2/1/74

DCS-CR {CHANGE/RESOLUTION} CPD DOCUMENT CONTROL FORM

TO BE FILLED OUT BY SUBMITTER {PLEASE TYPE OR USE BLACK INK}

CHANGE/RESOLUTION FOR THE DOCUMENT WITH DCS LOG ID 51962

DOCUMENT TYPE ERS - CCP Release 4 FEATURE — PHASE —

CHANGE/RESOLUTION SUBMITTED BY D. E. Wilson DATE 11-30-76

SECTION 256X Programming EXTENSION 4432 MAIL STATION SM02S

CHANGE THE DOCUMENT IS

- ☐ WITHDRAWN
- ☐ UNCHANGED
- ☒ CHANGED {NO. PAGES many}

IF CHANGED, APPROVALS AS ON THE ORIGINAL DOCUMENT ARE REQUIRED

	PRE-DISTRIBUTION APPROVALS	
	INITIALS	DATE
PROJECT MANAGER D. E. Wilson	DW	12-1
UNIT MANAGER D. E. Wilson	DW	12-1
SECTION MANAGER G. Hartmann		

RESOLUTION THE FOLLOWING COMMENTS ARE RESOLVED:

COMMENTS DISTRIBUTION NUMBER	DATE	CONCURRENCE BY THE SUBMITTER OF COMMENTS {IF POSSIBLE}

TO BE FILLED OUT BY DESIGN TEAM DT _____ DESIGN TEAM REFEREE _____

APPROVED FOR DISTRIBUTION BY _____ DATE _____

NEW DEFAULT APPROVAL CYCLE ____ WORKING DAYS

REMARKS:

Release to Vault
FCO 06652 - 12-2-76 RLO

34512-02-053-502
(74872551).

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 Scope

This External Reference Specification (ERS) is one of a set which, together, present a complete description of the concepts and implementation of a network. This ERS describes the Network Processing Unit (NPU) portion of the network as implemented by the Communications Control Program (CCP) for Network Products Release 4. External interfaces are specified for conversing with a CYBER 70/170 host, other NPUs, and various remote terminals across communications facilities.

1.2 Overview

The network is intended to provide a communications service to geographically dispersed users, represented by terminals. The network will transport data between users at terminals and CYBER host applications, using a network of mini-computers called Network Processing Units (NPUs) interconnected by sets of communications lines (trunks).

The CYBER host is a CYBER 70/170 series computer, running network host software which is documented in detail in other ERSs. The primary role of network host software is to establish and control logical connections between stations (terminals and host applications). CYBER network host software also provides the following capabilities:

- access validation for stations requesting connection
- control of NPU loading and initialization procedures
- recording and reporting of status changes and operating condition of network elements and physical connections between them
- provision of configuration management procedures via the supervisory station interfaces

An NPU is a 2550/2552 series mini-computer running the Communications Control Program (CCP) software. One or more NPU(s) interface to the CYBER host via a CYBER coupler. Such NPUs are known as front-ends or first-level NPUs. A front-end NPU contains CCP and the module known as the Host Interface Program (HIP), which provides the software interface to the host. A front-end NPU may also be connected via trunks to one or more second level NPUs and/or one or more terminals. The former requires the CCP module known as the Link Interface Program (LIP); the latter requires one or more Terminal Interface Programs (TIPs), depending on the types of terminals connected. A second level NPU will have CCP with the LIP module and, optionally, one or more TIP modules, again dependent on the types of terminals connected. The HIP module is not needed and is therefore not present.

The primary role of the NPU(s) is to move data between users at terminals and programs in the host in such a way as to minimize loss, duplication, or garbling of that data. The various protocols defined within this ERS provide the major portion of the necessary data protection. On-line diagnostic software is also provided in the NPU to permit timely detection, isolation, diagnosis, and recovery of faulty elements; this software is documented in a separate ERS.

1.3 Concepts/Terminology

Terminology used in this ERS is, in general, defined in the context in which it first occurs. Terms/concepts defined below are used throughout the ERS and are collected here as an aid to the reader.

1.3.1 Node - A node is either an NPU or a host coupler connecting a front-end NPU to the CYBER host. Each node is assigned a unique number, called the node ID. Various adjectives may precede the term "node" to further describe the node's position in the network. NS and CS will have their own node ID's.

Two NPUs that are connected to each other by a trunk are called neighbor nodes; a neighbor node will always be running CCP with at least the LIP module.

A terminal node is an NPU to which terminals are directly connected via communications lines; a terminal node will always be running CCP with at least one TIP module.

A host node is a CYBER host coupler.

The source of all data routed through the network is a station (either a terminal or a host program), as is the destination of all data. A source node is the node that interfaces directly to the source station. A destination node is the node that interfaces directly to the destination station.

1.3.2 Logical Connection - A logical connection is the association of two stations via a set of corresponding table entries at the nodes that interface the stations to the network, such that data entered by one station is delivered to the other station. Each terminal is logically connected to an application program by the host. When a line becomes operational, the host is informed and then establishes terminal control blocks for the line, each of which indicates a logical connection number to be used for all data communication with the corresponding application program.

1.3.3 Network Logical Address - The network logical address is comprised of the following three numbers:

Destination Node ID
Source Node ID
Connection Number

Connection numbers are dynamically assigned as logical connections are established. Connection numbers are returned to a free pool when a station is transferred to another connection or when a connection is dissolved.

1.3.4 Message - A message is any data sent from one station to another. Messages are sent via logical connections and each message has a network logical address associated with it. Messages may be subdivided into blocks/subblocks within the network for efficient transmission.

1.3.4 Message (cont.)

A service message is a special form of message used for internal network communications software. Service messages are sent over the service channel which is characterized by a connection number of zero and which is always assumed to exist. Service messages are used to control network operation and report status.

1.3.5 Physical Link

A physical link is another name for the trunk which connects two neighbor nodes.

1.3.6 Logical Link

A conceptual communication link which consists of all of the paths between a source node and a destination node. In this software release a logical link may be completely local to a mode or may have one physical link which is limited to one trunk. This simplification results in most of the functions of network assurance being provided by the HDLC protocol on the trunk.

At a terminal node the logical link functions are performed by the NPU.

At a host node the logical link is always associated with a single CYBER coupler and the logical link functions are divided between the NPU and the host.

1.4 Configuration

The number of NPUs, number of levels of NPUs, and bandwidth of the trunks are determined by the number and types of terminals requiring service, along with the geographic distribution of each terminal and the expected volume of traffic between terminals and the host.

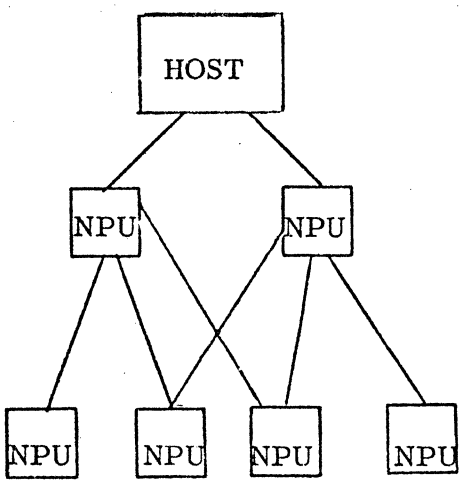
The following are intended as guidelines to what is and is not possible in planning a network configuration.

1.4 Configuration (cont.)

1. A front end may employ either a 2550 or a 2552, a remote concentrator is limited to being a 2550.

2. Up to 254 lines may be theoretically connected to any 2552 NPU. Up to 128 lines may theoretically be connected to any 2550 NPU of whatever level. The practical maximum may be less than the theoretical maximum and depends on NPU core size, terminal types, line speed and maximum character throughput rates.

3. Multiple links may be present in the hardware configuration between front end and remote concentrator NPUs, as shown in the figure below. If the path to one front end NPU becomes unavailable, SVM traffic may be routed to and from the host via the alternate path and alternate front end and logical connections may be broken and remade on the alternate logical links. Multiple logical links may be enabled to a remote concentrator concurrently. Alternate trunks may be concurrently enabled allowing upline SVMs only to use alternate paths automatically



Front-End

Remote Concentrators

4. From the CCP point of view, there is no theoretical limit to the number of front end NPUs that can be connected to the CYBER host. However, it is expected that practical limits on the number of front end NPUs will be established in host software documentation.

1.5 Document Organization

1.5.1 Information Representation

The NPU and CYBER 70/170 have different bit numbering conventions. Other manufacturers use conventions different than either of the above. These conventions also conflict with various standards, and with mathematical notation (least significant bit represents radix to zero power). The convention used for information representation throughout this document is defined below.

For the purpose of representing a field within a word or within a byte, the least significant bit of the smallest addressable unit is labeled bit zero. According to ASCII standard, this is the bit which is transmitted or received first. Pictorially, the least significant bit is depicted at the right side of a byte or word layout. Bit position numbers appear below the pictorial.

Bytes are numbered to correspond with the sequence in which they are transmitted or received over an interface, with the first byte being labeled byte zero.

Consecutive bytes of a data stream (e.g., as received from a communications line) are pictorially represented above the diagram as one or more rows of bytes with byte zero being the upper left most byte of the pictorial, independent of the way in which the data might be represented in a storage layout. In general, a contiguous data stream received from a communications line is not written to a single block of consecutive storage locations, but rather is split up into scattered storage buffers, which contain control and chaining information in addition to the data.

Consecutive words of a control block or table entry are labeled with the address of the word relative to the lowest addressed word of such a group. The lowest address is relative address zero, and appears at the top of a pictorial.

Numbers are used for the purpose of representing position and content. The numeric values of both position and content are represented in decimal, unless otherwise noted. Hardware-oriented interface definitions use the number system base (radix) common to the specific piece of equipment. The CYBER 70/170 series computers have 60-bit words and represent characters with combinations of 6 bits; octal representation is common (e.g., Ø'17 = 15 decimal). The CDC 2550/2552 series computers have 16-bit words and represent characters with combination of eight bits; hexadecimal representation is common (e.g., X'1B = 27 decimal).

1.5.2 Document Structure

The remainder of this ERS presents information relating to the CCP software, including the various Interface Programs.

Chapter 2 defines the block protocol interface between the host and the terminal NPU(s). The logical link protocol used within the NPU network is also defined.

Chapter 3 describes the host/NPU interaction necessary to dump, load, and initiallize the NPUs in the network.

Chapter 4 defines the normal operation of the network after NPU loading is complete. The definition is given primarily in terms of the service messages exchanged between the host and NPUs.

Failure and recovery mechanisms are discussed in Chapter 5. The method of detection of failure of each network element is described, followed by actions required to recover from that failure.

Chapter 6 defines the protocol between the CYBER host and the front-end NPU(s), as implemented by the Host Interface Program (HIP).

The link protocol, as implemented by the Link Interface Program (LIP), is described in Chapter 7. This protocol is used for communication between neighbor NPUs and is not of concern to the CYBER host.

Chapter 8 and 9 deal, respectively, with the Batch Virtual Terminal interface and the Interactive Virtual Terminal interface. These chapters define how data to/from real terminals is presented from/to the CYBER host. Each Terminal Interface Program (TIP) servicing a real batch or interactive terminal will transform data from that terminal such that a single, common (virtual) batch or interactive interface is presented at the CYBER host.

The remaining chapters of the ERS describe the individual TIPs provided in this release of CCP.

Appendix A is a collection of formats of tables, service messages, etc.

2.0 DATA TRANSPORT

2.1 Block Flow Control (BFC) Procedure

This section describes the procedure used to transport information between two logically connected processes. Since a supportive, lower level protocol provides delivery assurance between the two processes, this higher level procedure can, and does, assume that the logical connection between the processes is error-free. This does not imply, however, that the logical connection cannot be abnormally broken, nor does it preclude either of the processes from failing or becoming temporarily congested. Failure of either process is, whenever feasible, reported to the host via a service message. Temporary bottlenecks at a destination process are usually a result of inability to deliver data to an associated terminal or host. The procedure described herein provides a standard method for informing the transmitting process of the temporary abnormality so that subsequent data transfer can be held in abeyance until the problem is corrected.

The unit of transmission between the host and the NPU is referred to as a block. It is never more than 2047 bytes in length, including the block header. The actual length of a block is a function of the stations involved.

Figure 2-1 shows that the Block Flow Control (BFC) interface between the connected processes can be envisioned as two simultaneously active communications paths. It can be seen from the figure that the BFC procedure is fully symmetric. Ordering is maintained on each of the four paths and between FS, FD and RS in a given direction.

The types of traffic that will exist on each communications path will consist of:

- Forward Data (FD) - Textual information sent from a transmitter directly to a remote receiver. These blocks shall be either data or command blocks.
- Forward Supervision (FS) - Control information sent directly from a transmitter to remote receiver which is used to control/and or solicit status clarification of the receiver on the path.

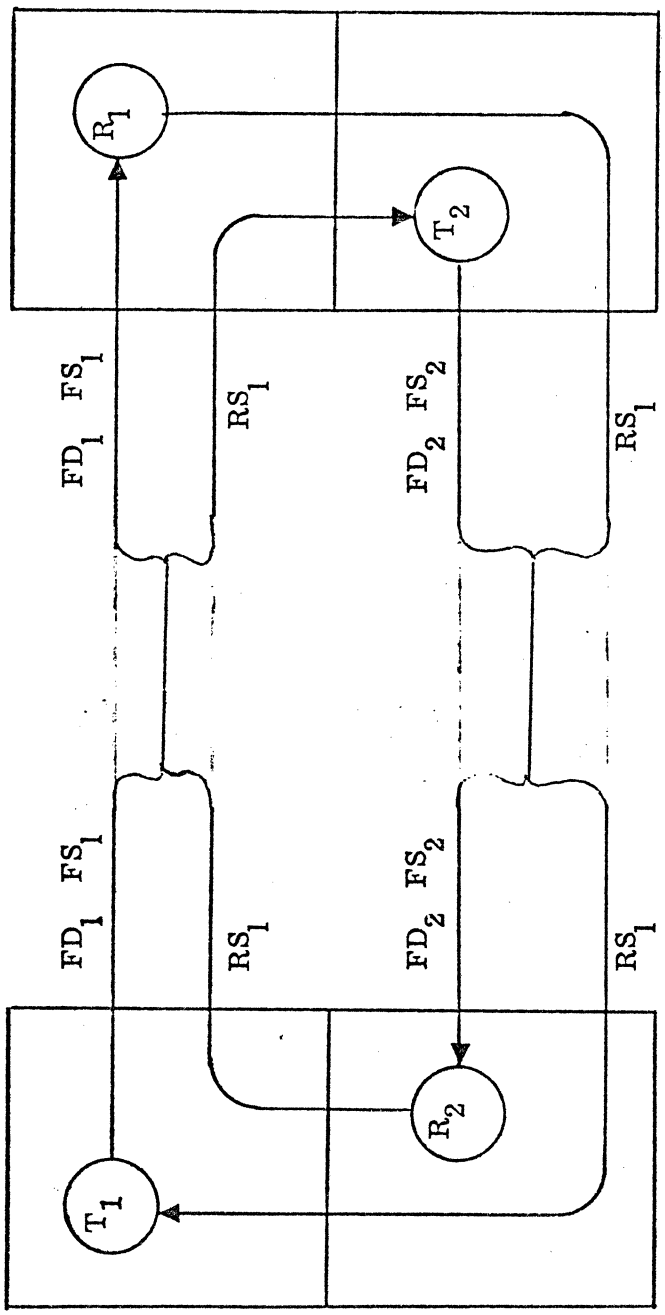
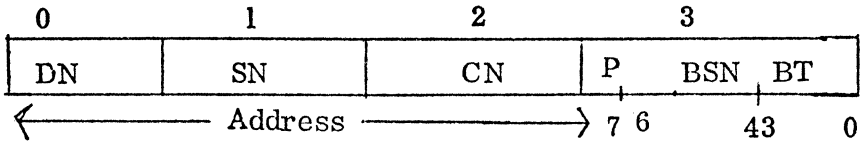


Figure 2-1 Communications Paths for Block Flow Control

2.1 Block Flow Control (BFC) Procedure (cont.)

- Reverse Supervision (RS) - answer backs sent from the receiver in response to receipt of forward data or forward supervision. RS blocks may be generated and sent even when not solicited under certain local abnormalities at the receiver.

2.1.1 Block Formats



DN	Destination Node
SN	Source Node
CN	Connection Number
P	Priority (1=high 0=low)
BSN	Block Sequence Number (0-7)
BT	Block Type

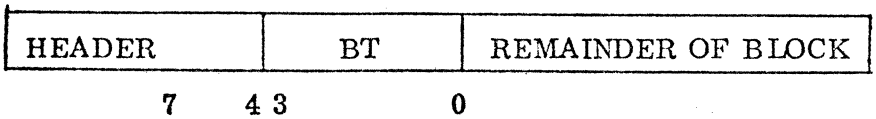
Figure 2-2 Block Header Format

Every block will have a header consisting of four bytes. The first three bytes provide a standard network address. The fourth byte contains the block priority, block sequence number and block type as depicted by Figure 2-2. The content of the remainder of the block, if any, varies with the block type.

The priority of the block is only significant when the block is required to traverse a network trunk and provides for preferential treatment for high priority blocks when trunk queueing occurs. All blocks (of any type) containing the same address must be assigned the same priority.

The BSN supplied in a downline block of type MSG, BLK or CMD must be returned in the BSN field of the upline BACK which acknowledges that specific block. When a BRK or STP is sent, the BSN field must contain the BSN which was contained in the last BACK sent for this connection. The BSN will always be zero on other upline and downline blocks.

Since only the block type differs between blocks, all block formats subsequently presented in this specification will be shown in the general form:



2.1.2 Addressing

The address, as shown in Figure 2-2, always consists of a destination node, a source node and a connection number.

