. ISO REFERENCE MODEL
   A. Definition Of ISO
   B. Layer Concept
   C. Layer Definition
   D. Data Flow Example

III. BIT ORIENTED PROTOCOL
   A. Introduction
      1. What is BOP
      2. Comparison with BSC
      3. HDLC SDLC comparison
   B. Station Configurations
      1. Types
      2. Data links
      3. Modes
   C. Data Link Controls
      1. Zero insertion
      2. Window concepts
      3. Frame numbering
   D. Frame Structure
      1. Definition of frame components
      2. Explanation of each field
   E. Information Frame
   F. Supervisory Frame
   G. Unnumbered Frame
IV. MUXPORT PROTOCOL

A. Overview
1. What is Codex Muxport Protocol
2. Definition of CMP layers
3. Comparison with other protocols
4. Frame structure

B. Line Layer
1. Function
2. Idle sequences

C. ARQ Layer
1. Functions
2. Addressing
3. Frame types supported
4. Remote reset sequence
5. System states

D. MUX Layer
1. Functions
2. Control status byte
3. Control slots
4. Data slots
5. Supervisory slots

E. Connection Layer
1. Functions
2. Escape sequence
3. CSU
4. DPI
5. Break
6. Data termination
7. Autospeed
8. Flow control

F. Data Scope Sequences
1. Idle
2. Boot
3. ISCC sequences
V. JUPITER PROTOCOL

A. Introduction
   1. What is Jupiter Protocol
   2. NPP overview and data flow

B. Paths
   1. Path definition
   2. Numbering
   3. Routing
   4. Enabling
   5. Rerouting

C. Line & ARQ Layers
   1. Line layer functions
   2. ARQ layer functions
      a. Address field
      b. Control field

D. MUX Layer
   1. Functions
   2. Command Codes
   3. Address Packet
   4. Protocol Announcement

E. Connection Layer
   1. Functions
   2. Supervisory functions performed

VI. 6760 EXTENDED MUXPORT

A. Overview
B. Performance
C. Compatibility
VII. X.25 OVERVIEW

A. Definition of X.25
   1. Packet switching concepts
   2. Virtual circuits

B. X.25 Structure
   1. Definition of layers
   2. Physical
   3. Frame
   4. Packet

C. Packet Layer Structure
   1. Format
   2. GFI
   3. LCN / LCGN
   4. Packet types

D. Calling Sequences
   1. Call establishment
   2. Call Clearing
   3. Data transfer

E. PAD Overview
   1. Definition of PAD
   2. Definition of protocols used
MUXPORT SEMINAR

NATIONAL TECHNICAL SUPPORT

STUDENT HANDOUT

Prepared by: Bart M. Zaino
LAYERS OF MODEL

APPLICATION
PRESENTATION
SESSION
TRANSPORT
NETWORK
DATA LINK
PHYSICAL

USERS OF TRANSPORT SERVICE

NETWORK SERVICE
LAYERS OF OSI REFERENCE MODEL

APPLICATION LAYER - Directly serves the end-user, which is the application process (AP), by providing the distributed information service to support the AP and manage the communication.

PRESENTATION LAYER - Provides the services to allow the AP to interpret the meaning of the information exchanged. Translation and formatting of information is performed at this layer.

SESSION LAYER - Supports the dialog between cooperating APs binding and unbinding them into a communicating relationship.

TRANSPORT LAYER - Provides end-to-end control and information interchange with the level of reliability that is needed for the application. The services provided to the upper layers are independent of the underlying network implementation.

NETWORK LAYER - Provides the means to establish, maintain, and terminate the switched connections between end-systems. Included are addressing and routing functions. An additional global sublayer may also be provided to ensure a consistent quality of service on connection traversing more than one network. The interface between this layer and the transport layer provides services that are independent of the underlying media.

DATA LINK LAYER - Provides the synchronization and error control for the information transmitted over the physical link.

PHYSICAL LAYER - Provides the electrical, mechanical, functional, and procedural characteristics to activate, maintain, and deactivate the physical connection.
LAYER CONCEPT

- Each layer requests services from the layer below and provides services to the layer above.
- Each layer is built on the capabilities of the layers below it.
- Layers appear to communicate only with their peer layer.

SERVICES

\[ \text{SYSTEM A: LAYER X:} \quad \text{SYSTEM B: LAYER X} \]

\[ \text{PEER PROTOCOL} \]

\[ \text{SERVICES} \]

\[ \text{(Requirements of the layer)} \]

\[ \text{(n - 1) layer} \]

\[ \text{SERVICES} \]

\[ \text{(n + 1) layer} \]

\[ \text{(Functions Provided)} \]
One module for each user session

One module for each user session

SESSION SERVICES SUBSYSTEM

TRANSPORT SUBSYSTEM

COMMON NETWORK

NETWORK CONTROL

PACKETS

One module for the entire machine

One module for the entire machine

One module for each communications circuit

One module for each communications circuit
a. Unbalanced Configuration

b. Balanced Configuration
ZERO INSERTION (BIT STUFFING)

Bit stream, for CRC computation
Transmitted bit stream

01111110 11111111 11000000 11111100 11111101

0111111011111101110000000111110100111110101

Inserted zeros

Deleted zeros

Received bit stream

Bit stream, for CRC computation

01111110 11111111 11000000 11111100 11111101

011111101111110111111000000111110100111110101

Transmit in parallel form

Serialize
Compute FCS
Generate Fs
Insert a zero, after 5 ones between Fs.

Recognize Fs
Delete a zero, after 5 ones between Fs.
Compute and check FCS.