2.1.2.1 Node

Each NPU has one unique node ID; each interface between host and an NPU has one unique node ID; each host has a unique node ID. Node ID = 0 is reserved for the Network Supervisor. The node ID is a single byte, yielding a range of permissible values from 0 to 255. Node IDs are assigned as build time parameters. For example, in a single-host, single-NPU system, the host interface might be node ID two, and the terminal node might be node ID three; this pair of nodes forms a Logical Link. Thus, traffic going upline (from NPU to host) has a destination node ID of two, and a source node ID of three. Traffic going downline from host to NPU has a destination node ID of three and a source node ID of two.

2.1.2.2 Connection Number

A logical connection is the association between a Terminal Control Block in an NPU and an application process in a host, by which traffic is communicated between the terminal and application process. The Terminal Control Block contains all status information relative to a particular terminal, and also contains a host-assigned connection number. The connection number is one byte, and has a possible value of 1-255. Every block traveling downline to the TCB or upline from the TCB bears the connection number of the TCB. The connection numbers of all TCB's within a given terminal node, and associated with a particular host node, i.e., on a given Logical Link, are unique.

2.1.2.3 Service Channel

A block having a connection number equal to zero is called a Service Message, and the logical connection via which it is communicated is called the Service Channel. Unlike logical connections which can be dynamically created and destroyed, the Service Channel always exists. Service Messages are always commands. Commands traveling via the service channel establish logical connections and communicate control, status and error data in support of the common equipment and software which service the logical connections.

2.1.3 Block Types

Table 2-1 provides a list of the block types, their BT codes, the traffic category to which they belong, and their function.

2.1.3.1 BLK (FD)

A BLK block is a data block containing a portion, but not the last segment, of a data message. All data blocks contain 1 to 2043 bytes of data immediately following the four byte header. The content of the data field is determined arbitrarily by the communicating processes. The format of a BLK block is shown below.

HEADER	1	DATA

2.1.3.2 MSG (FD)

A message is a self-contained unit of data communications. In half-duplex, two-party communications, the transmitter signals ready-to-receive by sending "end-of-message". Thus, a message is a data stream terminated with an end-of-message indicator.

If a message is 2043 bytes or less in length, it may be transmitted via a single MSG-type block. If a message is longer than 2043 bytes or if, as is usual, the message is segmented by the terminal or because of a desire to optimise NPU dynamic space, all segments but the last are transmitted via BLK blocks, and the last segment is transmitted via a MSG block.

The format for a MSG block is shown below.

HEADER	2	DATA

2.1.3.3 BACK (RS)

A BACK, shown below, is sent as BLK, MSG and CMD blocks are processed by the receiver to the transmitter to allow the transmitter to adjust the rate of issue of data to the delivery rate of the receiver. The transmitter should not issue unacknowledged blocks in excess of a Network Block Limit (NBL) for each connection. The BACK, which acknowledges a previously transmitted block, allows the transmitter to maintain an outstanding block count to ensure that the NBL is not exceeded.

HEADER	3

The Network Block Limit is established at each end of the connection as part of the configuration process. CMD blocks on CN=0 are not BACKed.

2.1.3.4 CMD (FD)

A command is provided to allow connected processes to communicate outside of the data stream but synchronous with that stream. The command will be received by the destination process in the same ordering sequence to the data stream or other commands as existed at source.

HEADER	4	Optional Parameters

Appendix A lists the use of the command by this software release. Note that a CMD with a CN of 0 has special system significance as a Service Message.

2.1.3.5 BRK (RS)

The BRK indicates a discontinuity in the data stream traveling in the opposite direction. The required action is to respond with a RST to delimit the point in the data stream that the BRK was actioned. Blocks are not retained by the Block Protocol for repetition. The sender of the BRK will discard all blocks received prior to the RST. A further BRK or STP must not be sent prior to receipt of the RST.

HEADER	5	RC

Any reason code (RC) included with the break will be passed to the connecting process.

Permissible reason codes for this software release are:

1 = User Break 1 received (typically means abort queue)
2 = User Break 2 received (typically means abort job)
3 = Output device not ready
4 = Illegal/invalidly formatted block received from host

2.1.3.6 STP (RS)

The STP is similar to the BRK except that no RST is sent in response and no further blocks should be sent until a STRT is received.

HEADER	6	RC

The use of an STP is indicated when a process is unable to deliver data to final destination. Examples of this are terminal inoperative, not ready, line inoperative, etc. Any reason code (RC) included with the STP will be passed to the connected process. The sender of the STP will discard all blocks received prior to the next RST received (normally caused by a STRT).

2.1.3.6 STP (RS) (cont.)

Permissible reason codes for this software release are:

1 = Terminal busy
2 = Terminal failure
3 = Batch interrupted by interactive input or output

2.1.3.7 STRT (RS)

The STRT is used after a STP to cause a resumption of data flow to the destination sending the STRT. The required action is to respond with a RST and to invite the connected process to resume data transmittal.

HEADER	7	RC

The RC field is not used in the STRT in this software release.

2.1.3.8 RST (FS)

The RST is sent in response to either a BRK or STRT. It serves to delimit the data stream and indicate the point in the data stream at which the BRK or STRT was actioned. From the time the BRK or STRT was sent until the receipt of the RST all unacknowledged blocks and all new blocks are discarded.

HEADER	8

2.1.3.9 INIT (FS)

The INIT element of the block protocol delimits the new data boundaries when a connection is first made. Either end of a connection when first set up will not accept blocks from the logical connection until an INIT is received. The second end of the connection to be set up will immediately send an INIT. Upon receipt of the INIT the first end to be set up will respond with an INIT and start accepting blocks from the logical connection. Upon receipt of the responding INIT the second end of the connection to be set up will also start to accept blocks from the logical connection.

HEADER	9

2.1.3.10 Bad Blocks

When NPU software detects a bad block, i.e., any block with block protocol fields that contain unexpected or undefined information, the NPU will discard the block. If the block is bad for some other reason, a BRK will be sent to the host. If the block is a BLK, CMD or MSG, no BACK is sent to the host. For any other other block type, no action solicited by the block is taken and it is not acknowledged. The NPU statistics word for "Block Discarded Due to Bad Address" is incremented. The header section of a bad block will be displayed at the 255X console.

Table 2-1 Block Types

Mnemonic	Name	Block Type	Traffic Type	General Function
BLK	Block	1	FD	Data block which is a non-EOM block of a multi-block message
MSG	Message	2	FD	Data block which is the EOM block of a multi-block message or all of a single block message
BACK	Block Acknowledgment	3	RS	Block acknowledgement for block transmitted in the opposite direction
CMD	Command	4	FD	Command
BRK	Break	5	RS	Indicates a discontinuity in the data stream traveling in the opposite direction
STP	Stop	6	RS	The forward data stream is undeliverable and should be stopped
STRT	Start	7	RS	The forward data stream may be started
RST	Reset	8	FS	Transmitter has cleared out logical connection after receiving a BRK or STRT
INIT	Initiate	9	FS	Initiate a logical connection.

2.2 Delivery Assurance

Delivery assurance is protection of data between a local NPU and its remote neighbor.

2.2.1 Segmentation of Blocks/Messages

The unit of data assured is the block. Blocks are generated by the source station, passed through the network and delivered to the destination station in the order of generation. One of two possible priorities must be assigned to a block by the source station. Obviously if ordering is to be preserved, all blocks and all forward supervision block protocol elements on a connection traveling in the same direction must be assigned the same priority.

Block delivery across inter-nodal physical links is performed in a manner which approximates a pre-emptive-resume priority queue dispatch discipline. For this process blocks transmitted in a link are segmented into sub-blocks to ensure that an opportunity for pre-emption will occur at discrete maximum intervals. Segmentation is of functional concern only to the Link Interface Program, although implementation considerations dictate that Host and Terminal Interface Programs and the receive side of the Link Interface Program position the data in buffers in a manner which facilitates sub-blocking. Block priority for blocks arriving from the host coupler is established by the host before setting up the data transfer.

2.2.2 Logical Link

A logical link is the logical entity which monitors the transfer of data blocks and block protocol elements for all connections between two end points in the network. Unless both ends of a logical link are configured and operational, all such data is discarded and no connections are permitted. When both ends of a host-to-local logical link are configured, the host is notified with a Logical Link Status Operational service message immediately; this logical link remains operational until deleted by the host. When both ends of a host-to-remote logical link are configured, the host is notified with a Logical Link Operational service message from the local NPU as soon as a Clear/Reset exchange occurs between the local and remote NPU's (see below); this logical link will go inoperative upon a physical link failure. The host is notified with a Logical Link Status Inoperative service message from the local NPU. NS must explicitly delete the logical link configuration which will cause all associated connections to be deleted and all data blocks and block protocol elements for these connections to be discarded. No connections are permitted on the logical link until a Clear/Reset sequence establishes an operational state again.

2.2.3 Block Format For Delivery Assurance

DN	SN	CN	TYPE	DATA

DN - Destination Node
SN - Source Node
CN - Connection Number
TYPE - Type of block. This field has the following format:

7	6	5	4	3	2	1	0
P R I D	Reserved for Block Sequence Number			BT			

PRID - Priority Designator. Set for high priority blocks.
BT - Block Type. The Block Types given in Section 2.1 are augmented by the following type used to control the logical link:

ACTL - Assurance Control. Combines infrequently used supervision blocks into one type. The subtype code is the first byte of the data field.

Subtype

CLR - Clear=0. Sent by the local end to the remote end of a logical link at initialization time. The CLR is repeated until the PRST is received. The CLR contains the Logical Link Regulation Level in the second byte of the data field.

PRST - Protocol Reset=1. Sent by the remote end of a logical link at initialization time after the receipt of a CLR. Normal data blocks may be transmitted following a PRST. The local end will accept blocks after receipt of a PRST. The PRST contains the Logical Link Regulation Level in the second byte of the data field.

REGL - Regulation=2. Sent by either end of the logical link when the local regulation level changes. The REGL contains the new Logical Link Regulation Level in the second byte of the data field.

2.2.4 Services Messages

When a physical link fails, all blocks to be transmitted on the link are discarded by the link protocol. Any service messages that must be protected across a link failure, i.e. unsolicited line status, will be retained by the Service Module and repeated when the link again becomes operational. While the physical link is inoperative and no alternate path is available, new service messages are retained by the Service Module.

INTENTIONALLY LEFT BLANK

3.0 NPU INITIALIZATION

This section describes the features necessary to make each NPU a fully operational node in the network. This includes loading, dumping, and initialization.

The host normally attempts to load/dump an NPU only if it fails or if the Network Operator specifically requests a load. In the case of a failure or suspected failure, the host will first dump the NPU. The NPU will then be loaded. If the first attempt to load the NPU fails, a dump will be taken. Dumping will be inhibited on subsequent consecutive attempts to load.

The Load/Dump process varies with the type and location of the failed element. NPU's that are local to the host are loaded and dumped via the CYBER Coupler under the control of the PPU.

Failure of a local NPU is always detected by the host PPU channel coupler driver. Upon detecting a failure condition or upon receiving a Force Load SM, the NPU will stop servicing the channel coupler. The PPU is then able to detect the NPU failure by a time out of the protocol over the channel coupler.

NPU's not local to the host are loaded and dumped using a down-line procedure. When a remote NPU fails, the deadman timer is activated which reads a primitive load/dump program into the NPU from a cassette and starts execution of this program. The failed NPU establishes contact with a neighbor NPU. The neighbor NPU notifies the host, which then sends a load/dump overlay program to the neighbor NPU. This overlay program allows the neighbor NPU to communicate with the failed NPU using a restricted set of the communication protocol during this load/dump procedure. From this point on, the load/dump procedure is controlled by the host.

3.1 Load/Dump Phases

The loading and dumping of 255X's is multiphase and differs between the 2550 and the 2552.

The 2550 dump is three phase.
First main memory is dumped; secondly a small program is loaded in to main memory and executed to read the file registers and checksum the RAM, and thirdly the file registers and checksums are dumped from main memory.

The 2550 load is two phase.
First the contents of the RAM are loaded into main memory together with a small loader program which then executes to move the data from main memory to RAM, and secondly the main memory is loaded.

The 2552 dump is three phase.
First main memory of the Base Processor is dumped. Secondly, a small program is loaded into the Base Processor and executed to transfer the Mux Processor main memory and the file registers of each processor to the Base Processor main memory and obtain the checksum of each RAM. Thirdly, the area of Base memory containing this information is dumped.

The 2552 load is two phase.
First the Base main memory is loaded with the contents of Mux main memory, both Mux and Base RAM's and a small loader program. The loader program then executes to move the data to Mux main memory and each of the RAM's. Secondly, the Base main memory is loaded.

The format of NPU dump output is described in Appendix E of the Network Supervision ERS - Release 4.

3.2 Local NPU Loading

Any NPU connected to the host is regarded as a local NPU and is loaded directly over a channel coupler by a PPU. Micro memory is always loaded before main memory.

Main memory is written directly by the PPU to the NPU. The PPU first specifies a start location by writing memory address zero and one. The PPU then performs successive data transfer to the NPU, re-reading each area and comparing word for word to ensure correct transfer. When complete, the PPU issues a Start NPU function. The NPU is now executing the program just loaded and responds to the PPU with an Idle response. If there is no Idle response, the NPU has failed.

3.2 Local NPU Loading (cont.)

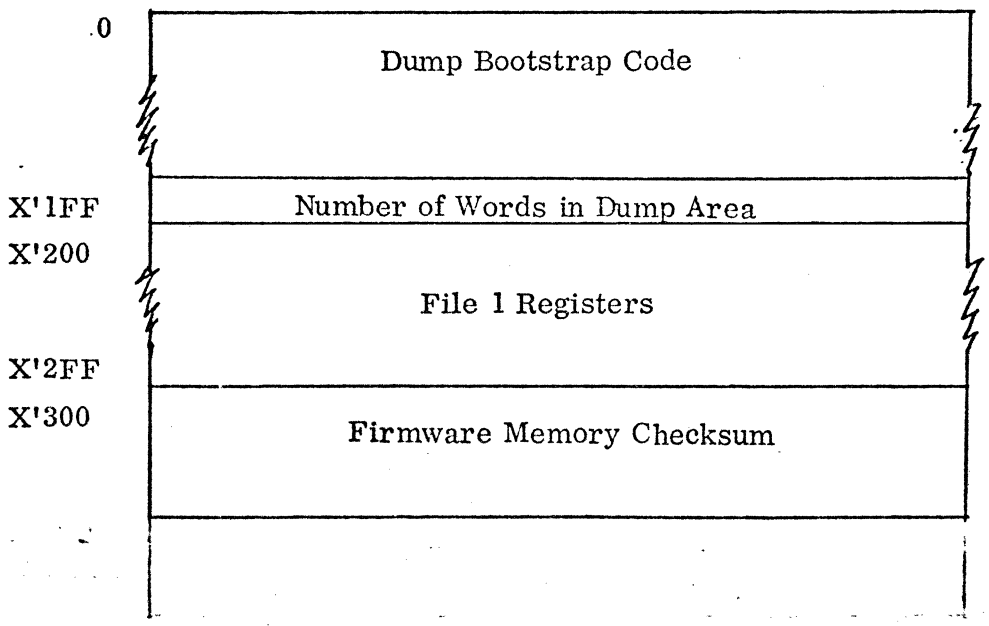
Micro memory cannot be directly loaded by the PPU, only by a program executing in the NPU. The PPU must load a special micro-memory-loading program into the NPU main memory and cause it to be executed. The NPU then loads its own micro memory and issues an "Idle" response to the PPU.

3.3 Local NPU Dumping

A local NPU is dumped by the PPU over the channel coupler. The PPU first halts the NPU. Then the PPU saves the coupler status word, the NPU status word and the order word. The PPU now reads the entire contents of NPU directly accessible memory over the coupler. The 2550 and 2552 require different "Dump Bootstrap" programs since machine organization is different. In both cases, the "Dump Bootstrap" program is loaded into the main memory which is accessible to the PPU via the coupler and the program is initiated starting at location zero. When the program has completed execution, the memory will be formatted as shown in Figure 3.1 for a 2550 and as shown in Figure 3.2 for a 2552. The "Dump Bootstrap" then writes a value of 8 to the NPU status register of the coupler to indicate "Ready for Dump" and halts execution.

The PPU reads the location X'1FF to determine how many words of memory need to be dumped.

Figure 3.1 Dump Bootstrap NPU Memory Format (2550)



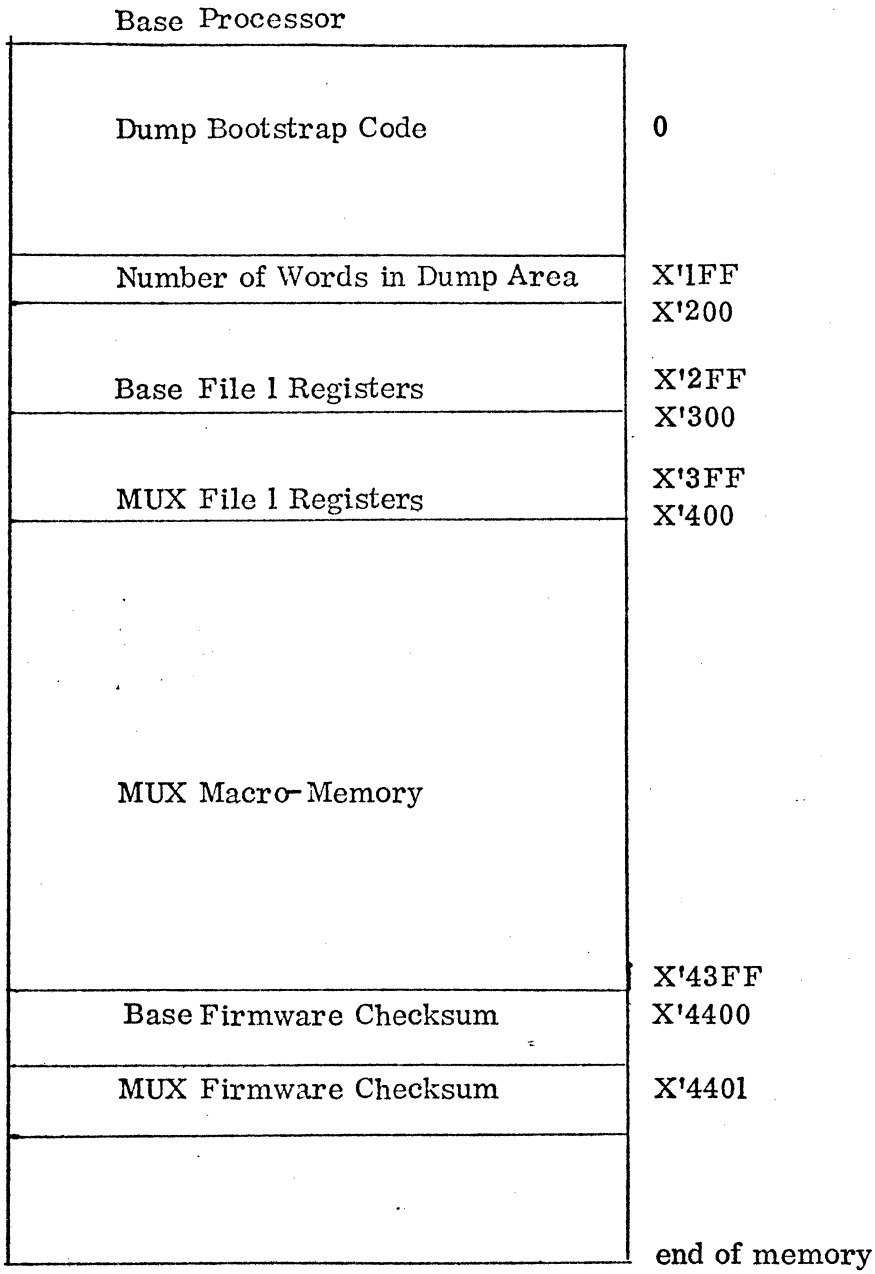


Figure 3.2 Upline Dump File Format (2552)

3.4 Remote NPU Loading

All NPU's which are not coupled to the CYBER Host in which NS is running will be downline loaded and dumped. An operational NPU that is an immediate neighbor to the failed NPU must be available. The failed NPU and its neighbor must have a trunk between them that is capable of operation. The sequence of events necessary to dump and then load an NPU are as follows:

1) The failed NPU sends a load request to its neighbor NPU (the RIM element of the CDCCP protocol).

2) The LIP in the neighbor NPU sends a Load Request Service Message to NS.

3) NS responds by loading an overlay program into the neighbor NPU's overlay area.

4) NS sends an Overlay Data Clear SM to signal the start of the load/dump process.

5) For a dump phase of the multi-phase dump/load process, NS sends an Overlay Data SM to the overlay program commanding it to dump a specified area of the failed NPU's memory.

6) The overlay sends a dump command to the failed NPU to dump a memory block. The memory block plus overhead must fit into the Overlay Data Response SM (i.e., less than 256 bytes).

7) The program in the failed NPU responds with the memory block, which the overlay sends on to NS as an Overlay Data Response SM.

8) The overlay continues to send dump commands until the dump is complete.

9) For a load phase of the multi-phase dump/load process, NS starts sending load commands to the overlay. The load command contains an address and one or more words of object code to be loaded into the failed NPU.

10) The load commands are sent in batches with an acknowledgement being required from the overlay for each batch of commands successfully executed. NS sets the response count field of the last command of each batch to the batch value. The overlay will acknowledge the batch only when the number of commands specified have been processed.

3.4 Remote NPU Loading (con'td)

10. NS continues to send batches of load commands until the load is complete. It is the NS responsibility to restrict the number of load commands outstanding to a level which avoids network overloading while keeping the load process running at a reasonable rate.

11. When NS has received an acknowledgement for all load commands, the START command is sent to the overlay. The START command is passed along to the failed NPU. If this is the last load phase of the multi-phase load/dump then NPU will enter its initialization procedure and attain operational status.

3.4.1 Load Request SM

Each remote NPU is provided with a deadman timer, which is reset periodically. If the timer expires, the NPU has failed and a program is bootstrapped in from a tape cassette to attempt a reload. This program sends a Request for Initialization over a trunk, using a nonsequenced element of the CDCCP protocol. This request is sent over the selected trunk for a time period and, should no reply be received, the request is repeated over a different trunk. The request will be sent to neighbor NPU's, in sequence, until a reply is received. The NPU expects to receive its load over the replying trunk. If the load does not commence after a time interval has elapsed, the deadman timer is activated and the NPU re-enters the Request Initialization Mode sequence.

Parameters

Link remote node ID

Port, Subport } from neighbor NPU to failed NPU

Purpose

To inform NS that an NPU is requesting a load.

Stimulus

For an NPU, receipt by the LIP of the Request Initialization Mode element of the CDCCP protocol.

3.4.1 Load Request SM (cont'd)

Action

Upon receipt of the Load Request SM, NS initiates the remote NPU load procedure if the NPU is enabled. The load/dump overlay is installed in the neighbor NPU that originated the Load Request SM. As each block is loaded, the overlay program will send an Overlay Data Response. Any error causes the procedure to restart with the Load Request SM.

3.4.2 Force Load SM

An NPU may be forced into a failure state by sending it a Force Load SM. This causes the NPU to activate the deadman timer and start sending Request for Initialization down the trunks.

Parameters

None

Purpose

To allow NS to force an NPU to the inoperative state so that it can be re-loaded.

Stimulus

A command entered at the NOP Station requesting the loading or disabling of an NPU.

Action

If the Destination Node (DN) specifies the primary node ID of an NPU, the NPU allows the deadman timer to expire which will cause a load to be requested.

3.4.3 Overlay Data SM/Dumping and Loading

The Overlay Data Service Message for dumping and loading has four forms: one to direct the loading of the NPU; the second to cause the NPU to begin execution of the loaded overlay the third to cause dumping of the NPU, and the fourth is the overlay data clear to signal the start of a new load.

3.4.3.1 Overlay Data SM Load Command

When a neighboring NPU detects an initialization request, it sends a Load Request SM to NS requesting a load of the failed element. NS must first install the load/dump overlay in the neighbor NPU program using Overlay Program Block SM's in order to provide the additional code necessary to perform the load of the failed NPU. It should be noted that NS will preempt an already existing overlay in the NPU in order to do the load. Thus, a diagnostic overlay may be forced to terminate to make room for the NPU load overlay.

The load module for the failed NPU is sent from NS to the neighboring NPU in several blocks using Overlay Data SM's. The overlay program attempts to load each block into the failed NPU using nonsequenced elements of the CDCCP protocol. The program in the failed NPU checksums and verifies each block received and causes an incorrectly received block to be retransmitted. Should a load be unsuccessful for any reason, the failed NPU reenters the initialization request sequence to restart the load process from the beginning. After several unsuccessful attempts to load a failed NPU, NS declares the NPU down.

Parameters

Overlay ID

Indicator
- Dump command
- Load command
- Start command
- Clear command

Port number (from the neighbor NPU to the failed NPU)

Subport number (from the neighbor NPU to the failed NPU)

Batch Count

Beginning address

Checksum

Overlay Data (1-105 words)

Purpose

To load data into main memory of a failed NPU.

Stimulus

Receipt by NS of a Load Request SM.

3.4.3.1 Action

Each load command received results in the data being passed by the CDCCP protocol over the trunk and installed in the failed NPU memory by the Bootstrap Program. When the protocol acknowledgement is received by the LIP this is passed back to the Load Overlay. If the load command contains a non-null batch count field and the Load Overlay has processed that many load commands then a batch acknowledgement is sent to NS. If a load command with a null batch count field is received and a batch response has been solicited and the batch total has now been reached, then a batch acknowledgement is sent to NS. A batch count of 1 is used by NS after the failure of any batch to cause a clean-up.

3.4.3.2 Overlay Data SM Start Command

After sending the last data block of a load, NS sends an Overlay Data SM start command to the NPU overlay. This causes the overlay program to send a start command to the failed NPU using a nonsequenced element of the CDCCP protocol, causing it to start executing at the address found in locations 0 and 1.

Parameters

Overlay ID

Indicator { Dump command / Load command / Start command / Clear command }

Port number (from the neighbor NPU to the failed NPU)

Subport number (from the neighbor NPU to the failed NPU)

Purpose

To start a program in an NPU.

Stimulus

Sent by NS after successful completion of a load sequence.

Action

Begin execution.

3.4.3.3 Overlay Data SM Clear Command

Parameters

Overlay ID

Indicator { Dump Command / Load Command / Start Command / Clear Command }

Port Number

Subport Number

Purpose

To clear status and count fields in the load overlay to initial conditions at the start of a load/dump process. Necessary to allow multiple attempts to load without reloading the overlay.

Stimulus

NS starts/restarts the load process.

Action

Clear overlay status and counters for the trunk specified.

3.4.4 NPU Initialized SM

Parameters

Version, Cycle, Level Number } Specifies which CCP software has been initialized

Purpose

To notify NS that the NPU is now operational.

Stimulus

CCP Initialization is complete.

The NPU Initialized SM is sent by an NPU after a reload, at the time it has completed initialization and is ready for configuration.

3.4.4 NPU Initialized SM (cont.)

Several procedures are called to initialize and bring to operational state various NPU functions. These include the initialization of software tables and data structures, the communication subsystem and local devices.

Finally OPSMON is given control and the NPU is fully operational.

One of the final tasks performed by the newly operational NPU is to send an NPU Initialized SM to NS, and repeat it until a SM is received from NS.

3.5 Remote NPU Dumping

Before performing a load of an NPU, NS normally attempts to dump the element. As the majority of loads are a result of an element failure, this ensures that any relevant diagnostic or debugging information is saved before the element is reloaded. In order to avoid redundant dumps however, NS inhibits the dump if the NPU has not been operational since the last dump. This occurs, for example, on the second and subsequent load attempts after the first load attempt has failed.

3.5.1 Overlay Data SM Dump Command

The NPU load program sent to the neighbor NPU also has the capability of dumping the failed NPU. If NS wishes to dump the NPU, it sends the neighbor NPU a series of Overlay Data SM's containing dump commands prior to performing the actual load. Each dump command contains a first and last address to dump and causes the NPU to send one or more Overlay Data Response SM's containing the dump information in fixed size blocks. NS may, therefore, meter the amount of dump data on the network by varying the amount of core dumped by each Service Message.

A remote NPU dump is multi-phase as previously described. Note that remote 2552s are not supported.

3.5.1 Overlay Data SM Dump Command (cont'd)

Parameters

Overlay ID

Indicator { Dump command / Load command / Start command }

Port number (from the neighbor NPU to the failed NPU)

Subport number (from the neighbor NPU to the failed NPU)

Beginning Address

Ending Address

Purpose

To request portions of an NPU main core to be sent to NS for evaluating an element failure.

Stimulus

NS will send a dump request on discovering the NPU has failed and is now in the initialization process.

Action

The neighbor NPU sends a Dump command to the failed NPU and receives a response from the failed NPU. The response is converted into an Overlay Data SM Dump Response and transmitted to NS.

3.5.2 Overlay Data SM Dump Response

Parameters

Overlay ID

Indicator { Dump Response / Load Response / Start Response }

Port Number
Subport Number
Response Code
Beginning Address
Overlay Data (1-105 words)

3.5.2 Overlay Data SM Dump Response (cont.)

Purpose

To transmit a portion of NPU main memory to NS for evaluating an NPU failure.

Stimulus

Sent in response to the Dump Command.

Action

The Load/Dump Overlay sends the SM to NS.

4.0 NETWORK OPERATION

4.1 Logical Connection Procedures

A logical connection is the association of two stations via the assignment of a network logical address. The network logical address is a set of three numbers consisting of two node IDs followed by a connection number. The two node IDs represent the nodes at which each station interfaces to the network; the order in which they appear in the network logical address specifies the direction of the connection (the destination node appearing first). The connection number specifies a full duplex logical channel connecting the stations. Connection number zero is reserved as a permanent service channel for service message communications.

The set of logical connections which potentially exist between stations supported by a node pair is referred to as a logical link. A logical link must be explicitly established before logical connections may be assigned to it. (The service channel is an exception to this rule.)

The Network Supervisor Program (NS) and the Communication Supervisor Program (CS) in the CYBER host are responsible for the control of logical links and connections respectively in the network. All logical links and connections are explicitly configured, reconfigured and deleted by NS/CS by the use of network Service Messages (SM). Since the logical connection scheme is related to the general scheme for bringing up the network, this relationship is now described.

4.1.1 Initiating Logical Connections

NS will establish all logical links which the current state of the network permits. NS informs CS of each logical link which is to be established. The network will inform CS of the initial logical link regulation level. CS will configure the lines or recover the configuration status of the lines, depending upon NPU status. Whenever a line is reported to CS as operational, CS will configure and attempt to connect each terminal on the line.

A configure or reconfigure terminal SM is issued which includes the CN assigned by CS for the connection. When the configure or reconfigure action has been performed, the block protocol will be initiated and the connection will be in use. CS is informed of the successful completion of the configuration by a normal response.

NS is informed of an NPU entering this active state by the arrival of an NPU Initialized SM in the case of a failed NPU or the arrival of the first Trunk Operative SM in the case of an operational NPU rejoining the network.

4.1.2 Changing Logical Connections

A change to a logical connection may be required when a TCB is already configured. This is accomplished with a Reconfigure TCB SM.

4.1.3 Deleting Logical Connections

A logical connections fails when an element (i.e. line, logical link, application) required to support it fails, or is disabled by operator command. The NPU can be informed of the termination of the logical connection either explicitly, via a Reconfigure TCB SM changing the connection number, or implicitly by deleting the TCB or the LCB on the logical link configuration.

4.1.4 Configure Logical Link SM

Parameters

ID1, ID2 } Node IDs (destination node, source node) forming logical link
HO Host Ordinal

Purpose

Sent by NS to establish both ends of the logical link. The association between node IDs and coupler is predefined. The SM has a DN corresponding to the primary node ID of the NPU supporting the logical link.

Stimulus

NPU initialized SM or Trunk Status (Operational) arrives at NS or an enable logical link operator command is received.

Action

Establish the data structure necessary to support the Logical Link and send a response to NS. Following the start up of the Logical Link, a Logical Link Status SM will be sent on the Logical Link reporting the presence of the logical link.

4.1.5 Delete Logical Link SM

Parameters

(As for Configure Logical Link)

Purpose

To delete a logical link which may no longer be allowed or required or has failed.

4.1.7 Logical Link Status Response SM (cont.)

Stimulus

NS/CS request status of one or more logical links.

The NPU responds with an unsolicited status message when the logical link regulation level changes.

Action

The NPU constructs and sends the status message.

4.2 Lines

The dynamic configuration of lines and terminals is performed by the exchange of SM's between the NS/CS in the CYBER host and the Service Module in the NPU(s). An overview of the process and the SM's involved is given by Figure 4.?-1. The different actions performed for modem conditioning and auto recognition will be found in the appropriate TIP section.

After a line is configured, it is automatically enabled. This allows the line to be monitored. When the line is reported operational, Terminal Control Blocks are configured.

4.2.1 Line Numbers and Line Types

A line number is a two byte quantity: port, subport. The port number corresponds to the two hexadecimal digits which appear on the CLA address thumbwheel switches, and has the values 1-X'FE. The subport field is not currently used and must have a value of zero.

Each line is configured for a line type and a TIP type. A combination of transmission facility, carrier control, CLA, modem and circuit types is called a line type. The line type codes are defined in Appendix A. TIP types include ASYNC TIP, Mode 4 TIP and HASP TIP.

Each line is configured with a Host Ordinal. This value (0-15) is validated each time a service message is received for the line and included in each service message sent to CS referring to the line.

4.1.5 Delete Logical Link SM (cont.)

Stimulus

A node or host interface supporting a logical link fails, an operator command is received, or a Logical Link Status SM reporting logical link failure is received.

Action

Clean up and release space associated with the connection directory and data storage where found. Set CN=0 for TCB's. Respond to NS.

4.1.6 Logical Link Status Request SM

Parameters

ID1, ID2 } Node ID's forming the Logical Link
HO Host Ordinal

Purpose

To allow NS/CS to request status of any or all logical links.

Stimulus

Sent when NS is recovering the status of Logical Links in the network.

Action

One or more Logical Link Status Response SM's are sent.