Deserialize

Receive in parallel form
WINDOW CONTROL

WINDOW SIZE = 7

OUTSTANDING

CAN'T BE SENT BEFORE ACKNOWLEDGMENT OF FRAME 5

LAST SENT

Rec. Pg. 2-10
## FRAME STRUCTURE

<table>
<thead>
<tr>
<th>F</th>
<th>A</th>
<th>C</th>
<th>I</th>
<th>FCS</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPENING</td>
<td>ADDRESS</td>
<td>CONTROL</td>
<td>INFORMATION</td>
<td>FRAME CHECK</td>
<td>CLOSING</td>
</tr>
<tr>
<td>FLAG</td>
<td>8 BITS</td>
<td>8 BITS</td>
<td>VARIABLE</td>
<td>SEQUENCE</td>
<td>FLAG</td>
</tr>
<tr>
<td>8 BITS</td>
<td>8 BITS</td>
<td>8 BITS</td>
<td>VARIABLE</td>
<td>16 BITS</td>
<td>8 BITS</td>
</tr>
<tr>
<td>SPAN OF</td>
<td>FRAME CHECK</td>
<td>ZERO INSERTION</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FRAME**
NON EXTENDED

<table>
<thead>
<tr>
<th>FRAME TYPE</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-FRAME</td>
<td>Nr</td>
<td>P/F</td>
<td>Ns</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-FRAMES</td>
<td>Nr</td>
<td>P/F</td>
<td>S</td>
<td>S</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U-FRAMES</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>P/F</td>
<td>M</td>
<td>M</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

EXTENDED

<table>
<thead>
<tr>
<th>TYPE</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-FRAME</td>
<td>Nr</td>
<td>P/F</td>
<td>Ns</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-FRAME</td>
<td>Nr</td>
<td>P/F</td>
<td>X</td>
<td>X</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U-FRAME</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>P/F</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>X</td>
<td>M</td>
<td>M</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

LSB
Information "I" Format

Open Flag Field | Address | Control Field | Message | FCS | Closing Flag Field

<table>
<thead>
<tr>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receive Sequence Number (N_r)</td>
<td>Poll/Final Bit</td>
<td>Send Sequence Number (N_s)</td>
<td>Frame Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Supervisory "S" Format

Opening Flag Field | Address | Control Field | FCS | Closing Flag Field

8 7 6 5 4 3 2 1
Receive Sequence Number (N_r) | Poll/Final Bit | Command/Response | Frame Type
<table>
<thead>
<tr>
<th>Command/Response</th>
<th>8 7 6 5 4 3 2 1</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receiver Ready</td>
<td>Nr P/F 0 0 0 1</td>
<td>RR</td>
</tr>
<tr>
<td>Receiver Not Ready</td>
<td>Nr P/F 0 1 0 1</td>
<td>RNR</td>
</tr>
<tr>
<td>Reject</td>
<td>Nr P/F 1 0 0 1</td>
<td>REJ</td>
</tr>
<tr>
<td>Selective Reject</td>
<td>Nr P/F 1 1 0 1</td>
<td>SREJ</td>
</tr>
</tbody>
</table>
Unnumbered "U" Format

Opening Flag Field | Address | Control Field | FCS | Closing Flag Field

<table>
<thead>
<tr>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command/Response</td>
<td>Poll/Final Bit</td>
<td>Command/Response</td>
<td>Frame Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### UNNUMBERED FORMAT (U-FRAME)

<table>
<thead>
<tr>
<th>Command/Response</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set Normal Resp. Mode</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td>P</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>SARM</td>
</tr>
<tr>
<td>Set Normal Resp. Mode Ext.</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td>P</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>SNRME</td>
</tr>
<tr>
<td>Set Async Resp. Mode</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>P</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>SARM</td>
</tr>
<tr>
<td>Set Async Resp. Mode Ext.</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td>P</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>SARME</td>
</tr>
<tr>
<td>Set Async Balanced Mode</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td>P</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>SABM</td>
</tr>
<tr>
<td>Set Async Bal. Mode Ext.</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td>P</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>SABME</td>
</tr>
<tr>
<td>Disconnect</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td>P</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>DISC</td>
</tr>
<tr>
<td>Set Initialize Mode</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>P</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>SIM</td>
</tr>
<tr>
<td>Exchange Station ID's</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td>P/F</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>XID</td>
</tr>
<tr>
<td>Unnumbered Acknowledge</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td>F</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>UA</td>
</tr>
<tr>
<td>Disconnect Mode</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>F</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>DM</td>
</tr>
<tr>
<td>Request Disconnect</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td>F</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>RD</td>
</tr>
<tr>
<td>Request Initialize Mode</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>F</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>RIM</td>
</tr>
<tr>
<td>Command (Frame) Reject</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td>F</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>CMDR</td>
</tr>
<tr>
<td>Unnumbered Poll</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td>P</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>UP</td>
</tr>
<tr>
<td>Unnumbered Information</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>P/F</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>UI</td>
</tr>
<tr>
<td>Layer</td>
<td>ISO / X.25</td>
<td>SNA</td>
<td>DECNET</td>
<td>MUXPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>-----</td>
<td>--------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>APPLICATION</td>
<td>END USER</td>
<td>APPLICATION</td>
<td>NOT USED BY MUXPORT PROTOCOL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>PRESENTATION</td>
<td>NAU SERVICES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>SESSION</td>
<td>DATA FLOW TRANSMISSION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TRANSPORT</td>
<td>PATH CONTROL</td>
<td>NETWORK SERVICES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>NETWORK/PACKET</td>
<td>TRANSPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>DATA LINK/HDL</td>
<td>SDLC</td>
<td>DDCMP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>PHYSICAL</td>
<td>PHYSICAL</td>
<td>PHYSICAL</td>
<td>PHYSICAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MUXPORT PROTOCOL SUMMARY

- GENERAL
  - Four layer protocol
  - X.25 level 2 & HDLC line layer compatible
  - Balanced mode operation

- LINE LAYER
  - FCS calculation
  - Zero insertion
  - Idle fill
  - Abort & flag generation/detection

- ARQ LAYER
  - Address & control field processing
  - Error recovery
  - State machine/timer maintenance

- MUX LAYER
  - Compiles data & control information
  - Multiplex data
  - VCTP operation
  - Formats data, control & supervisory slots

- CONNECTION LAYER
  - Transfer data to/from customer equipment
  - Perform ISCC functions
  - Manage flow control
<table>
<thead>
<tr>
<th>Flag</th>
<th>PCS</th>
<th>Data Slot</th>
<th>MUX layer control slot</th>
<th>ARQ layer control field</th>
<th>Addr. Field</th>
<th>Flag</th>
</tr>
</thead>
</table>

- **Connection Layer**
- **Multiplex Layer**
- **ARQ Layer**
- **Line Layer**

**Protocol Layers**
Line Layer Functions

- Performs FCS calculation.
- Maintains data transparency through zero insertion technique.
- Performs frame pacing and idle line fill.
- Idle fill is the transmission of flags (7E).
- Frame pacing is the transmission of 'empty' frames to keep the muxports in sync when no traffic is present.
Idle Sequences By Product

6005 - Sends RR's when connected to 6050, 6040, 6760. Connected to a 6740 it sends a RR then an empty I-Frame.