4.1.7 Logical Link Status Response SM

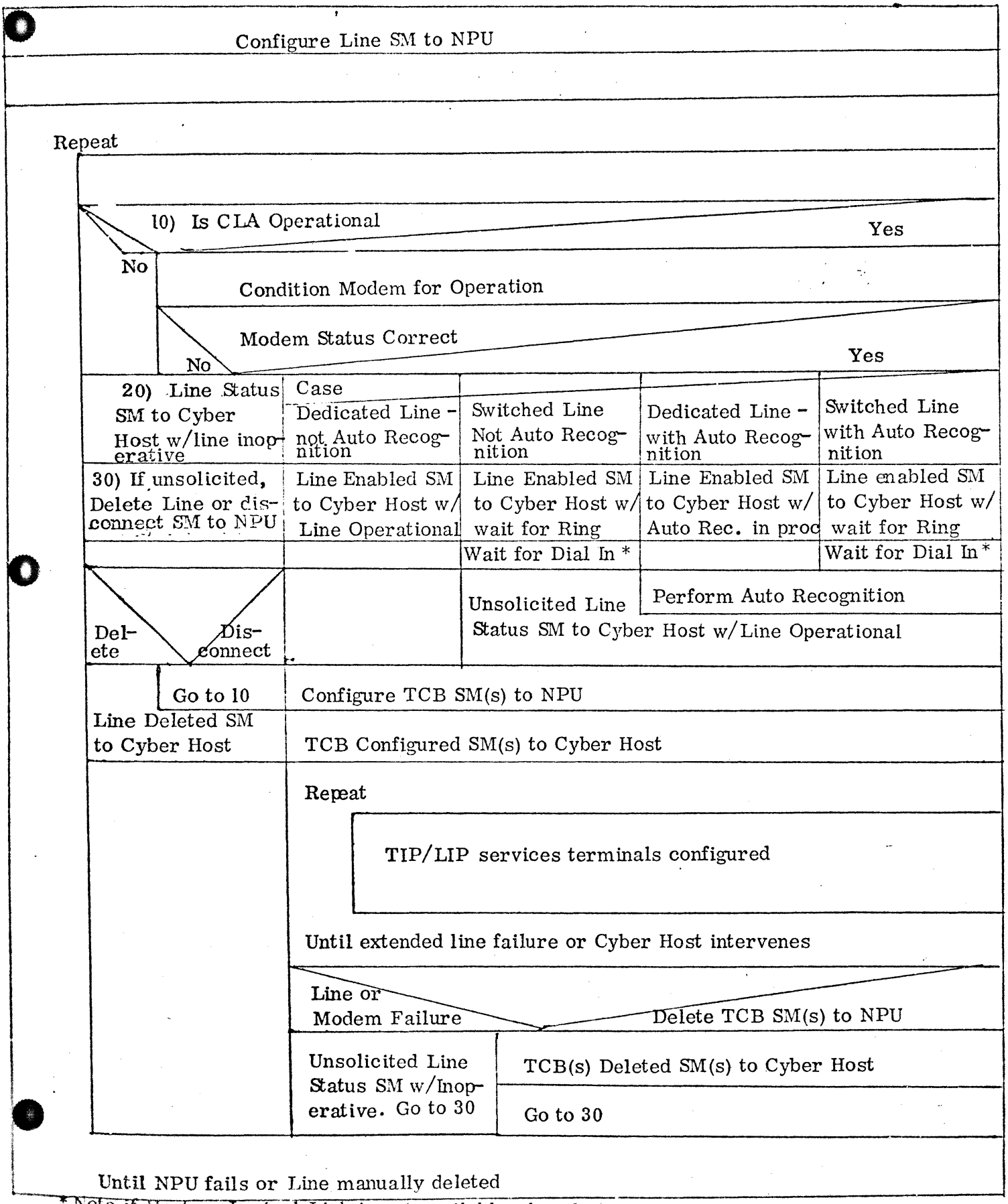
Parameters

ID1, ID2 } Node ID's forming the Logical Link
HO Host Ordinal
RC Response Code
RL Regulation Level
INIT Initial Flag
TOT Total Responses

Purpose

To inform NS of the operational status of logical links.

To inform CS of the initial logical link regulation level and each change in the regulation level.



* Note if Host or Logical Link is not available when dial in occurs, the dial in is ignored.

Figure 4-2.1 Configuration Process

4.2.1.1 CLA Types

2561-1: General Purpose asynchronous CLA with the following features:

- Half or full duplex
- Character length of 5, 6, 7 or 8 bits (exclusive of parity, if any)
- All standard speeds to 9600 baud
- Input and output speed may be different
- Even, odd or no character parity checking and generation
- Stop bit length of 1, 1.5 or 2 bit times
- Self test mode (loop back)
- All of above selectable by program command
- Full RS232/CCITT V24 interface including reverse channel detection and control, terminal busy, and originate mode
- Break detection and generation
- Data transfer overrun detection

2560-1: General purpose synchronous CLA with following features:

- Half or full-duplex operation
- Code length 6, 7, 8 or 9 (8 + 1 parity) bits
- Frame synchronization using character established by software
- Even, odd, or no character parity checking and generation
- Self-test mode (loop back)
- All of above selectable by program command
- Speeds up to 9600 bps determined by modem - provisions for external clock source
- Full RS232C/CCITT V24 interface
- Data transfer overrun/underrun detection

4.2.1.1 CLA Types (continued)

2563-1: Synchronous CLA for use with CDCCP

- Half or full-duplex operations
- Self-test mode (loop-back)
- Code length selectable on output
- All of the above selectable by program command
- Compatible with SDLC frame and bit structure
- Speeds up to 9600 bps determined by modem - provisions for external clock source
- Full RS232C/CCITT V24 interface
- Data transfer overrun/underrun detection
- Frame checksum generation and verification

4.2.1.2 Modem Type - The Modem Type specifies an interface standard (e.g., EIA RS232C) and one or more AT&T Data Sets for which the defined control procedures are compatible. Any other manufacturer's modem may be used provided it is compatible with the listed AT&T Data Sets.

4.2.1.3 Circuit Type - Switched or Dedicated. For switched lines, the modem is conditioned by means of the Data Terminal Ready interface signal to answer incoming calls upon receipt of the Ring Indicator signal from the modem.

4.2.1.4 Transmission Facility - The communications line must be identified as being half-duplex (HDX) or full-duplex (FDX). This represents the characteristics of the communications facility, not the mode of data transfer over the line. It is important not to assume that a two-wire circuit (line) is necessarily an HDX facility; some modems will operate FDX with 2-wire circuits.

4.2.1.5 Carrier Control - The NPU can operate FDX facilities with constant carrier or controlled carrier. Constant carrier means that transmit carrier remains on continuously, and line failure is reported if received carrier remains off for a contiguous period of time which equals or exceeds the Failure Verification Period. With controlled carrier, the transmit carrier is raised and lowered with each transmission block, and received carrier is expected to behave in similar fashion.

4.2.2 Configure Line SM

Parameters

Port

Subport

Host Ordinal

Line Type

Terminal Type (including TIP Type and the Auto Recognition flag)

Field Name$_n$ } One or more field name/field value pairs (see Table A-7)
Field Value$_n$ }

Purpose

To create the control structure in the NPU to support the operation of a Line which has not been previously configured.

Stimulus

CS requires to configure lines after an NPU is loaded or in response to an operator command.

Action

All Configure SM's are driven by a control block descriptor string. There is one such descriptor string for each type of configurable control block in the NPU. This descriptor string equates a Field Name to a Field Position within the control block and allows the Field Value to be correctly assigned. Additionally, an optional Field Action may be defined and associated with a Field Name. This Field Action allows such features as validating the Field Value, assigning chains to other structures, performing other associated actions required by the field.

The port and subport must not be greater than the maximum allowable port and subport defined for the NPU.

After performing the configuration defined by the control block descriptor string together with any defined actions, the Service Module attempts to enable the newly configured line. At the completion of the enable process, the Line Enabled Response SM is returned.

4.2.2 Configure Line SM (cont.)

Action (cont.)

Response to configuration of dedicated line: if the modem of a dedicated line signals Data Set Ready and for constant carrier, both Clear to Send and Data Carrier Detect are on, line enabled is reported; otherwise, line inoperative is reported.

Response to configuration of switched line: Line enabled as a normal response is generated immediately if a Ring Indicator is present. Line enabled with no Ring Indicator is generated immediately if no Ring Indicator is present, followed by a Line Operational SM when a dial-in connection occurs. At this time, Ring Indicator is signalled and the NPU returns Data Terminal Ready (DTR) to answer the call. If, when Ring Indicator is signalled, the host or logical link is not available, the NPU will ignore the dial in.

Line Operational is reported if auto-recognition is not specified, A 30-second timer is started if auto-recognition is specified. If no response is obtained within the 30-second period, the TIP responds with line not operational; the host should disconnect at the earliest opportunity. If a response is obtained, line operational is reported containing the results of auto-recognition.

4.2.3 Delete Line SM

Parameters

Port
Subport
Host Ordinal

Purpose

To change line status to LCB Not Configured.

Stimulus

Disable Line Command from the LOP.

Action

Delete TCB's and set line state to Not Configured. The Delete Line SM is also treated as a positive response to an unsolicited Line Inoperative Status SM.

4.3 Terminals

4.3.1 Terminal Identification

Terminals are identified in service messages by specifying the line, the hardware address, Device Type (DT), Terminal Class (TC), and Host Ordinal (HO). The line identification is given in Section 4.2.1.

	CA	TA
Mode 4A	X'70-X'7F	X'60
Mode 4C	X'70-X'7F	X'61-X'6F
ASYNC	0	0
HASP	0	0-7

The hardware address varies with the protocol being used by the terminal. Mode 4A may have one or more cluster controllers on a line but only a single terminal on the cluster. Mode 4C may have one or more cluster controllers per line and one or more terminals per cluster. The ASYNC TIP does not support any terminal addressing capability. The HASP TIP uses the TA as the stream number and does not use the CA. For HASP, the Device Type is combined with the TA to form the hardware identifier. Card readers and line printers may use the full range of stream numbers but plotters must share the range with card punches.

4.3.2 Configure Terminal SM

Parameters

Port
Subport
Host Ordinal
Cluster Address
Terminal Address
Terminal Class and Device Type
Host Ordinal
Field Name$_n$, Field Value$_n$ } One or more field name/field value pairs; i.e., the connection number. (See Table A-8)

Purpose

Allows CS to configure a TCB establishing initial values for all fields in the TCB.

4.3.2 Configure Terminal SM (cont.)

Stimulus

Line Operational SM is received by CS and the processing of the configuration file reveals terminals which need to be configured or an operator command is received and the line has previously been reported operational.

Action

For each type of configurable control block, the NPU keeps a descriptor string. This descriptor string has an entry for each defined field name of the control block which associates the field name to a field position in the control block (displacement, field start bit position, field bit length) and optional action sequences to be performed. Typical actions which might be defined are: validate field value, chain block to existing structure, assign a connection number and set up two-way association between connection directory and control block, clean up any existing data structure associated with field.

4.3.2 Configure Terminal SM (cont.)

Action (cont.)

The service module in the NPU will be driven by the descriptor string in inserting values in the fields and executing any associated actions. All actions are described by the NPU designer and CS is unaware of any special effects; Appendix A will show field name/field value order dependence, if any.

When the TCB has been configured, connected to its line structure and the two-way association with the network logical address has been established, a TCB Configured Response is returned to CS. Exception responses are described in Appendix A.

A TCB can be built only when a line is enabled and operational, and remains in existence until a Delete Terminal Service Message, a Disconnect or Delete Line Service message is processed.

If the line becomes inoperative prior to receipt of the Configure Terminal SM, the NPU first reports Line Inoperative then responds to the Configure Terminal with a Configure Terminal Response indicating Line Inoperative.

4.3.3 Reconfigure Terminal SM

Parameters

Port

Subport

Host Ordinal

Cluster Address

Terminal Address

Terminal class and Device type (current)

Host Ordinal

Field Name$_n$
Field Value$_n$
} One or more field name/field value pairs, i.e. the connection number. (see Table A-8)

4.3.3 Reconfigure Terminal SM (continued)

Purpose

From a Logical Connection Procedure viewpoint to change a logical connection number in an existing TCB.

Stimulus

CS detects a need to establish or change a connection or modify other values in the TCB.

Action

As for Configure Terminal SM except TCB should already exist and is modified as specified in the Service Message. A response is sent to CS.

The Reconfigure Terminal SM provides a general mechanism for CS control of terminals. Any action required coincident with the field change is also provided for by the reconfiguration mechanism.

4.3.4 Delete Terminal SM

Parameters

Port
Subport
Host Ordinal
Cluster Address
Terminal Address
Terminal Class and Device Type
Host Ordinal

Purpose

To delete a TCB which is no longer required.

Stimulus

The operator requests that a terminal be deleted.

4.3.4 Delete Terminal SM (cont.)

Action

Clean up all table and data space associated with TCB. Remove connection from logical connection directory. Respond to CS with TCB Deleted. CS is responsible for correctly deleting both ends of a connection.

4.4 Line States

Figure 4.2-1 shows the sequencing of upline and downline SM's.

4.4.1 Enable Trunk SM

Parameters

Port

Subport
Host Ordinal

Purpose

To cause a LIP to start to service the trunk.

Stimulus

NOP enables a disabled trunk.

Action

Initialize the CLA for line operation. Condition the modem for line operation.
Note Line Inoperative status if CLA or modem does not function correctly.
Note Line Operational status if all functions are performed correctly.

TABLE 4.4-1

SYSTEM CONDITION/ACTION SEEN BY TERMINALS

SYSTEM CONDITION	ACTION SEEN BY TERMINALS			
	BUSY	NO ANSWER	NPU ANSWERS	OTHER
Host Unavailable Line Configured only		X		
Host Unavailable Line configured, line enabled		X		
Host Unavailable terminal connected				NPU sends "Input Stopped" Continue polling
Recovery from Host Unavailable terminal still connected				NPU sends "Repeat La: Input"
Host Up, Line being configured		X		
Host Up, NPU overloaded			X No response to input until NPU recovers from overload	Issue "Input Stopped" to terminals at each attempt to input.
NPU Down		X		
All Ports Active	X	Disconnect existing dial-up lines.		Provided by Telephone Co. at end of rotary.

INTENTIONALLY LEFT BLANK

4.4.2 Disable Trunk SM

Parameters

Port

Subport

Host Ordinal

Purpose

To cause the LIP to stop servicing the trunk. To cause the CLA and modem to be returned to idle status.

Stimulus

Manual request to disable.

Action

If the specified Trunk exists, and is enabled, it is disabled.

4.4.2 Disable Trunk SM (cont.)

Action (cont.)

Return the trunk CLA and modem to idle status. Stop servicing the trunk.

4.4.3 Disconnect Line SM

Parameters

Port
Subport
Host Ordinal

Purpose

To combine the action of a Delete/Configure pair in one SM. Specifically designed for the switched line case. A Disconnect Line SM accomplishes exactly the same functions as a Delete Line SM immediately followed by an Configure Line SM.

Stimulus

When wishing to disconnect the current caller on a switched line and accept new calls immediately. To ensure that proper conditions are set after a failure condition.

Action

A combination of a Delete followed by a Configure. The response to this SM is identical to the Line Status Response SM (see Appendix A) except that the function codes differ. The Disconnect Line SM is also treated as a positive response to a Line Inoperative Status SM and provides protection against loss of SM's.

4.4.4 Trunk Status Request SM

Parameters

Port
Subport
Host Ordinal
} Optional

Purpose

To allow NS to request the status of any or all trunk(s).

4.4.4 Trunk Status Request SM (continued)

Stimulus

Sent when NS records are incomplete or erroneous due to CYBER Host failure.

Action

The NPU sends a Trunk Status Response SM back to the sender. If port/subport are absent, the request is for all trunks; the NPU will send one response for each configured trunk.

4.4.5 Trunk Status Response SM

Parameters

Port

Subport

Host Ordinal

Response Code

Line Type

Configuration State

Link Remote Node

Total Statuses Being Reported

Purpose

To allow the NPU to report status of a trunk.

Stimulus

Sent in response to a Trunk Status Request SM.

Action

The NPU sends the SM to the requestor.

4.4.6 Line Status Request SM

Parameters

Port
Subport
Host Ordinal } Optional

Purpose

To allow CS to request status of any or all line(s).

4.4.6 Line Status Request SM (continued)

Stimulus

Sent when CS records are incomplete or erroneous due to CYBER Host failure.

Action

The NPU sends a Line Status Response SM back to the sender. If port/subport are absent, the request is for all lines; the NPU will send one response for each configured line owned by the requesting CS.

4.4.7 Line Status Response SM

There are two basic forms to this SM: line operational and line inoperative.

4.4.7.1 Line Operational

Parameters

Port
Subport
Host Ordinal
Response code = 0
Line type
Configuration state
Number of terminals configured
{Terminal type (if unsolicited status SM) or
Total statuses reported}

Purpose

To allow the NPU to report that a line is operational.

Stimulus

Sent in response to a Line Status Request SM.

On a dial-in circuit, a Line Enabled Response is generated by the NPU immediately following a Conference Line SM. When a user dials in, the modem interface signals will then indicate an active line, at which time the NPU will generate an unsolicited Line Status - Operational SM (following auto-recognition if applicable).

Action

Upon receiving the Line Status - Operational SM, the host configures the terminals for the line by sending one or more Configure Terminal SM(s).

4.4.7.2 Line Inoperative

Parameters

Port

Subport

Host Ordinal

Response code

Line type

Configuration state

Number of terminals configured

Total statuses reported

Purpose

To allow the NPU to report that a line is inoperative.

Stimulus

Sent in response to a Line Status Request SM.

Sent as an unsolicited message whenever the TIP senses conditions causing the line to be inoperative, including normal disconnect on a dial-in line.

Line Inoperative is reported when line or modem conditions cause the line to become inoperative; it is not reported if the line is made inactive by terminating its logical connections or by disabling the line.

The following modem signal conditions cause the line to be reported inoperative. The timeouts involved insure that a line is not declared inoperative because of transient conditions which are to be normally expected.

Data Set Ready: If DSR drops at any time, DTR is immediately turned off and Line Inoperative is reported.

Clear to Send (201 and 208 modems): If CTS does not come on within one second of the rise of RTS, remain on for the duration of RTS and drop within one second of the fall of RTS, DTR is turned off (causing a switched line to disconnect) and Line Inoperative is reported. CTS is not monitored for 103/113/202 modems.

Data Carrier Detect (FDX Constant Carrier): Once line is operational, if DCD drops and remains off for a contiguous period of 10 seconds, DTR is turned off, and Line Inoperative is reported. Abnormal operation of DCD on HDX or controlled carrier lines does not influence line status.

4.4.7.2 Line Inoperative (continued)

Action

TCB's are not automatically deleted when a line becomes inoperative. The host must terminate each logical connection explicitly with a Delete Terminal SM, or implicitly by sending a Delete Line SM or a Disconnect Line SM.

4.4.8 Line Count Request SM

Parameters

None

Purpose

To allow CS to request count of configured lines it owns.

Stimulus

Sent when CS records may be incomplete or erroneous due to a failure.

Action

The NPU sends a Line Count Response SM back to the sender.

4.4.9 Line Count Response SM

Parameters

Number of configured lines.

Purpose

To allow the NPU to report the count of configured lines.

Stimulus

Sent in response to a Line Count Request SM.

Action

The NPU sends the SM to the requester.

4.4.10 Terminal Status Request SM

Parameters

Port

Subport

Host Ordinal

Purpose

To allow CS to request the status of all terminal(s) on a specified line.

Stimulus

Sent when CS records are incomplete or erroneous due to CYBER Host failure.