6050 - Sends empty data frames (I-Frames).

6040 - Same as the 6050.

6740 - Sends a RR then an empty I-Frame.

6760 - Same as the 6050.
MUXPORT ARQ LAYER FUNCTIONS

- Addressing of each station using X.25 LAPB notation.
- Frame definition (control field interpretation).
MUXPORT FRAME ADDRESSING

- Primary is addressed as 'B' = 0000001 = 01 = DCE.
- Secondary is addressed as 'A' = 00000011 = 03 = DTE.
- Commands contain the address of the receiving unit.
- Responses contain the address of the transmitting unit.
- The exception is the 6050 who always sends address 'A'.

<table>
<thead>
<tr>
<th>FRAME</th>
<th>8 7 6</th>
<th>5</th>
<th>4 3 2 1</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-FRAME</td>
<td>Nr</td>
<td>P/F</td>
<td>Ns</td>
<td>0</td>
</tr>
<tr>
<td>S-FRAME</td>
<td>Nr</td>
<td>P/F</td>
<td>0 0 0 1</td>
<td>RR (Receiver ready)</td>
</tr>
<tr>
<td></td>
<td>Nr</td>
<td>P/F</td>
<td>0 1 0 1</td>
<td>RNR (Receiver not ready)</td>
</tr>
<tr>
<td></td>
<td>Nr</td>
<td>P/F</td>
<td>1 0 0 1</td>
<td>REJ (Reject)</td>
</tr>
<tr>
<td>U-FRAME</td>
<td>0 0 1</td>
<td>P</td>
<td>1 1 1 1</td>
<td>SABM async balanced mode</td>
</tr>
<tr>
<td></td>
<td>0 1 1</td>
<td>P</td>
<td>1 1 1 1</td>
<td>SABME (SABM - extended)</td>
</tr>
<tr>
<td></td>
<td>0 1 1</td>
<td>F</td>
<td>0 0 1 1</td>
<td>UA (unnumbered Ack.)</td>
</tr>
<tr>
<td></td>
<td>1 0 0</td>
<td>F</td>
<td>0 1 1 1</td>
<td>FRMR (Frame reject)</td>
</tr>
</tbody>
</table>
# FRAME REJECT COMMAND

## Normal Mode

<table>
<thead>
<tr>
<th>Byte #1</th>
<th>1 0 0 F 0 1 1 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte #2</td>
<td>Rejected Control field</td>
</tr>
<tr>
<td>Byte #3</td>
<td>Vr C/R Vs</td>
</tr>
<tr>
<td>Byte #4</td>
<td>0 0 0 0 Z Y X W</td>
</tr>
</tbody>
</table>

Where:

- **Byte #1** = The Frame reject command control field.
- **Byte #2** = The rejected control field.
- **Byte #3** = The current value of the XMT state variable.
- **Vr** = The current state of the RCV state variable.
- **C/R** = Indicates if the frame rejected was a command (0) or a response (1).
- **W** = If = to "1" indicates the control field is invalid.
- **Byte #4** = If = to "1" indicates the control field received is invalid because a data field was in an S or U-Frame.
- **Y** = If set to "1" indicates that the I-field received exceeded the maximum allowable limit.
- **Z** = If set to "1", indicates the control field received contained an invalid Nr count.

## Extended Mode

<table>
<thead>
<tr>
<th>Byte #1</th>
<th>1 0 0 X 0 1 1 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte #2</td>
<td>0 0 0 0 0 0 0 F</td>
</tr>
<tr>
<td>Byte #3</td>
<td>First byte of rejected control field</td>
</tr>
<tr>
<td>Byte #4</td>
<td>Second byte of rejected control field</td>
</tr>
<tr>
<td>Byte #5</td>
<td>Vr 0</td>
</tr>
<tr>
<td>Byte #6</td>
<td>Vs C/R</td>
</tr>
<tr>
<td>Byte #7</td>
<td>0 0 0 0 Z Y X W</td>
</tr>
</tbody>
</table>
Any number of bytes

<table>
<thead>
<tr>
<th>Command Field</th>
<th>Unit B</th>
<th>Unit A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Go to Loopback</td>
<td>X'B8'</td>
<td>X'B8'</td>
</tr>
<tr>
<td>I am going in Loopback</td>
<td>X'45'</td>
<td>X'45'</td>
</tr>
<tr>
<td>Go to Normal</td>
<td>X'44'</td>
<td>X'44'</td>
</tr>
</tbody>
</table>
MUXPORT STATE MACHINE TIMERS

- **T1 TIMER** - Used in Link Set-Up to check for its completion. If Link Set-up is not complete when T1 expires the procedure (SABM or SABME) is restarted. T1 is a 3 second timer.

- **T2 TIMER** - Used in the Information Transfer state as a framing timer. When a frame is transmitted T2 starts and is not stopped until an ack. for the frame is received. If T2 expires the link set up procedures (SABME) start. T2 = 20 sec.

- **N2 COUNTER** - Used in link set up. The maximum number of SABM(E)'s sent before a remote reset is sent. N2 = 20 tries.

- **N1 COUNTER** - The maximum number of bits allowed in a frame. If exceeded the frame is rejected. N1 = 16,0000 bits/frame.

- **RC COUNTER** - The retransmission counter, or the number of times to retransmit a frame. RC = 20 retransmits.
LINK SET UP STATE and
EXCEPTION TRANSITIONS FROM INFORMATION TRANSFER STATE
ARQ Operation
A to B Information Transfer
MUX LAYER FUNCTIONS

- The actual multiplexing of data occurs here.
- Compile the information field of Muxport I-Frames.
- Responsible for the format of data and supervisory information, provided by the connection layer.
- Control information is added by this layer.
MUXPORT CONTROL SLOT FORMAT

+---------------------------------+------------------------+------------------------+
<table>
<thead>
<tr>
<th>FLAG</th>
<th>A</th>
<th>C</th>
<th>80 AA BB XXXXXXXXXX</th>
<th>Rest of Frame</th>
<th>FCS</th>
<th>FLAG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Control slot</td>
<td>Command Dep. Par.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status Byte</td>
<td>Command Code</td>
<td>Byte Count</td>
<td>Command Dep. Par.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Where:

80 = Control Slot Status Byte (MSB set) indicates control information follows.
AA = Byte Count, the number of control information bytes to follow.
BB = Command Code indicates one of the following commands:
CONTROL SLOT COMMAND CODES

0  Error Message Report
1  Read Port Configuration Parameters
2  Return Port Configuration Parameters
3  Write Port Configuration Parameters
4  Write Acknowledge Port Configuration Parameters
5  Read Port Statistics
6  Return Port Statistics
7  Read Statistics Threshold
8  Return Statistics Threshold
9  Write Statistics Threshold
A  Write Acknowledge Statistics Threshold
20  CTP Command
21  CTP Response (continuing)
22  CTP Response (completed)
23  Unsolicited Error Messages and Reports
24  End of VCTP
25  Address Packets
27  6740 Protocol Announcement
29  6005-6005 Move TP Configuration Command
### MUXPORT DATA SLOT HEADER FORMAT

#### NORMAL

<table>
<thead>
<tr>
<th>XXXXX</th>
<th>XXX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LSB</td>
</tr>
</tbody>
</table>

- **Thread Number**: __
- **LSB**: ___
- **Number of bytes to follow**: 1 - 7
- **Max address of 31**: $(2^4, 2^3, 2^2, 2^1, 2^0)$

#### EXTENDED/LONG

<table>
<thead>
<tr>
<th>XXXXX</th>
<th>000</th>
<th>XXXXXXXX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Thread Number**: ___
- **Bytes to follow**: 8 - 255
- **Indicates**: Look at next byte
- **Number of bytes to follow**: ___

---

Ref A 3-5

32
## MUXPORT DATA SLOT FORMAT

<table>
<thead>
<tr>
<th>Flag</th>
<th>A</th>
<th>C</th>
<th>I - Field</th>
<th>FCS</th>
<th>Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>7E</td>
<td>03</td>
<td>0Ns Nr: 00</td>
<td>77 41 42 43 44 45 46 47</td>
<td>xx xx</td>
<td>7E</td>
</tr>
</tbody>
</table>

"A" I-Frame

Control Status Byte

Thread 14, 7 bytes to follow 77 = 01110 111

---

Ref: PG 3-15
SUPERVISORY SLOT FORMAT

- Passes end-to-end flow control information for ports.
- Interleaved with data slots.

<table>
<thead>
<tr>
<th>FLAG</th>
<th>A</th>
<th>C</th>
<th>I FIELD</th>
<th>FCS</th>
<th>FLAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>7E</td>
<td>01</td>
<td>0Ns Nr</td>
<td>00 08 02 XX XX</td>
<td>FCS</td>
<td>7E</td>
</tr>
</tbody>
</table>

Control Status Byte_ (DATA) _Command Dependent Supervisory Function_

Slot Header (00001 000)_ 2 bytes to follow supervisory slot
Thread #1, long data slot slot (Long format less than 8 bytes)
See next byte for length
CONNECTION LAYER FUNCTIONS

- Transfers data between the customer and codex equipment.
- Transfers all data is transferred as eight bit bytes.
- Performs the following In-stream control functions:
  a. Control signal updates
  b. Data path initialization
  c. Autospeed
  d. Break
  e. Data slot termination
  f. Transmission of a literal X'01'
  g. Port flow control
CONTROL SIGNAL UPDATES

- Used to pass the value of terminal or modem signals.
- Updates for DTR, RTS and MB are passed.
- Each CSU begins with the sequence (X'01') followed by a second byte identifying the signal(s) being updated.
- A bit set indicates the signal is 'high' a bit not set (zero) indicates a signal is 'low'.

CSU = 01 ln

Where n = 0 MB RTS DTR

| 01 10 | All signals low |
| 01 11 | DTR high        |
| 01 12 | RTS high        |
| 01 13 | RTS & DTR high  |
| 01 14 | MB high         |
| 01 15 | DTR & MB high   |
| 01 16 | RTS & MB high   |
| 01 17 | Not used for CSU's |
| 01 40 | CSU request to remote |
DATA PATH INITIALIZATION

- Prevents user data from becoming trapped after an abnormal disconnect sequence. Flushes the port buffers.

- Two functions are used: Data Path Initialization (DPI) and Data Path Initialization Acknowledge (DPIA).

- When a high to low transition of DTR is detected, a DPI (X'01 20') is sent to the remote unit following a CSU.

- The remote unit responds with a DPIA (X'01 30').
Data Path Initialization

Events

1) TP at Unit A senses drop of DTR
2) Unit A sends a CSU to signal DTR
3) Unit A send DPI
4) Unit B responds with DPIA
BREAK

- A break is a constant space condition.
- Two ISCC functions are used X'7n' and X'80' for break.
- The X'01 7n' is used to initiate the break. The time period (length) of the break is "n" character times.
- If n = 0 the break is continuous.
- The X'01 80' is used to stop the continuous break.
DATA SLOT TERMINATION

- Added to the end of each of each data slot according to the protocol used.
- Signals the end of the ports allocated space in the Muxport frame. It may be continued in another frame.