Action

The NPU sends one Terminal Status Response SM back to the sender for each terminal on the specified line.

4.4.11 Terminal Status Response SM

Parameters

Port

Subport

Host Ordinal

Cluster address

Terminal address

Host Ordinal

Terminal Class and Device type

Response Code

Destination node

Source node

Connection number

Total statuses being reported

Purpose

To allow the NPU to report status of any or all terminals.

Stimulus

Sent in response to a Terminal Status Request SM. Sent as an unsolicited SM when terminal failure is detected or when the terminal recovers from a failure.

4.4.11 Terminal Status Response SM (continued)

Action

When terminal failure is detected, the correspondent is informed via the logical connection (if any) and the terminal status is communicated via the Service Channel. Terminal failure does not change the state of the TCB with regard to the logical connection, nor is the state of the line, as recorded in the Line Control Block, modified. Operator action is required to delete the terminal if desired.

4.5 Regulation

Network regulation is divided into two types:

- Logical Connection Regulation
- General Network Regulation

Logical connection regulation requires that a source station not enter data into the communications network at a sustained rate greater than the destination station can absorb the data. Disparity of data rates exists between stations if a sustained flow of input from a station with a high data rate would present a large storage requirement upon the destination node which interfaces to a destination station with a slower rate of delivery. This requires that the rate of acceptance of data by a source be controllable by the rate of delivery to the destination. This facility is provided by the block protocol described in Section 2.

General network regulation requires that each node protect itself against excessive transient data storage requirements. General network regulation is accomplished by use of the following techniques:

- Logical Link Regulation
- NPU Input Regulation
- Physical Link Discard
- Station Input Discard

4.5.1 Logical Link Regulation

Logical link regulation is the regulation of total traffic in the communications network to the level which provides acceptable service to connected stations. First, the requirement is to distribute the entire load across the total capacity available for data transport on a geographical basis. After the load is shared geographically, each NPU has a direct measure of local load, and this is used to determine the advisability of admitting new data to the communications network. When an NPU determines that a buffer regulation boundary has been crossed, a Logical Link Status SM is sent to the opposite end of all logical links.

4.5.1 Logical Link Regulation (continued)

Similarly when a logical link failure is detected, the regulation level for the logical link is lowered to zero. Prior to inviting new input, each TIP checks its defined regulation level against both the current local buffer level and the regulation level of the logical link to be used for any data which may arrive at the NPU as a result of the invitation to send. Any local congestion results in feedback to neighbor nodes by regulation on incoming links which reflects this loading and in feedback to distant nodes of the network by changes in logical link regulation levels. Thus, network regulation depends on the various local algorithms for regulation and the feedback effect these algorithms have on data transfer in other NPUs. A sampling technique is used to record regulation occurrences. The NPU Statistics SM is used to report the samples. Table 4.5-1 represents a summary of the regulation measures in force at the different regulation levels.

TABLE 4.5-1 Regulation Measures/Levels

	Logical Link Regulation	NPU Input Regulation		Physical Link Discard	Station Input Discard
Regulation Level In Effect At NPU	Send Logical Link Regulation SM	Inhibit New Input; Terminal Regulation Is 2	Inhibit New Input; Terminal Regulation Is 1	Discard Frames From Trunks; Trunk Will Repeat	Reject Input Which is In Process; Terminal Will Repeat
3	Yes	No	No	No	No
2	Yes	Yes	No	No	No
1	Yes	Yes	Yes	No	No
0	Yes	Yes	Yes	Yes	Yes

4.5.2 NPU Input Regulation

The NPU software structure ensures that events are serviced in order of priority. Thus, the result of congestion in the NPU is evidenced by increasing work queues and data storage requirements, and is reflected directly by buffer availability. Regulation in the NPU consists of reference to buffer levels prior to invitation or acceptance of input. Three levels of input acceptance criteria are provided, and each TIP checks against its defined level prior to inviting input. The priority scheme for regulation is that higher speed stations are regulated first and lower speed stations last.

4.5.3 Physical Link Discard

Link discard is the action taken by an NPU when correctly received frames must be discarded in order to relieve a shortage of free buffers in the NPU. This action is taken only when the NPU input regulation and logical link regulation have failed to relieve the shortage.

4.5.4 Station Input Discard

Station input discard is the action taken, in extreme situations, when an NPU must discard already initiated input from a station in order to relieve a shortage of free buffers in the NPU. Station input discard is executed only when all other efforts have failed to relieve the shortage of free buffers. The line protocol is used to cause a repeat of the discarded input by the terminal.

4.6 User Communication

4.6.1 Host Broadcast One SM

Parameters

P, SP } Line number
Host Ordinal
CA, TA }
Terminal class and Device type
Host Ordinal
Text (1-50 characters) in IVT compatibie format. (See <DOWNLINECONTENT> in Table 9-1)

Purpose

To allow the CYBER Host to send a message to one interactive terminal user.

Stimulus

Operator type-in at the Cyber Host.

Action

The NPU will enqueue the text of the message for delivery to an interactive terminal and will send a response to the Cyber host. An Error Response will be sent if a previous Host Broadcast is not yet complete.

4.6.2 Host Broadcast All SM

Parameters

ID1 } Logical Link ID
ID2 }

Host Ordinal

Text (1-50 characters) in IVT compatible format. (See <DOWNLINECONTENT> in Table 9-1)

Purpose

To allow the CYBER Host to send a message to all console users on a logical link, or an NPU.

Stimulus

Operator type-in at the CYBER Host.

Action

The NPU will enqueue the text of the message for delivery to all interactive terminals and will send a response to the CYBER Host. An Error Response will be sent if a previous Host Broadcast is not yet complete.

4.6.3 Message to NOP

Parameters

Text (1-50 characters) in IVT compatable format. (See <UPLINECONTENT> in Table 9-1.)

Purpose

To allow an operator at an NPU console to send a message to the NOP.

Stimulus

Operator type in at the NPU console.

Action

The NPU will construct the SM and send it to NS.

4.7 NPU Overlay Control

An overlay capability is provided by the NPU Service Module which allows infrequently used functions to be written as overlays and to be loaded from the CYBER host to the NPU as required. Certain restrictions on system interfaces are imposed on accesses to global data structures and Level 1 procedures are performed indirectly via a System Interface Table (SIT). Access to global data and Level 1 procedures is thus limited to those defined by the SIT.

All overlays are required to provide the addresses of two procedures in the first two locations of their space.

The first entry is the procedure which will receive control when an Overlay Data SM from NS is received.

The second entry is the procedure to be called when the overlay space is being pre-empted for use by a higher priority function. Each overlay must provide a procedure which performs housekeeping to return all NPU space and leave all other conditions as found. The overlay may send one or more Overlay Data Response SM's prior to releasing control.

Overlay functions must not attempt to maintain any local status in an NPU from one overlay load to the next load.

The loading and control of overlays is performed by cooperation between NS, the Service Module in the NPU and the overlay. The following SM's are used in this control.

4.7.1 Overlay Program Block SM

Parameters

Block Number of this overlay block
Last Block number in overlay load
ID of overlay
Checksum

Overlay object code to be loaded in overlay space in consecutive locations.

Purpose

To install the overlay object code in the NPU prior to executing the overlay.

Stimulus

An overlay function is requested by the Network Operator or NS itself requires an overlay function.

4.7.1 Overlay Program Block SM (continued)

Action

If the overlay space is available, load the object code from the SM to the relative position in the overlay space indicated by the overlay block number. All Overlay Program Block SMs are accepted regardless of matching overlay IDs. A change of overlay ID constitutes the start of a new overlay load. Respond with an Overlay Program Block Response SM.

If the overlay space is in use, discard the SM and return an Overlay Program Block Response SM with an error indication.

4.7.2 Overlay Program Block Response SM

Parameters

Block Number

Last Block Number

Overlay ID

Purpose

To provide a positive response to each downline Overlay Program Block SM.

Stimulus

Receipt of a valid or invalid Overlay Program Block SM at an NPU.

Action

For a valid block, send the next sequential block, if any. For a rejected block, determine if a Terminate Overlay SM is to be sent.

4.7.3 Terminate Overlay SM

Parameters

None

4.7 3 Terminate Overlay SM (continued)

Purpose

To terminate the currently resident and active overlay to allow a new overlay with a higher priority to be loaded.

Stimulus

A need exists to execute an overlay function with a higher priority than the existing overlay.

Action

Call the current overlay at its Abort entry. When control is returned, return a Terminate Overlay Response SM.

No validation is performed on this SM. If an overlay is not active, the response is returned immediately.

4.7.4 Terminate Overlay Response SM

Parameters

None

Purpose

To provide positive response to a Terminate Overlay SM.

Stimulus

Receipt and execution (or no action) of a Terminate Overlay SM.

Action

Record overlay space is available and reuse if desired.

4.8 Reporting and Statistics

The NPU maintains a statistics reporting timer which causes statistics messages to be generated and sent upline at a program-build-defined frequency. Each time the timer times out, a single statistics message is generated and sent to NS/CS. Elements whose statistics are all zero are skipped.

A statistics block is maintained for the NPU itself and for each Line Control Block and Terminal Control Block. One statistics block is dumped and all counters are cleared each n seconds; in addition, a statistics block will be dumped irrespective of time period if:

- a counter overflows. The counter is set to all ones and the statistics block is dumped.
- a Line Disconnect, Delete Line, or Delete Terminal SM is received. The affected line and terminal statistics blocks are dumped prior to sending the response SM.

4.8.1 NPU Statistics SM

Parameters

Service Messages generated

Service Messages processed

Bad Service Messages received

Blocks discarded due to bad address

Blocks discarded due to bad format

Number of times no regulation

Number of times at regulation level 2

Number of times at regulation level 1

Number of times at regulation level 0

4.8.1 NPU Statistics SM (continued)

Purpose

To send statistical information accumulated by each NPU to NS.

Stimulus

Sent periodically by each NPU.

Action

The words containing the statistical counters for a particular network element are moved from the element's control block into the service message and then the control block words are cleared. When each SM arrives at the host, it is time stamped and added to the appropriate file.

4.8.2 Trunk/Line Statistics SM

Parameters

Port number
Subport number
Host Ordinal
Link remote node ID (trunks only)
Blocks transmitted
Blocks received
Characters transmitted } Good blocks only
Characters received }

Purpose/Stimulus/Action

(As for NPU Statistics SM except Line Statistics go to the owning CS.)

4.8.3 Terminal Statistics SM

Parameters

Port number
Subport number
Host Ordinal
Cluster address
Terminal address
Terminal Class and Device Type
Host Ordinal

4.8.3 Terminal Statistics SM (continued)

Parameters (con't)

Blocks transmitted

Blocks received

Blocks retransmitted

Blocks received in error

Upline Breaks (Terminal Dependent)

Purpose/Stimulus/Action

(As for NPU Statistics SM except Terminal Statistics go to the owning CS.)

4.8.4 CE Error SM

A service message is created for the occurrence of every hardware-related abnormality. This includes all NPU-related hardware such as the coupler, MLIA, loop multiplexers, CLA's, and also all connected hardware: modems, lines and terminals. The creation of the Service Message is separate from and in addition to the statistics accumulated in the NPU and periodically dumped to the host.

To prevent swamping the NPU or host with error messages when an oscillatory condition arises, an error counter is incremented with each error message generated. When the counter reaches a program-build-defined-limit, the event is discarded rather than recorded. The counter is periodically reset to zero, where the period is another program build parameter.

Parameters

The first parameter is an error report code whose value determines the content of the remaining parameters, if any, as given in Appendix A.

4.8.4 CE Error SM (continued)

Purpose

To allow each NPU program to report detected errors to NS/CS.

Stimulus

Occurrence of a program-detectable error.

Action

As errors are defined, an appropriate action will be defined also.

4.9 Software Inconsistencies

When NPU software detects an inconsistency for which no recovery action is planned, the NPU immediately halts execution, leaving a unique identifying number in the "A" register. The list of all such numbers and their interpretation will appear in a subsequent version of this document when program implementation is completed.

5.0 FAILURE/RECOVERY MECHANISMS

The actions performed on failure and recovery of various system elements are described or implied in the sections of this document which relate to those elements.

This section summarizes the failure and recovery mechanisms to provide a more effective understanding of their interrelationship.

5.1 CYBER Host Failure

The unavailability of the host will be detected at the local NPU(s). The unavailability will be communicated to the other end of logical links (local or remote) and further input will be inhibited. Where so defined, users will be informed of the host unavailability by means of system messages.

5.2 CYBER Host Recovery

After recovery, all logical links will be reinitialized and new connections will be made. Existing configuration status will be recovered by means of Status Request SM's.

The network will repeat unsolicited line status changes which remain unactioned. Most SM's sent to the network have a possibility of rejection; the rejection code allows NS/CS to determine the true state of the element for which they have incorrect records.

5.3 NPU Failure

The CYBER host may or may not be aware of the condition depending on its own state and on the availability of network paths. From a network point of view, the action is to treat the failure using the logical connection and logical link mechanisms described later.

5.4 NPU Recovery

The CYBER host will dump (optional) and reload a NPU after receiving a request for load. The stimulus for this reload comes from either the CYBER host PPU driver or the bootstrap program of the NPU. The reasons for requesting a load are:

- Software failure causes hardware deadman timer to expire
- Hardware failure causes deadman timer to expire
- Operator-initiated software halt to force reload
- Manually initiated reload by depressing Master Clear button

5.4 NPU Recovery (cont.)

The CYBER host will not request a dump after the second and subsequent attempts to reload. After 'n' successive attempts to load, the attempt is aborted and ignored until manually reactivated. After the NPU is successfully loaded and initialized, NS will set up all logical links for that NPU which the present state of the network allows.

NS will report the presence of each logical link that will be established to CS. CS will examine its configuration tables or files for elements which have been affected by the change in status. CS will configure and enable lines which are supported by the NPU. For all lines which are then reported as operational, an examination of the configuration file will reveal those terminals which can be connected. For each such terminal, both the terminal and the host support tables are configured and thereby connected.

5.5 Logical Link Suspension

A logical link suspension is detected by either the local NPU determining that the channel(s) to the host has become inactive or when the HDLC protocol on the trunk supporting the logical link fails. In the first case, the presumed host failure is communicated to the distant or local ends of all logical links. When a loss of ability to communicate is detected at the end of a logical link, all sources of data which are connected to that logical link are prohibited from accepting new data. In the case of the CYBER host being the data source, a Logical Link Regulation SM is used to inform the host of the suspension of each logical link. Interactive terminals with connections on the logical link will be informed of the suspension by INPUT STOPPED message.

5.6 Logical Link Recovery

A logical link may recover spontaneously (e.g. return to service of a failed Cyber channel) or may be reinitialized by Cyber host (NS) action. In the case of spontaneous recovery, the Logical Link Protocol allows a restart without loss of data. Otherwise, all logical connections must be re-made and the session must restart. Restart facilities are application dependent.

5.7 Trunk Failure

A failure of a trunk is detected by failure of the protocol. At this time, data in queue for the trunk is discarded. A trunk failure will cause the NPU to report the failure of the logical link supported by the trunk.

5.8 Trunk Recovery

Recovery of a trunk is always detected by the trunk protocol. The logical link protocol will determine when the trunk may be used for data other than SM's to/from NS.

5.9 Line Failure

A line failure is detected by abnormal modem status or line protocol failure. The change of status is reported to CS. CS will delete all TCB's supported by the line via the disconnect line SM.

5.10 Line Recovery

A line cannot recover from a failure spontaneously. It is necessary for CS to first action the Line Inoperative Status SM by deleting the supported TCB's and disabling and then enabling the line. At this time, the TIP will commence to monitor for a change in status. When the line status changes to operational, this is reported to CS. CS, receiving a line status change to operational, will attempt to configure the supported terminals as previously described.

5.11 Terminal Failure

Where the protocol is capable of determining terminal status, the protocol will maintain records of such status. Terminal failure status will be reported to CS for network management purposes. The correspondent to which the terminal is logically connected will be informed of the failure by the Block Protocol (STP).

Undeliverable traffic will be discarded. The logical connection will not be broken on terminal failure.

5.12 Terminal Recovery

When terminal failure is detected, terminal recovery will be monitored. This will typically be by performing a periodic status or diagnostic poll. Terminal recovery status will be reported to CS and the logically connected correspondent will be informed by the use of the Block Protocol (STRT).

6.0 HOST INTERFACE PROGRAM

The general method used to interface the host is by a Host Interface Program (HIP). A HIP is the NPU program which controls data transport between the NPU and the host. A HIP, along with a CYBER coupler, implements the CYBER host protocol.

The CYBER coupler is used to connect the CYBER host to a front-end NPU. The CYBER coupler is not programmable. The coupler access mode of the CYBER coupler is Direct Memory Access (DMA).

Host Regulation

The primary objective of host regulation is to provide a vehicle to control the following:

- prevent saturation or overloading of the host or network in the event of an abnormality; i.e., emergency regulation
- allow data flow between the network and the host to ensure that continuity of service and performance standards are maintained
- smooth data flow (prevent over-regulation) using appropriate feedback control techniques

The host is totally governed by the various regulation functions described elsewhere. Regulation control for the host is effected through mechanisms designed into the host coupler interface and the block protocol. The host coupler interface is controlled, variable bandwidth I/O channel in which the bandwidth will be increased or decreased as a function of load balancing, regulation thresholds reached (e.g., buffer utilization, CPU utilization), etc.

6.1 CYBER Host Interface Program

6.1.1 CYBER Coupler Hardware Programming Description

The CYBER Coupler is the hardware interface between a PPU of a CYBER 70/170 and an NPU. A PPU may interface one or two couplers on the same channel. An NPU interfaces one or two couplers (but only one coupler may be connected to any host).

The coupler has essentially three transmission circuits:

1) A half-duplex data circuit for transmission of programs or data between the memory of the PPU and the memory of the NPU.

2) A full-duplex control circuit via which the NPU and the PPU perform transaction setup "handshaking".

3) A supervisory circuit via which transaction status is monitored.