<table>
<thead>
<tr>
<th>Slot Type</th>
<th>Termination sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asynchronous</td>
<td>No termination required</td>
</tr>
<tr>
<td>Synchronous</td>
<td>X'01 17'</td>
</tr>
<tr>
<td>BOP (flag idle)</td>
<td>X'01 72'</td>
</tr>
<tr>
<td>BOP (mark idle)</td>
<td>X'01 72 01 17'</td>
</tr>
<tr>
<td>BOP (abort)</td>
<td>X'01 17'</td>
</tr>
</tbody>
</table>
AUTOSPEED

| FLAG | A | C | 00 08 08 01 A6 01 A1 01 A0 01 A2 | FCS | FLAG |

Status Byte
Port 1, 8 bytes to follow
Lo nibble speed code

Hi nibble ARC
Lo nibble ARC
Hi nibble speed code

SPEED CODES

<table>
<thead>
<tr>
<th>SPEED</th>
<th>CODE</th>
<th>SPEED</th>
<th>CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>01</td>
<td>1800</td>
<td>14</td>
</tr>
<tr>
<td>75</td>
<td>06</td>
<td>2000</td>
<td>15</td>
</tr>
<tr>
<td>110</td>
<td>08</td>
<td>2400</td>
<td>16</td>
</tr>
<tr>
<td>134.5</td>
<td>0A</td>
<td>3600</td>
<td>17</td>
</tr>
<tr>
<td>150</td>
<td>0B</td>
<td>4800</td>
<td>18</td>
</tr>
<tr>
<td>300</td>
<td>0E</td>
<td>7200</td>
<td>1A</td>
</tr>
<tr>
<td>600</td>
<td>10</td>
<td>9600</td>
<td>1C</td>
</tr>
<tr>
<td>1200</td>
<td>13</td>
<td>19200</td>
<td>1E</td>
</tr>
</tbody>
</table>

Ref. A 3-A
FLOW CONTROL

- Does not use an X'01' escape sequence.
- Transmitted in supervisory slots.
- Used to regulate the flow of data between ports.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'10'</td>
<td>FCI - (Flow Control Initialization) sent by the controlling port to initiate flow control.</td>
</tr>
<tr>
<td>X'11'</td>
<td>FCIA - (Flow Control Initialization Acknowledge) Affirmation sent by port to be controlled.</td>
</tr>
<tr>
<td>X'2n'</td>
<td>FCA - (Flow Control Authorization) Sent to controlled port (n+1) *16 = number of bytes authorized for transmission.</td>
</tr>
<tr>
<td>X'12'</td>
<td>FCIR - (Flow Control Initialization Request) Sent by controlled port asking to send more data.</td>
</tr>
<tr>
<td>X'01 40'</td>
<td>FCD - (Flow Control Disable) Signal to controlled port that data flow is not controlled.</td>
</tr>
</tbody>
</table>
Flow Control - No lost FCAs

Events

1) Unit A sends FCI
2) Unit B responds with FCIA
3) Unit A responds with FCA
4) Unit B sends text
5) Unit A outputs text
6) Unit A sends FCA
7) Unit B sends text
JUPITER PROTOCOL OVERVIEW

- An enhanced version of Codex Muxport Protocol.
- Requires a 6740 Network port or Muxport.
- Runs in extended mode only.
- Enhancements over CMP for 6740 delta networks.
  1. Multiple 'paths' for data transfer.
  2. Frame routing at the ARQ level.
  3. Flow control on a per path basis.
NPP OVERVIEW

- Network Port Processor.
- Used to transfer data between the master processor and the configured links.
- Up to 4 links are available per NPP.
- NPP's can run a combination of Jupiter or CMP.
- Link connection establishment.
- Thread and link routing and rerouting.
- Act as an endpoint for Muxport connections.
- Provide 'special' Muxport 'EXTRA FUNCTIONS'

  1. Loopback.
  2. Data monitoring.
  3. FOX diagnostics.
  4. UDR (Hunt Group)
NPP DATA FLOW - RECEIVE

- A valid HDLC address is received.
- If the frame is destined for another node, the frame is held in a buffer until it is complete then sent to its final destination.
- Local frames are scanned by the NPP for frame type.
- U and S-Frames are processed immediately and discarded.
- I-Frames are demultiplexed, flow control information reformatted, extra functions invoked and sent to the Switch processor for distribution to the TP's.
NPP DATA FLOW - TRANSMIT

- NPP obtains data for the specified link and path.
- The data is 'packed' (stored) for the frame.
- Extra functions are performed for Mux links.
- The data is multiplexed into data slots.
- Flow control information is added.
- Control slots are added (if required).
- The HDLC address and sequence numbers are added.
- The frame is transmitted over the link.
PATH DEFINITION

- Defines the routing information for traffic between nodes.
- Identifies the route a frame travels between nodes.
- Each path is uni-directional.
- Two paths are required for two-way traffic.
- Multiple slots may travel on the same path.
PATH NUMBERING

- Identifies a specific path.
- Path '0' is used on data sent directly from source to destination.
- Path '1' is used for frames sent to an intermediate node.
- Path '2' is used for frames transmitted from an intermediate node to the final destination node.
PATH ENABLING

- At boot up all paths are disabled, no data flow allowed.
- Upon NPP initialization path '0' is automatically enabled.
- Enabling of paths 1 & 2 requires explicit Addressed Packet commands.

```
+----------+ +-----------+ +-----------+
|       a   |      b    |      c    |      d    |
|    |        |        |        |
|_path 0 & 1|_path 2   |
```

The interaction to enable the paths from node 1 to node 3 is:
```
Node 1 -------> Node 2 -------> Node 3
```
Jupiter link Set-Up is complete for all nodes

Link 'a' path 0 enabled

Link status broadcast ------>

Enable path AP (addressed packet)
link 1 path 'b' ------>

<--------Path Enabled

Enable Path AP
link 'c' path 2

<--------Path Enabled

Enable Path AP
link 'd' path 2

<--------Path Enabled

Ref. Pg. 4-8
PATH REROUTING

1. Link 'a' goes down
2. Link Status message broadcast ------>
   <------ Disable path AP is sent
   Link a path 2

When all data from path 2 received. Flushing the buffers.

Disable path reply ---------->

Reconnect destination threads
to alternate path 0 link 'e'
RRTPATH (reroute) ---------->
Link 'd' path 1

Disable path (link 'd' path 1)
Do Not Flush buffers

Send trapped data of path 1 over new path 0 of new link
Remove all outgoing threads on path 1.

Reconnect source threads
to link 'e' path 0

<---------- RRPATH ACK (data rerouted)
JUPITER ARQ LAYER FUNCTIONS

- Retains the major functions of the CMP ARQ layer:
  - Extended Control field only.
  - Performs path addressing.
  - Frame routing in a 6740 delta network.
  - Link error recovery via the Nrete variable.
JUPITER ADDRESS FIELD

- Path addressing.
- Frames cannot travel over more than 2 links.

<table>
<thead>
<tr>
<th>F</th>
<th>ADDRESS</th>
<th>CONTROL</th>
<th>DATA</th>
<th>FCS</th>
<th>F</th>
</tr>
</thead>
</table>

Bit # | 8 7 6 5 4 3 2 1 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>PATH</td>
</tr>
</tbody>
</table>

Ref. P. 4-3
JUPITER CONTROL FIELD

- Extended control field only
- Nrete = the window feature used for end to end acknowledgement.
- Used for error recovery and flow control for paths 1 & 2.

<table>
<thead>
<tr>
<th>F</th>
<th>ADDRESS</th>
<th>CONTROL</th>
<th>DATA //</th>
<th>FCS</th>
<th>F</th>
</tr>
</thead>
</table>

```
Bit # | 8 7 6 5 4 3 2 1 |
-------+-------------------|
 Byte 1 | Ns 0           |
 Byte 2 | Nr P            |
 Byte 3 | C   Nrete       |
```
JUPITER MUX LAYER FUNCTIONS

- Performs the same functions as the CMP MUX layer.
  1. Multiplex customer data.
  2. Control slot commands.

- Two Jupiter specific Command Codes.
  1. Addressed packet, #25.
  2. Protocol announcement, #27.

```
+-----------------------------------------------+
<table>
<thead>
<tr>
<th>FLAG</th>
<th>A</th>
<th>C</th>
<th>80 BC CC XXXX.... DATA</th>
<th>FCS</th>
<th>FLAG</th>
</tr>
</thead>
</table>
+-----------------------------------------------+
```

Control Slot | Command Code | Command Parameters | Status Byte | Byte Count
---------------|--------------|--------------------|-------------|--------------
```

<table>
<thead>
<tr>
<th>Command Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>VCTP command</td>
</tr>
<tr>
<td>21</td>
<td>VCTP response</td>
</tr>
<tr>
<td>23</td>
<td>Unsolicited report</td>
</tr>
<tr>
<td>24</td>
<td>Quit VCTP mode</td>
</tr>
<tr>
<td>25</td>
<td>Jupiter address packet</td>
</tr>
<tr>
<td>27</td>
<td>Jupiter protocol announcement</td>
</tr>
</tbody>
</table>
```
JUPITER ADDRESS PACKETS

- Command Code of 25.
- Performs control functions for the NPP.
- Reinitialize a device, processor, link or thread
- Statistics.
- Connection establishment.
- Error reports.
- Diagnostics.
- Link error recovery.
JUPITER ADDRESS PACKET FORMAT

+---------------------------------------------+
| F | A | C | 80 BC 25 DN DN DP DP DM DM SN SN SP SP SM SM DATA... | FCS | F |
+---------------------------------------------+

Where:

80 = Control status byte.
BC = Byte count (not including self).
25 = Command code for Addressed Packet.
DN = Destination node (2 bytes).
DP = Destination port (2 bytes).
DM = Destination software module (2 bytes).
SN = Source node (2 bytes). MSB = 1 for response, 0 for command.
SP = Source port (2 bytes).
SM = Source software module.
Data = Up to 242 bytes of command dependent data.
JUPITER PROTOCOL ANNOUNCEMENT

- Identified by a command code of 27.
- Always sent by a Jupiter NPP whenever a link comes up.
- Indicates the sender's identity.
- Defines the link parameters.

Upon reception of a #27 PA by a Jupiter node:
1. The CPS (switch) decodes the slot into link status.
2. Updates the local NPP link variables.
3. Respond with its own control slot of 27.
4. Both units enter Jupiter protocol data transfer.

A non-Jupiter device receiving a #27 control slot,
1. The device will reject the control slot.
2. The original control slot is returned with a CC of A7,
   this is a 27 and the MSB of the byte is set.
3. The returned control slot is decoded, by the switch.
4. The NPP is informed to use Muxport protocol.
JUPITER PROTOCOL ANNOUNCEMENT FORMAT

`|F|A| C |80 BC 27 N# DC PT LINK SPECIFICATION...|FCS|F|

Where:

80 = Status control byte.
BC = Byte count (not including self).
27 = Command code for protocol announcement.
N# = Source node number.
DC = Device code (internal use).
PT = Protocol type 0 = Empty
     1 = IPL
     2 = Muxport
     3 = Jupiter

Link Specification = data about the source processor.
CONNECTION LAYER FUNCTIONS

- Innermost layer of the protocol.
- Interface to customer equipment.
- Rerouting of thread data between Jupiter nodes.
- All CMP connection layer functions performed.
JUPITER SUPERVISORY COMMAND FORMAT

```
+------------------------------------------+
| FLAG | A | C | 00 08 01 SF | FCS | FLAG |
+------------------------------------------+

Control Status Byte---

Supervisory Function

Slot Header---

---1 byte to follow.
```

JUPITER ADDITIONAL SUPERVISORY FUNCTIONS

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>Clear Reroute - reroute a thread going to the remote unit.</td>
</tr>
<tr>
<td>81</td>
<td>Clear cancel - request cancelation of reroute (DPI).</td>
</tr>
<tr>
<td>82</td>
<td>Clear previous - data preceding this slot to be flushed.</td>
</tr>
<tr>
<td>85</td>
<td>Clear previous acknowledge.</td>
</tr>
<tr>
<td>86</td>
<td>Clear cancel acknowledge (DPIA).</td>
</tr>
<tr>
<td>87</td>
<td>Clear reroute acknowledge.</td>
</tr>
<tr>
<td>FF</td>
<td>NPP data loss.</td>
</tr>
</tbody>
</table>