6.1.1 CYBER Coupler Hardware Programming Description (cont'd)

The coupler also provides an execution control circuit via which the PPU can start or stop NPU microprogram execution, or reset the microinstruction address counter.

6.1.1.1 Registers (See Figure 6-1)

Those coupler registers directly accessed by the PPU program for normal data transmission are:

1) Coupler Status Register - a group of 16 hardware defined flags (the low order twelve bits of which can be read by the PPU) which identify to the NPU the reason for interrupt, and which indicate to the NPU and PPU transaction and register status.

2) NPU Order Word - a 16-bit register, the lower order twelve bits of which are written by the PPU to communicate to the NPU a software-defined order code.

3) NPU Status Word - a 16-bit register (the low order twelve bits of which can be read by the PPU) by which the NPU communicates to the PPU a software-defined status code.

4) NPU Address Register - this is a 17-bit register, all bits of which can be written by the PPU, for the purpose of loading or dumping the NPU. The high order 9 bits (Address Register bits 16-8) are called Memory Address Zero. The low order 8 bits (Address Register bits 7-0) are called Memory Address One. The PPU must function the coupler twice to write the entire register. The high order bit of the Address Register (bit 16) is actually implemented as bit 8 of the NPU Status word, which can therefore not be used for other purposes.

6.1.1.1.1 Coupler Status Register (See Table 6-1)

Table 6-1. Coupler Status Register

Bit Number	I/A	Flag Name	SET Condition	RESET Condition
0	A	Memory Parity Error	NPU Memory Parity Error	*
1	A	Memory Protect Fault	NPU Memory Protect Fault	*
2	-	NPU Status Word Loaded	NPU Writes Status Word	PPU reads NPU Status Word **
3	-	Memory Address Register Loaded	PPU or NPU Writes Memory Address One	---
4	I	External Cabinet Alarm	Power Failure	*
5	I	Transmission Complete	PPU completes any Input or Output operation	*
6	I	Transfer Terminated by NPU	NPU Terminates Transfer (not used)	*
7	I	Transfer Terminated by PPU	PPU sets channel inactive during Data I/O	*
8	I	Order Word Register Loaded	PPU writes Order Word	NPU reads Order Word
9	-	NPU Status Read	PPU reads NPU Status Word	*
10	I	Timeout	Inactive returned during a PPU Data I/O operation because coupler was selected and active for more than 3 seconds.	*
11	A	CYBER 170 Channel Parity Error	12-bit word plus parity from data channel not odd parity and Enable Parity Switch on.	Enable * Parity Switch positive transition
12-13		Unused		
14		Chain Address Zero	Coupler finds zero in last word of NPU buffer.	*
15	-	Alarm	Positive Transition of any Flag marked "A"	*

All flags (** except bit 2) are reset when NPU or PPU Clears the Coupler. Those flags marked with * are also cleared when the NPU reads the Coupler Status Register. All flags are cleared by Master Clear.
I/A: I = Raising Flag causes NPU Interrupt; A = Raising Flag causes Alarm.

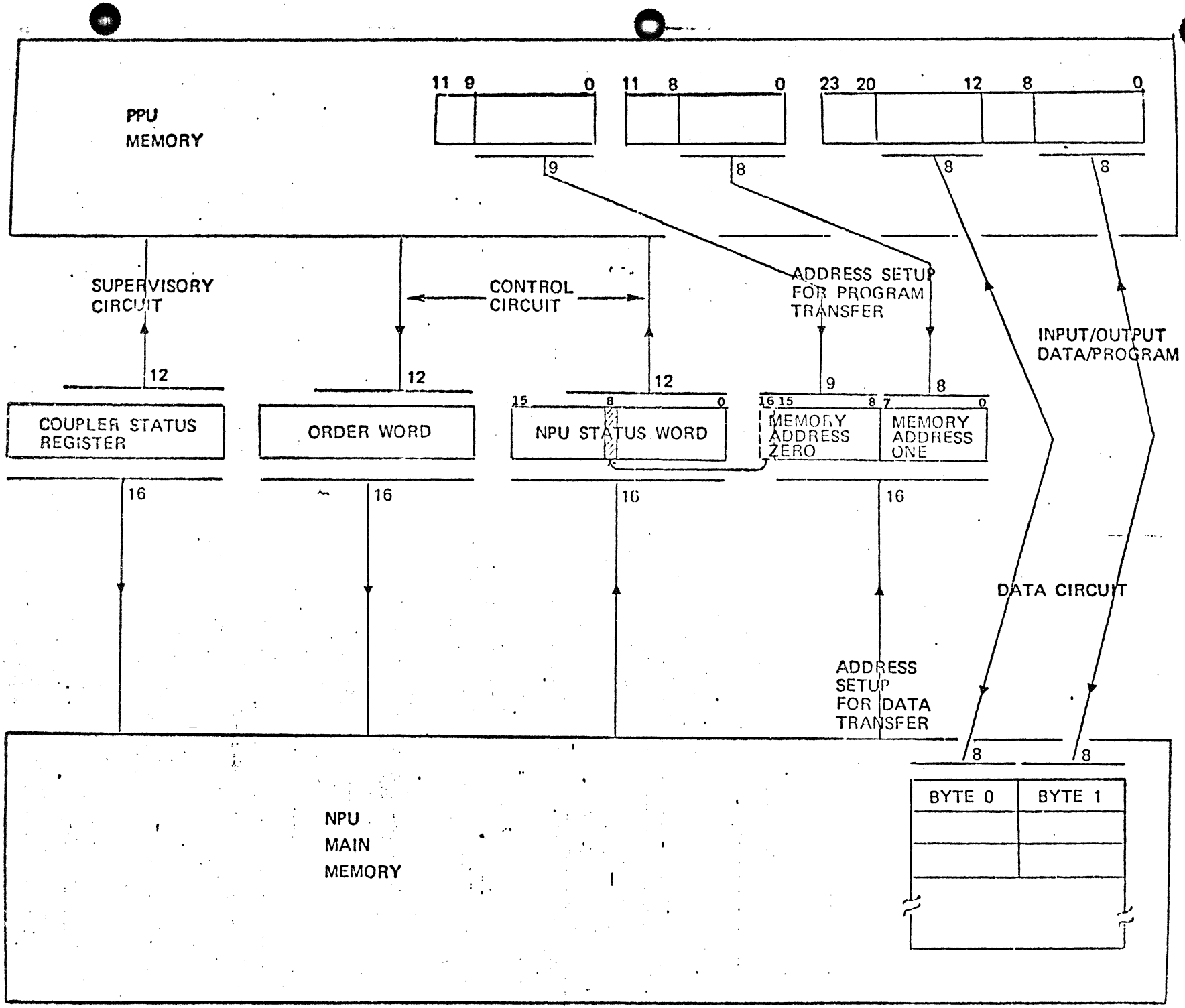
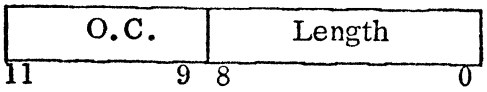


Figure 6-1. CYBER Coupler Registers

6.1.1.1.2 Order Codes

Four order codes are defined for the data transfer protocol. See Section 6.1.2 for description of the interpretation of these codes.



OC - Order Code (See Table below)

Order Code Value	Name	Regulation Level
1	Output Level 1 (Service Messages)	1
2	Output Level 2 (High Priority Data)	2
3	Output Level 3 (Low Priority Data)	3

Length - In 8 byte increments, of the output block to be transferred. The value is rounded up when the length is not a multiple of 8.

6.1.1.1.3 Status Codes

Six status codes are defined for the data transfer protocol. See Section 6.1.2 for interpretation of the codes.

Code Value	Name
0	Ignore value and read again
1	Idle
4	Ready for Output
7	Not Ready for Output
13	Input Available, $\leq$ 256 bytes
14	Input Available, > 256 bytes

Another status code is used only for the dump protocol. See Section 6.1.2

Code Value	Name
8	Ready for Dump

6.1.1.2 Data Formats

In two of the four coupler operational modes (see 6.1.1.3) data is transferred between the coupler and the memory of the NPU via a Direct Memory Access (DMA) port. The DMA port transfers 16-bit memory words, but the PPU transfers 12-bit words to/from the coupler.

6.1.1.2.1 Load/Dump Word Format

The PPU must transfer an even number of PPU words. The first word of a pair of words transferred by the PPU corresponds to bits 15-8 of the NPU word (Byte 0). The low order eight bits of the second word of the pair transferred by the PPU corresponds to bits 7-0 of the NPU word (Byte 1). The high order four bits of the PPU words are not transferred to the NPU. When transferring from the NPU, the coupler sets the high order four bits of the PPU words to zero.

6.1.1.2.2 Buffer Word Format

When executing the Data Transfer Protocol, an arbitrary number of characters are transferred between contiguous locations in the PPU and a set of chained buffers in the NPU. The location of the characters in NPU memory, and the operation of the buffer chaining mechanism are transparent to the PPU.

From the point of view of both NPU and PPU, input means data flowing upline, that is, from NPU to PPU. Similarly, output means data flowing downline, from PPU to NPU.

The high order four bits of each PPU data word control the operation of the output transaction, although bits 10-8 are not used in the defined protocol, and are always set to zero. (If any of bits 10-8 are set, this forces buffer chaining at other than end-of-buffer in the NPU, and will cause excessive buffer use in the NPU). Bit 11 is set to one on the last character of the transaction, and causes the coupler to stop storing data into NPU memory. When reading input, the last word transferred to the PPU will have bit 11 set to 1.

6.1.1.3 Modes of Operation

6.1.1.3.1 NPU Control

The PPU can stop the execution of the NPU microprogram and set the micro-instruction counter to zero, or start the microprogram execution.

6.1.1.3.2 Load/Dump

To load or dump the main memory of the NPU, the PPU must first specify a start location by writing Memory Address Zero, and Memory Address One. It then performs successive data transfers. The first pair of PPU words transferred corresponds to the contents of the specified NPU main memory address. The NPU memory address register is then automatically incremented by one, such that successive word pair transfers correspond to the contents of successively higher numbered NPU main memory locations. The memory load or dump is terminated when the PPU sets the channel inactive.

6.1.1.3.2 Load/Dump (cont'd)

The PPU will read back memory contents to verify the load of each module prior to issuing the Start NPU code. If the load does not verify, the PPU will retry the load three times before alarming the host operator.

6.1.1.3.3 Single Word Transfer (Control)

The NPU can write the NPU Status Word at any time. The PPU can read the NPU Status Word only if it has been loaded by the NPU. When the PPU reads the register, it must not read the register again until the NPU again writes the register. The PPU determines that the NPU Status Word has been loaded (written) by interrogating bit 2 of the Coupler Status Register. This bit is automatically reset when the PPU reads the NPU Status Word.

The PPU can write the Order Word at any time. The NPU will read the Order Word only if it has been loaded by the PPU, as indicated by bit 8 of the Coupler Status Register. This bit is automatically reset when the NPU reads the Order Word.

6.1.1.3.4 Multiple Character Data Transfer (Block Transfer)

This is the only mode of operation of the coupler which requires the cooperation of both NPU and PPU to achieve. Either the NPU or the PPU may initiate the operation. When both have completed the setup, the transfer takes place.

The PPU must function the coupler to either Input Data, or Output Data. For Output, there is no way by which the PPU can then directly determine if the NPU has set up its side of the coupler to transfer the data. This can only be accomplished via preceding communications by which the NPU and PPU agree that setup for output will be the next thing done by both sides. For input, once the PPU has functioned the coupler and activated the channel, it can test the channel to determine if it is full. If it is, the NPU has already set up. If the channel is empty, the NPU has not set up the coupler. In the protocol specified, the latter is not permitted; thus, if the channel full test fails after functioning the coupler for input, a failure exists. The channel should become full within 12 μs of functioning the coupler for input.

The NPU sets up its side of the coupler for data transfer by writing the Buffer Length Register (not used by the PPU), then writing the address of the first buffer of a chain to the NPU Address Register.

Output transfer is terminated by the presence of bit 11 in any character in the PPU output data stream. The PPU must disconnect the channel following transfer of this word.

Input transfer is terminated when the last character of an NPU buffer is transmitted, and when bit 11 in the last word of the buffer has a value of one. The last character transferred will be stored in PPU memory with bit 11 on. The coupler automatically disconnects the channel after this word is transferred.

Table 6-2. PPU FUNCTION CODES

PPU Function Code	Octal Value	PPU Usage
Clear NPU	200	Used prior to loading or dumping the NPU. Stops the NPU and sets μ-memory address register to location 0.
Start NPU*	040	Starts the NPU emulator (μ-code) at the location in the μ-memory address register. The emulator must always be started at location 0.
Input Program	007	Used to dump NPU main memory.
Output Program	015	Used analogously to Input Program to load the NPU main memory. μ-memory can neither be loaded nor dumped from the PPU.
Clear Coupler	400	Resets the coupler's control logic and most registers. The protocol defined herein allows only the NPU to clear the coupler.
Output Memory Address Zero and One	010 011	Sets NPU main memory accessing for loading and dumping.
Output Order Word	016	Load the Coupler Order Word Register. Causes an NPU interrupt.
Input Coupler Status	005	Used to check the state of various registers and flip-flops in the coupler. It is used to test whether the NPU has loaded the NPU Status Word.
Input NPU Status	004	Inputs the NPU Status Word previously loaded by the NPU.
Input Order Word	006	Allows the PPU to read back the Order Word it had written. Used only prior to dumping the NPU.
Input Data	003	Allows characters to be input to the PPU. The coupler must have been previously set up by the NPU.
Output Data	014	Allow characters to be output from the PPU. The coupler must have been previously set up by the NPU.

* Must be delayed at least 10 ms following a Clear NPU Function Code.

6.1.1.4 PPU Interface

The only other situation in which the coupler automatically disconnects the channel is when the PPU functions the coupler. The disconnect will occur within one microsecond of executing the function code. If a parity error is detected on the function code (CYBER 170) the channel will not be disconnected. The coupler function code occupies the low order nine bits of the 12-bit PPU function code. The high order 3 bits of this PPU word contain the equipment code (coupler address on the channel). The equipment code is determined by the setting of switches on the coupler.

The coupler is programmed from the PPU side by setting a function code, then executing an I/O Instruction. See Table 6-2.

Load/Dump and Multiple Character Data Transfer take place at a maximum instantaneous rate of one PPU word per microsecond. The actual instantaneous rate may be lower as transfer to/from NPU memory may encounter DMA contention;however, such delays are unlikely to exceed a couple of microseconds per character, and will happen with very low frequency.

6.1.1.5 NPU Interface

The coupler is programmed from the NPU side by issuing commands over the Internal Data Channel (IDC). The IDC is also used to read the order word and write the status word. Data block I/O takes place via Direct Memory Access (DMA). See Table 6-3 for a list of the NPU Commands.

6.1.2 Data Transfer Physical Protocol

The Data Transfer Physical Protocol performs data transfer and error checking. Errors are of three types:

- Contaminated Data
- Incomplete Transaction
- Failure of Interface to Respond

The first two types of errors are handled at the physical protocol level by accepting only good blocks, and discarding bad blocks in their entirety. The physical level protocol does not perform transmission retry or attempt recovery of lost blocks.

Interface failure causes the interface to be declared down, but the protocol returns to the initial state and continues to wait for interface response.

Since the coupler is assumed to provide a noise-free channel and have only hard rather than intermittent failure modes, a logical level protocol, whose purpose is recovery of lost blocks, is not proposed at this time, however errors are detected and logged by the host. A logical level protocol may be incorporated at a later time if operational failure data demonstrates the utility of such a procedure.

The physical protocol is described by a pair of state diagrams: one showing operation of the PPU (Figure 6-2) and the other showing operation of the NPU (Figure 6-3).

6.1.2 Data Transfer Physical Protocol (cont'd)

The coupler can perform block mode transfer in only one direction at a time. Therefore, the protocol is half duplex. NPU and PPU independently bid for the channel. The NPU bids by commanding "Output Memory Address" to point to the start of the INPUT block buffer chain and then commanding "Output NPU Status" with one of the INPUT AVAILABLE status codes. The PPU bids for the channel by functioning the coupler to "Output Order Word" with one of the "OUTPUT" codes. If both the PPU and NPU bid for the channel at approximately the same time, the NPU normally allows the output by changing the value in the coupler's memory address register to point to an output buffer chain and responding with "READY FOR OUTPUT" in the NPU status word. The NPU will then rebid for the channel at the completion of the output transaction.

The NPU receives an interrupt when the PPU writes the Order Word, reads the NPU Status Word, or completes a data transfer. The Coupler Status Register indicates to the NPU the reason for interrupt. Therefore, the PPU does not separately indicate via the control circuit that the transaction is complete - this information is automatically available via the supervisory circuit.

6.1.2.1 Timers

Timers are used in five ways by this protocol. Both the PPU and the NPU have timers implemented locally.

6.1.2.1.1 Failure Detection

Three timers are used to accomplish failure detection. A Keep Alive Timer of one second is used by the NPU to provide periodic IDLE status to the PPU when no traffic is in progress. The PPU has a Dead Timer of 10 seconds. This timer times out only if the PPU misses an IDLE or INPUT request, at which point the PPU declares the NPU to be dead, and enters the NPU dump/reload sequence.

The NPU also has a Dead Timer of 10 seconds. If the NPU fails to get a coupler interrupt within this period, it declares the host unavailable. The NPU Dead Timer is not explicitly shown on Figure 6-3, but is implicit in the notation, as explained in Section 6.1.2.2.

6.1.2.1.2 Contention Resolution and Hog Control

If the NPU and PPU both signal requests to use the channel at approximately the same time, the contention is resolved in favor of the PPU by permitting output. When the output transaction completes, the PPU starts a brief (1-10 ms) Output Continue Timer to allow the NPU to request INPUT if it has data queued for the PPU. This timer prevents the PPU from monopolizing the channel with output and flooding the NPU. If, however, the NPU does encounter a scarcity of buffers, it will reject the PPU's request to output. To regulate the rate at which the PPU bids for the channel when this situation arises, an Output Rejected Timer of 100 ms is used. This limits the frequency of coupler interrupts to the NPU when the NPU is short of buffers.