PACKET SWITCHING STANDARDS

X.25
X.29

X.75

PAD

PACKET NET

X.3

X.28

X.75

PAD

PACKET NET

X.25
X.29

X.75

X.28

X.75

X.3

PAD

PACKET NET

X.25
X.29

X.75

X.28

X.3
X.25 INTERFACE

Packet Level Procedures ------- DTE (user) ------- DCE (node) ------- Frame Level Procedures

Frame Level Procedures

Physical Level

Synchronous Circuit

Packet Level Procedures

Frame Level Procedures

Physical Level
**X.25 PROTOCOL**

**DTE/DCE Electrical Interface**

- X.21 or RS 232C compatible
- Independent of other levels

**DTE/DCE Frame Level Interface**

- Link access procedures
  - Responsible for transferring packets between DTE & DCE
  - Does not know about virtual calls

**DTE/DCE Packet Level Procedures**

- Virtual call procedures
  (Set-up, Maintain, Flow control, Clear)

- May be end-to-end (DTE/DTE)
PACKET/FRAME RELATIONSHIP

F  A  C  I  FCS  F

PACKET HEADER  USER DATA
X.25 PACKET FORMAT

- Each packet contains a 3 octet header.
- The first 4 bits = the general format identifier (GFI).
- The next 12 bits form the logical group and channel number.
- The last octet (8 bits) determines the packet type.

```
+-------------------+ 8 7 6 5 4 3 2 1
| Octet 1          |
| GFI              |
| LCGN             |
+-------------------+
| Octet 2          |
| LCN              |
+-------------------+
| Octet 3          |
| PACKET ID        |
+-------------------+
```

Where:
- GFI = General Format Identifier
- LCGN = Logical Channel Group Number
- LCN = Logical Channel Number
- Packet ID = Type of Packet
X.25 GENERAL FORMAT IDENTIFIER

- Determines format of the packet.
- Identifies control information.

<table>
<thead>
<tr>
<th>8 7 6 5</th>
<th>Packet Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 1</td>
<td>Clearing, Flow Control, Reset,</td>
</tr>
<tr>
<td></td>
<td>Restart, Interrupt Packets.</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>Call Setup Packet</td>
</tr>
<tr>
<td>Q 0 0 1</td>
<td>Data Packet</td>
</tr>
</tbody>
</table>

Where:

Bit 8 = Qualifier Bit (Q-Bit). If set (1) this bit signals that the data packet contains control information.

Bit 7 = Delivery Confirmation Bit (D-Bit). If this bit is set the acknowledgement is from end-to-end. If the bit is zero the local DCE acknowledges the packet.

Bits 6 & 5 = Indicates the modulo sequencing to be used.

<table>
<thead>
<tr>
<th></th>
<th>Modulo 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Modulo 128</td>
</tr>
<tr>
<td>11</td>
<td>Datagrams</td>
</tr>
</tbody>
</table>
LOGICAL CHANNEL NUMBERS

- All packet entering the network are assigned a LCN.
- LCN's range from 0 to 255, but 0 is reserved.
- Assigned by the DTE during call request, starting with an agreed upon upper limit and working down.
- The DCE assigns LCN's working from the bottom up.

LCN's are assigned in one of four categories:
1. Permanent Virtual Circuits (leased line).
2. One-Way Incoming Calls (contention).
3. Two-Way Switched Virtual Calls (dial up).
4. One-Way Outgoing Calls.
PACKET TYPE IDENTIFICATION

- The third octet of the header.
- Similar to the control field in an HDLC frame.
- $Pr = $ Receive Sequence Number.
- $Ps = $ Packet send sequence number.
- $M = $ More Data Bit.

<table>
<thead>
<tr>
<th>Data</th>
<th>$P(R)$</th>
<th>$M$</th>
<th>$P(S)$</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call request</td>
<td>0 0 0 0 0 1 0 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Call accepted</td>
<td>0 0 0 0 1 1 1 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clear request</td>
<td>0 0 0 1 0 0 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clear confirmation</td>
<td>0 0 0 1 0 1 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interrupt</td>
<td>0 0 1 0 0 0 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interrupt confirmation</td>
<td>0 0 1 0 0 1 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive ready</td>
<td>$P(R)$</td>
<td>0 0 0 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive not ready</td>
<td>$P(R)$</td>
<td>0 0 1 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reject</td>
<td>$P(R)$</td>
<td>0 1 0 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reset request</td>
<td>0 0 0 1 1 0 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reset confirmation</td>
<td>0 0 0 1 1 1 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Restart request</td>
<td>1 1 1 1 1 0 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Restart confirmation</td>
<td>1 1 1 1 1 1 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diagnostic</td>
<td>1 1 1 1 0 0 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Call Establishment procedure

<table>
<thead>
<tr>
<th>CALLING DTE</th>
<th>X.25 NETWORK</th>
<th>CALLED DTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALL REQUEST (LCN = 014)</td>
<td>call (LCN = xyz)</td>
<td>INCOMING CALL (LCN = 003)</td>
</tr>
<tr>
<td>CALL CONNECTED (LCN = 014)</td>
<td>call accepted (LCN = xyz)</td>
<td>CALL ACCEPTED (LCN = 003)</td>
</tr>
</tbody>
</table>
Call Clearing Procedure

CALLING DTE | X.25 NETWORK DCE | CALLED & REFUSING DTE
--- | --- | ---
CALL REQUEST (LCN = 014) | call (Lcn = xyz) | INCOMING CALL (LCN = 003)
CLEAR REQUEST (LCN = 014) | clear (Lcn = xyz) | CLEAR REQUEST (LCN = 003)
CLEAR CONFIRMATION (LCN = 014) | clear confirmation (Lcn = xyz) | CLEAR CONFIRMATION (LCN = 003)
CCITT Recommendations X.3, X.28, and X.29 define the necessary elements for a PDN to support non-intelligent asynchronous "start/stop" terminals.

When a DTE supports asynchronous terminals connected through a PDN, many of the terminal handling functions are performed by the DCE. The software in the DCE is called the Interactive Terminal Interface (ITI) Packet Assembler/Disassembler (PAD).

Recommendation X.28 defines control procedures used to establish the physical connection to the PDN, the commands the terminal user sends to the PAD, and the service signals sent from the PAD to the terminal user.

Recommendation X.3 defines a set of parameters that the PAD uses to control the terminal. The parameter values can be pre-set in network tables, set by the terminal user, and/or set by the remote DTE.

Recommendation X.25 procedures are used by the PAD to establish a virtual call to the remote DTE, to transmit and receive DATA packets, and to clear virtual calls.

Recommendation X.29 defines the control messages sent between the PAD and the remote DTE. All control or PAD messages are special X.25 DATA packets, called Qualified DATA packets.

**INTERACTIVE TERMINAL INTERFACE**
# Quick Reference Card

**International Standards Organization (ISO) Open Systems Interconnect (OSI)**

**Level 1: Physical Interface**
- Mechanical
- Electrical
- Functional

**Level 2: Data Link Interface**

<table>
<thead>
<tr>
<th>COP</th>
<th>SYN</th>
<th>SYN</th>
<th>SYN</th>
<th>SOH</th>
<th>Header</th>
<th>STX</th>
<th>Text</th>
<th>ETX</th>
<th>BCC</th>
<th>BCC</th>
</tr>
</thead>
</table>
- BSC
- Uniscope 100/200
- Mode 4C
- VIP7700

<table>
<thead>
<tr>
<th>BOP</th>
<th>Flag</th>
<th>Address</th>
<th>Control</th>
<th>Data</th>
<th>FCS</th>
<th>FCS</th>
<th>Flag</th>
</tr>
</thead>
</table>
- HDLC
- SDLC
- ADCCP

## Command/Response Summary

<table>
<thead>
<tr>
<th>Command/Response</th>
<th>Control Field</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information (Command)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information (Response)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive Ready</td>
<td></td>
<td>RR</td>
</tr>
<tr>
<td>Receive Not Ready</td>
<td></td>
<td>RNR</td>
</tr>
<tr>
<td>Reject</td>
<td></td>
<td>REJ</td>
</tr>
<tr>
<td>Selective Reject</td>
<td></td>
<td>SREJ</td>
</tr>
<tr>
<td>Set Normal Resp. Mode</td>
<td></td>
<td>SNRM</td>
</tr>
<tr>
<td>Set Normal Resp. Mode Ext.</td>
<td></td>
<td>SNRME</td>
</tr>
<tr>
<td>Set Async Resp. Mode</td>
<td></td>
<td>SARM</td>
</tr>
<tr>
<td>Set Async Resp. Mode Ext.</td>
<td></td>
<td>SARME</td>
</tr>
<tr>
<td>Set Async Bal. Mode</td>
<td></td>
<td>SABM</td>
</tr>
<tr>
<td>Set Async Bal. Mode Ext.</td>
<td></td>
<td>SABME</td>
</tr>
<tr>
<td>Disconnect</td>
<td></td>
<td>DISC</td>
</tr>
<tr>
<td>Set Initialize Mode</td>
<td></td>
<td>SIM</td>
</tr>
<tr>
<td>Exchange Station ID’s</td>
<td></td>
<td>XID</td>
</tr>
<tr>
<td>Unnumbered Acknowledge</td>
<td></td>
<td>UA</td>
</tr>
<tr>
<td>Disconnected Mode</td>
<td></td>
<td>DM</td>
</tr>
<tr>
<td>Request Disconnect</td>
<td></td>
<td>RD</td>
</tr>
<tr>
<td>Request Initialize Mode</td>
<td></td>
<td>RIM</td>
</tr>
<tr>
<td>Command Reject</td>
<td></td>
<td>CMDR</td>
</tr>
<tr>
<td>Unnumbered Poll</td>
<td></td>
<td>UP</td>
</tr>
<tr>
<td>Unnumbered Information</td>
<td></td>
<td>UI</td>
</tr>
</tbody>
</table>
Level 3: Network Interface—X.25:

<table>
<thead>
<tr>
<th>GENERAL FORMAT IDENTIFIER</th>
<th>Q</th>
<th>D</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LOGICAL CHANNEL IDENTIFIER (LCGN &amp; LCN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(S)</td>
</tr>
</tbody>
</table>

| USER DATA FIELD | |

<table>
<thead>
<tr>
<th>GFI</th>
<th>General Format Identifier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCGN</td>
<td>Logical Channel Group Number.</td>
</tr>
<tr>
<td>LCN</td>
<td>Logical Channel Number.</td>
</tr>
<tr>
<td>LCI</td>
<td>Logical Channel Identifier.</td>
</tr>
<tr>
<td>P(S)</td>
<td>Packet Send Sequence Number.</td>
</tr>
<tr>
<td>P(R)</td>
<td>Next Receive Packet Expected.</td>
</tr>
<tr>
<td>WINDOW</td>
<td>Maximum Number of Outstanding Packets.</td>
</tr>
<tr>
<td>PACKET SIZE:</td>
<td>Maximum Number of User Data Octets.</td>
</tr>
<tr>
<td>M-BIT</td>
<td>More Data Bit.</td>
</tr>
<tr>
<td>Q-BIT</td>
<td>Data Qualifier Bit.</td>
</tr>
<tr>
<td>D-BIT</td>
<td>Delivery Confirmation Bit.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data</th>
<th>P(R)</th>
<th>M</th>
<th>P(S)</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call request</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Call accepted</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Clear request</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Clear confirmation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Interrupt</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Interrupt confirmation</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Receive ready</td>
<td>P(R)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Receive not ready</td>
<td>P(R)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Reject</td>
<td>P(R)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Reset request</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Reset confirmation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Restart request</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Restart confirmation</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Diagnostic</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

TRANSMISSION: Asynchronous—uses start/stop bits; no clocks
Synchronous—clock at both ends synchronizes the data bits.

DATA CODES:
- ASCII—7-bit plus parity (=CCITT Alphabet #5)
- EBCDIC—8-bit (IBM)
- EBCD—6-bit shifted
- IPARS—6-bit shifted (airlines)
- SBT—6-bit
- Selectric—6-bit shifted
- Baudot—5-bit shifted

ERROR CHECKING:
- VRC—Character Parity
- LRC—Block Parity
- CRC—16-bit remainder; used to generate BCC/FCS