Table 6-3. NPU COMMAND CODES

NPU Command	Hex Value	NPU Usage
Input Switch Status	0654	Allows the NPU to check PPU data channel device address, on-line/off-line switch setting, alarm override switch setting, etc. Executed during initialization.
Output Buffer Length	0658	Sets the coupler to follow NPU buffer chains for the current buffer length in use. Executed during initialization.
Clear Coupler	060C	Resets the coupler control logic and most registers. Used during protocol error processing. The contents of the NPU Status Word is not affected.
Input Coupler Status	0650	Used in NPU interrupt handler to determine the reason for interrupt.
Input Order Word	0660	Used in NPU interrupt handler to input Order Word previously loaded by PPU.
Output NPU Status	0648	Used to send control codes to the PPU.
Output Memory Address	066C	Used to set up the coupler for data transfer. Points the coupler to the start of an NPU buffer chain.

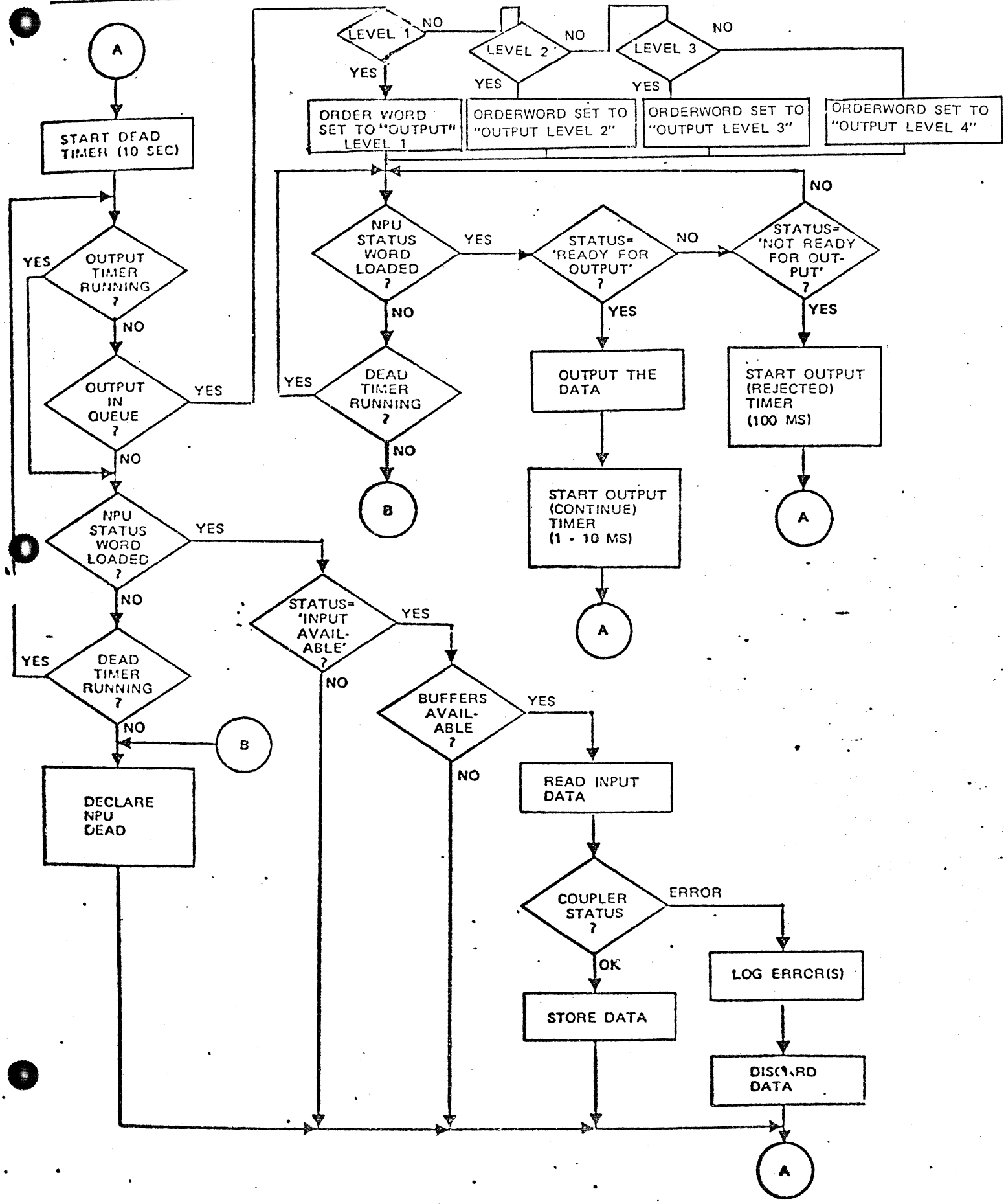


Figure 4-4. Data Transfer Protocol - PDN Software

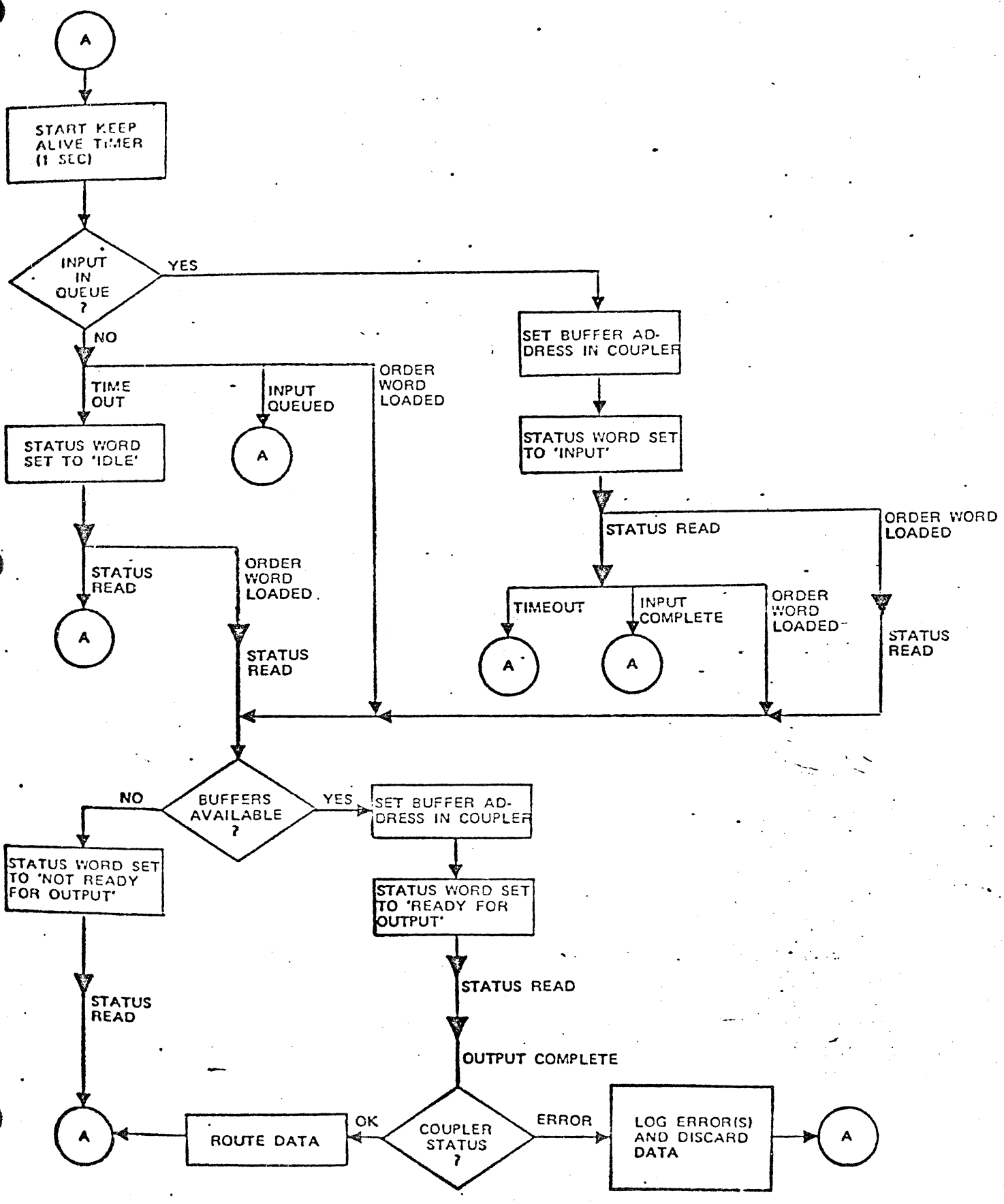


Figure 6-3. Data Transfer Protocol - NPU Sequence

6.1.2.1.3 Priority Regulation and Buffer Management

Data passing through the coupler in upline and downline directions are classified by the NPU and PPU, respectively, as follows:

Upline (One common queue in NPU)

1. Data and Supervision less than 256 bytes in length
2. Data and Supervision greater than 256 bytes in length

Downline (Separate queues in Host)

1. Service Messages.
2. Data Blocks and related Forward and Reverse Supervision of the highest priority.
3. Data Blocks and related Forward and Reverse Supervision at the lowest priority.

The NPU will accept all output offered by the PPU in normal operation. When the NPU goes into regulation due to buffer levels dropping below pre-determined thresholds, it will reject output offered at Level 3, then Levels 3 and 2 and finally, in an extreme situation, all output offered by the PPU. As buffer levels rise above these regulation thresholds, the NPU will reverse this procedure until it is again capable of accepting all inputs.

The order in which the PPU offers the various output Levels will be determined by host considerations.

There is no priority associated with the two upline queues offered by the NPU to the PPU; the separation into two length ranges is only to allow the PPU to utilize its buffer space more efficiently.

6.1.2.2 Description of the State Diagrams

A larger arrow is used in some places on Figure 6-3. Such an arrow indicates the point at which the NPU waits for the next coupler interrupt. While waiting, the coupler program re-entry point is saved in a state vector, the deadtimer is running and the NPU is servicing other processes. When the interrupt occurs, the NPU resumes service of the coupler at the location specified by the state vector. If the reason for interrupt is one of the items listed below the arrow, service proceeds as shown. If the interrupt occurred for some other reason, an error has occurred. Such an error is logged to the CE error file and the protocol is restarted at "A". If the deadtimer timeout occurs before the interrupt, the HIP calls a routine to note Host Unavailable - see Section 6.1.3 and restart the protocol at "A".

6.1.2.2 Description of the State Diagrams (cont'd)

The principal features of the protocol detailed by the state diagrams are:

- The NPU can specify INPUT AVAILABLE and set up the coupler for input data transfer at any time.
- The PPU can order OUTPUT at any time.
- If conflict occurs, the NPU will normally allow output.
- The NPU can refuse to take output if it does not have buffer space.
- The PPU can refuse input from one or both NPU queues simply by ignoring INPUT AVAILABLE.
- If either dead timer times out, protocol is reset to the start condition, but continues.
- If a given output type is refused by the NPU, the PPU performs a short timeout before re-requesting output, to prevent swamping the NPU with interrupts. The type of output offered in succeeding attempts is determined by host logic.
- If output is accepted by the NPU, the PPU allows the NPU to indicate if input is available before again ordering output.
- Once data transfer is initiated, the transaction must complete, or the entire transaction unit must be discarded.
- Error checking is performed by the receiver. If an error is detected, it is logged to the CE error file, the data received is discarded, the protocol is reset, and retry is not attempted.

6.1.3 Host Failure and Recovery

When the NPU software determines that communications across the coupler has failed, a Regulation level of zero will be communicated to the other end of each Logical link terminating at the coupler. This will inhibit acceptance of further input traffic from terminals logically connected via the coupler. Additionally, an informative message will be sent out to each affected interactive terminal.

When the NPU software determines that communication across the couplers has been restored, a normal Regulation level will be communicated to the other end of each logical link terminating at the coupler. This will remove the inhibitions on input from terminals logically connected via the coupler and cause an informative message to be sent to all affected interactive terminals.

7.0 LINK INTERFACE PROGRAM

The Link Interface Program (LIP) provides the total control of information transfer between NPU and each of its neighbors. A physical link between two NPU's consists of one or more trunks providing the required bandwidth.

The functions performed by the LIP are:

(a) Implement a class of the Control Data Corporation Control Procedure (CDCCP) for information interchange. CDCCP treats each trunk as a separate entity and is not concerned with either the contents of the information frame or of the relationship between the trunks which constitute a link. The specific protocol implemented is equivalent to ISO HDLC class, Symmetrical, Asynchronous Response Mode, Basic Numbering Range with two way simultaneous reject and initialization options (SAB, 2, 5).

(b) Creation of an information frame on demand by a trunk by selecting blocks or sub-blocks from the link queue using a pre-emptive priority resume dispatching scheme.

(c) Restructuring of the blocks from the blocks and sub-blocks received in the information frames.

(d) Support the use of trunks as a method of downline loading a failed NPU using a degenerate form of the CDCCP protocol.

(e) Provide a control and reporting mechanism for trunk statistics and status to support network management and visibility.

(f) Provide a physical link protocol which recognizes a trunk failure at both ends of the link, and which is cause to fail the associated logical link.

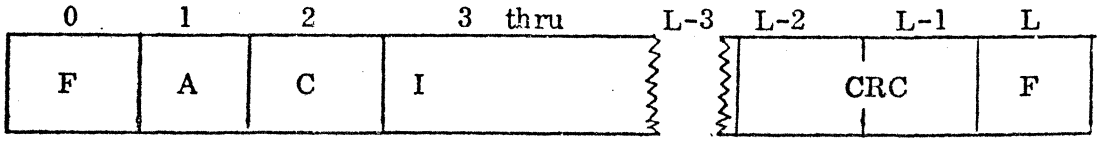
(g) For this release, block ordering is maintained during delivery by restricting links to only one trunk.

7.1 CDCCP LIP

The CDCCP LIP supports CDCCP in the full duplex, point-to-point, asynchronous response mode. Error detection is effected using transmission frame sequencing and Cyclic Redundancy Checking (CRC).

7.1.1 Transmission Frame Format

The Trunk Transmission Frame (TTF) fields are as follows:

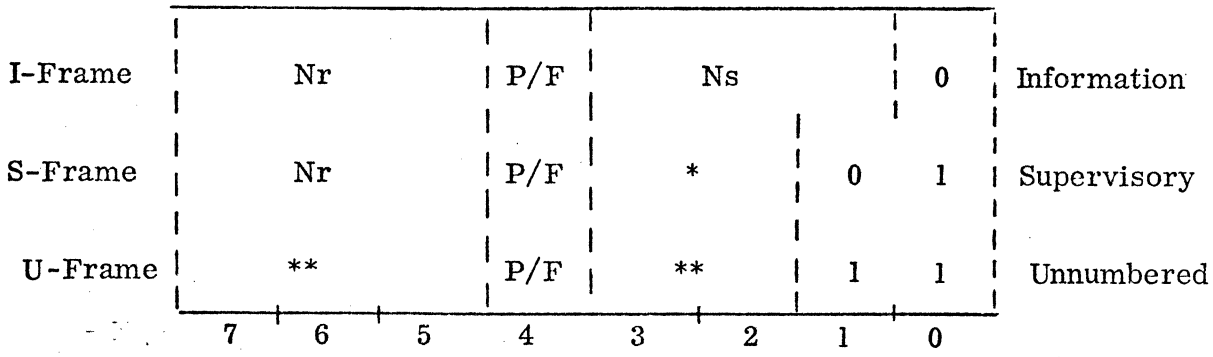


F Flag is a unique bit pattern (01111110) to identify the start and end of a TTF. The uniqueness is preserved by inserting a zero bit after every string of five ones when a frame is transmitted and removing the zero when the frame is received.

7.1.1 Transmission Frame Format (cont'd)

A Address is the secondary function identifier (1 for a local NPU, 0 for a remote NPU).

C Control is the field containing the commands and responses necessary to control a trunk. Three formats are available for the control field: Information, Supervisory and unnumbered.



Note: O bit is first bit transmitted.

N (R) Sequence Number Next Expected

N (S) Sequence Number of this Frame

P/F Poll/Final Flag

* Supervisory Commands/Responses

00 Receive Ready

01 Receive Not Ready

10 Reject

** Unnumbered Commands/Response

Binary Value (bits 2-7 of C field)

000P00	Unnumbered Information	UI
000F01	Request Initialization Mode	RIM
000P01	Set Initialization Mode	SIM
010P00	Disconnect	DISC
011F00	Unnumbered Acknowledgement	UA
100F01	Command Rejected	FRMR
000P11	Set Asynchronous Response Mode	SARM
000F11	Disconnect Mode Response	DM

7.1.1 Transmission Frame Format (cont'd)

I The Information field contains the data transferred over the link. The I field is allowed only in a frame having a C field of Information Transfer format or unnumbered information.

CRC The Cyclic Redundancy Check field is a 16 bit result of mathematical computation on the digital value of all bits in the frame (excluding inserted 0's). The transmitter performs the calculation and sends the result. The receiver performs the calculation and compares the result with the CRC received. If the comparison fails, the frame is discarded and must be retransmitted.

The CLA uses CRC procedures to determine the integrity of the incoming frame. The CRC field is the binary pattern found by multiplying the binary value of the A, C, and I fields by X^{16} and dividing the result by $X^{16} + X^{12} + X^5 + 1$. If, at the end of the received frame, the CRC field does not equal the calculated value of this remainder, the frame check sequence error FCSE status is sent to the controlling processor.

7.1.2 Normal Protocol

ISO HDLC protocol class SAB, 2, 5 defines a primary-to-secondary function coupling for each enabled trunk and each direction of transmission. With this symmetric system, either end of the link may initiate data transmission when conditions warrant. The Poll/Final flag in the control byte will not be used in information frames.

FRQ Frame Retention Queue - An eight entry list for each trunk. As an information frame is transmitted, it is entered into the FRQ according to its transmission sequence number. When acknowledged, the frames are released from the FRQ. Frames are retransmitted from the FRQ as necessary starting with the oldest first.

7.1.2.1 Transmit

An I frame is prepared for transmission as described in 7.2

The I-frame header consists of the address byte and the control byte. The address byte is the node ID of the node at the other end of the link. The sequence number of this frame N(s) is placed in the control byte. This defines the slot in the frame Retention Queue where the pointer to this frame will be stored.

7.1.2.1 Transmit (Con't)

Supervisory frames (S-frames) are transmitted under the following conditions:

- Receive Ready (RR) -
  1) An I-frame is received correctly.
  2) In response to an RNR command received with the P/F flag set and regulation not in effect.
- Receive Not Ready (RNR)
  1) Buffer levels require that trunk inputs be regulated.
  2) In response to any I-frame or RNR command received with regulation in effect.
- Reject (REJ)
  1) An I-frame is received without error but the sequence number N (s) is not the one expected. The received frame is discarded and a REJ frame is sent. All subsequent I-frames are discarded until the expected frame is received.

Unnumbered frames (UI/UA) are used to initialize and give error responses in ARM.

- Unnumbered Acknowledgement (UA) is sent when a Set Asynchronous Response Mode (SARM) command is received.
- Set Initialization Mode (SIM) is sent when a Request for Initialization Mode (RIM) is received.
- Frame Reject (FRMR) is sent whenever a frame, whose control field is not defined in Section 7.1.1, is received.
- Disconnect (DISC) is sent when a Disable Trunk Service Message is received by the NPU.
- Disconnect Mode (DM) is sent in response to a disconnect.

7.1.2.2 Receive

Frames received from a neighbor are processed according to type. I-frames contain information. The information is processed as described in 7.2. S-frames contain acknowledgements and may interrupt the flow or cause retransmission. Unnumbered frames indicate initialization is needed or an error has occurred and are processed by the LIP as necessary.

Acknowledgements come across the trunk in the control byte of an S-frame. The number N(R) is the neighbor's next expected number for the trunk. Thus all frames up to and including N (R)-1 that are saved in the Frame Retention Queue may be released. Failure to receive an acknowledgement after a suitable timeout will cause the transmitter to poll for an acknowledgement. If the acknowledgement does not allow all frames to be released from the FRQ, those remaining are re-transmitted. If repeated polls do not receive an acknowledgement, the trunk is declared inoperative.

An RNR command received with the Poll/Final flag set will cause a response to be sent as soon as possible. The response will be an S-frame RR or an RNR if regulation is in effect.

Supervisory functions performed when the following S-frame responses are received:

- Receive Ready (RR) - Acknowledges frames as described above.
- Receive Not Ready (RNR) - The receiving primary inhibits further I-frame transmission over the trunk. An S-frame RNR with the P flag set will be sent to inquire if the secondary can again receive I-frames. The trunk will be declared inoperative if after several inquiries the neighbor is not ready to receive.
- Reject (REJ) - After the acknowledgement contained in the REJ S-frame is processed, all frames remaining in the FRQ are re-transmitted starting with the oldest frame.

Certain unnumbered commands/responses may be received during the normal protocol. Any event not mentioned will cause a Command Reject (CMDR) to be sent.

- Request Initialization Mode (RIM) - This indicates the neighbor NPU has failed and the load/dump process is to be initiated.
- Disconnect (DISC) - The LIP goes to the disconnected state (ADM).

7.1.2.2 Receive (Con't)

- Command Rejected (CMDR) - The Information field contains the reason the command was rejected. The event is noted in statistics and the trunk is re-initialized using the SARM-UA handshake procedure.

- Set Asynchronous Response Mode (SARM) - Any current frame going out is aborted if possible. An Unnumbered Acknowledgement (UA) is immediately transmitted on the trunk. Any frames remaining in the FRQ are renumbered giving the oldest frame "zero", the next oldest "one", and so on. The frames can then be retransmitted.

7.2 Queue Dispatch Discipline

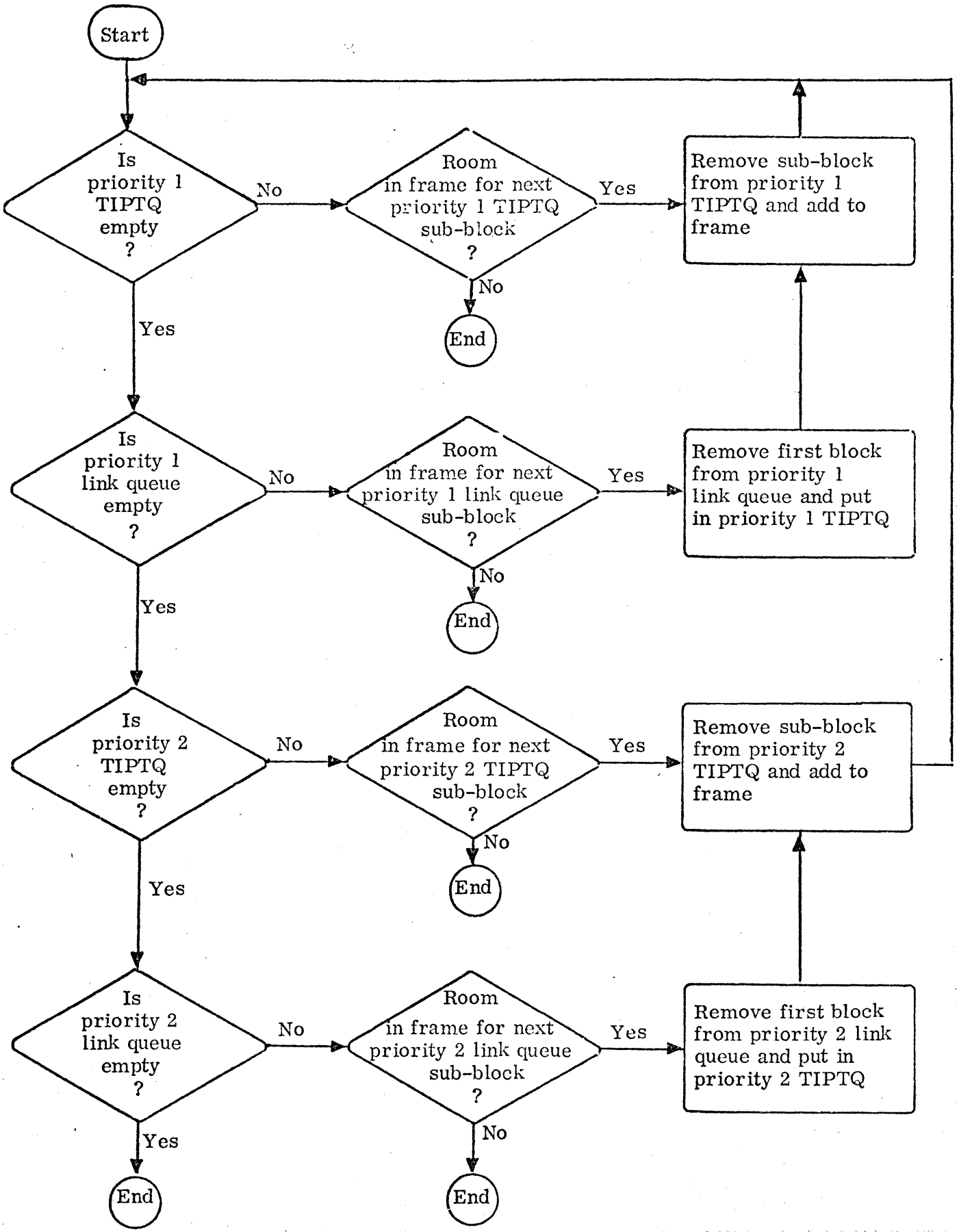
Blocks for internodal delivery are queued by link and, within link, according to priority. These queues are input to the LIP. Priority one is for high priority traffic; priority two is for low priority traffic. The LIP interrupts low priority traffic delivery at frame boundaries in order to deliver queued high priority traffic. This in conjunction with an appropriately small frame size optimizes high priority response.

Information frames are constructed as a concatenation of sub-blocks whose total length does not exceed a defined maximum frame size. A sub-block is defined to have a relatively short maximum length, and may be all or part of a block. By not requiring frames to end on a block boundary, frames of fairly constant length are constructed whenever a sufficient number of sub-blocks are awaiting transmission.

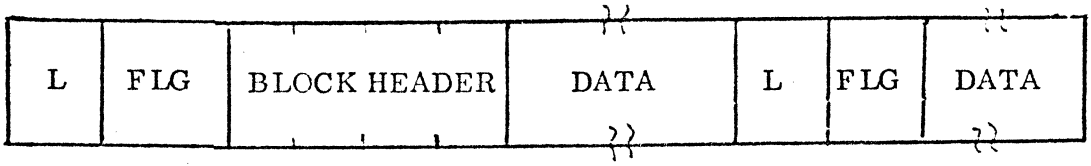
Each trunk of a link has a Transmit In-Process Text Queue (TIPTQ) for each priority. If not empty, TITTQ contains the untransmitted remainder of a block which has been removed from the link queue and partially transmitted on the trunk.

The LIP uses the logic shown in the following flow chart to select sub-blocks for the next information frame to be transmitted on a particular trunk.

Each frame is headed by the A and C fields defined above. Each sub-block in the frame is headed by (1) an L field containing the length in characters of the sub-block following, and (2) an FLG field containing a priority flag and an end-of-block flag. The L and FLG fields are used by the receiving LIP to restructure the original blocks for processing by the CCP.

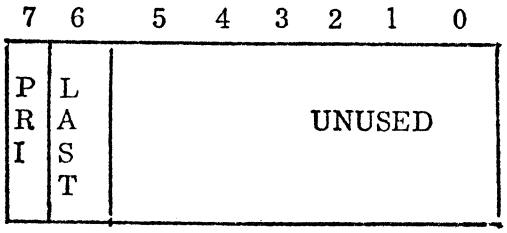

Start
Is priority 1 TIPTQ empty ?
No
Room in frame for next priority 1 TIPTQ sub-block ?
Yes
Remove sub-block from priority 1 TIPTQ and add to frame
No
End
Yes
Is priority 1 link queue empty ?
No
Room in frame for next priority 1 link queue sub-block ?
Yes
Remove first block from priority 1 link queue and put in priority 1 TIPTQ
No
End
Yes
Is priority 2 TIPTQ empty ?
No
Room in frame for next priority 2 TIPTQ sub-block ?
Yes
Remove sub-block from priority 2 TIPTQ and add to frame
No
End
Yes
Is priority 2 link queue empty ?
No
Room in frame for next priority 2 link queue sub-block ?
Yes
Remove first block from priority 2 link queue and put in priority 2 TIPTQ
No
End
Yes
End

7.2.1 Sub-block Format



L - Length. The length of the sub-block in bytes. This includes the block overhead. L may range from 8 to 255.

FLG - Flags. These flags are used by the LIP when disassembling blocks into sub-blocks and reassembling sub-blocks into blocks. The format is:

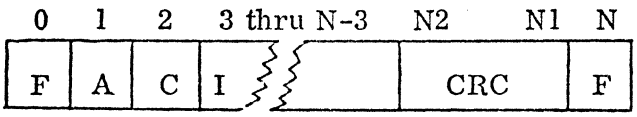


PRI - Set for all packets of a high priority block.

LAST - Set on the last packet of a block.

7.3 Load Dump Trunk Protocol

A degenerate form of the CDCCP protocol is used to bring a NPU to operational state. The following command/response frames will be supported by the bootstrap for loading and dumping:



A = Address = Secondary ID

C = Control

RIM (X'17) Request for Initialization

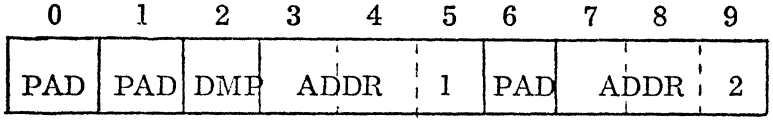
UA (X'73) Non-sequenced Acknowledgement

UI (X'13) Unnumbered Information

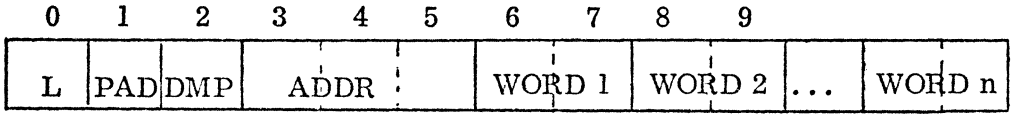
SIM (X'07) Set Initialization Mode

I - Information (allowed only on UI frame)

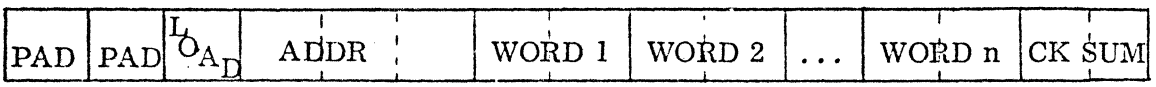
Dump Command



Dump Response



Load Command



7.3 Load Dump Trunk Protocol (Cont'd)

Load Response

An Unnumbered Acknowledgement frame (no data).

Start Command

0	1	2
PAD	PAD	STRT

Start Response

An Unnumbered Acknowledgement frame (no data).

Address - Indicates the starting core address of the dump, load, or executable program.

Data - 1 thru 100 16 bit NPU words of load or dump data and a 16 bit check sum. The data field is not used in the dump command sent from the neighbor NPU to the failed NPU nor in the Start Command.

The bootstrap in the failed NPU sends the RIM command over each of a fixed set of trunks in turn. A neighbor NPU on receipt of the RIM sends an SIM to the failed NPU and also sends a service message (SM) to NS requesting a program load. NS responds by initiating a dump overlay module in the neighbor NPU. The dump process is started when NS sends a service message to the dump module requesting the first block. The dump module sends a UI frame to the failed NPU specifying a block of core to be dumped. The program in the failed NPU responds to each UI dump command with a UI frame containing the appropriate core block. The dump overlay module constructs a dump message of the format described above and forwards this to NS. The process is repeated until the dump is complete. NS will then start the load process.

The load module is loaded into the neighbor NPU overlay area by NS. Load messages are sent over the service channel to the load module. Header information is removed from the message, the UI framing information added to the message, and it is sent to the failed NPU. The program in the failed NPU will load the core block from the message.

7.3 Load Dump Trunk Protocol (cont.)

and perform a checksum calculation. A UA frame is sent to the neighbor NPU. The process is repeated until the failed NPU is completely loaded. The load module then sends a UI frame containing a start execution command to the failed NPU.

Errors of any kind that are not recovered by repeating the affected transmission will cause the failed NPU to restart the process of searching for an available trunk.

7.4 Initialization/Recovery

The LIP is first given control of a fully configured inoperative trunk after initialization and when a disabled trunk is enabled via Service Message from NS. This is called Asynchronous Disconnected Mode (ADM). An attempt is made to make the trunk operational.

The local NPU LIP attempts to change the downline trunk from ADM to Asynchronous Response Mode (ARM) by transmitting an Unnumbered frame (U-frame) containing the SARM command. A UA frame is the expected response. When it is received, the trunk is in ARM and NS is informed via a Trunk Operational SM. All sequence numbers are cleared and I-frames may be transmitted and received as available. The number of operational trunks for the link is incremented to reflect the increased bandwidth. Other possible responses to the SARM command include:

- RIM - This indicates the neighbor NPU is inoperative and is requesting a load/dump over this trunk. The LIP sends SIM as a response and initiates the Load/Dump Process.
- SARM - The other end of the trunk is also in ADM. The LIP sends UA and continues to wait for UA.
- Any other valid frame or a no response timeout will cause the LIP to repeat the SARM command. If repeated SARM commands do not get a response, a Trunk Inoperative SM is sent to NS. The SARM command is repeated indefinitely by the local NPU but the SM is not repeated until some external event occurs such as a request for trunk status. The remote NPU initializes in response to local NPU initialization and does not send trunk status.

Trunk failures occur if a repeated protocol error occurs. All blocks except certain service messages remaining in this Link Queue are discarded. Additional such blocks will not be placed in the Link Queue after the trunk has failed. The associated logical link is failed, and the secondary state is changed, if necessary, to force the other end of the trunk to fail simultaneously. Keep-alive frames (I-frames with no I-field) are used when necessary to ensure timely recognition of a trunk failure. Trunk recovery proceeds the same as for initialization.

8.0 BATCH VIRTUAL TERMINAL (BVT)

The Batch Virtual Terminal provides a standard interface to permit application programs which exchange files with remote batch terminals to be transparent to specific terminal characteristics. Each TIP servicing a batch device, e.g., Mode 4, HASP multileave workstation will transform certain characteristics of the real device to present a common Batch Virtual Terminal Interface to the application in the host.

8.1 Batch Virtual Terminal Characteristics

The BVT is considered to be a multi-device terminal operating remotely and connected to the 255X by a synchronous medium to high speed line. The protocol on the line differs by equipment type, but the BVT is assumed to be a block oriented terminal.

A separate logical connection exists for each device supported. Device types that may exist at the remote site include card readers, printers plotters and card punches. The BVT is defined to allow full use of the features of Mode 4 terminals. Features considered are known characteristics of other target terminals. Features considered are data compression, printer carriage control, code conversion, transparent data mode control and file structure. For downline blocks, the host process is responsible for ensuring that downline network blocks will not exceed the allowable device block size after processing by the TIP and that output print lines do not exceed the device print line width. Similarly the host process is responsible for compressing data. Downline only blank and zero compression is permitted, compression of contiguous identical characters other than blanks or zeros will result in rejection by means of a BRK if sent to a MD 4 Terminal. Upline the degree of compression is determined by the terminal. Full compression should be assumed. At any multi-device terminal the interactive devices conform to IVT and the batch devices to BVT.

8.2 BVT Host Interface

The host interface is described in terms of the block protocol and block content.

8.2.1 Block Protocol Usage

The block protocol is used upline and downline to transfer information and control. Each block protocol element and its usage is described in turn.

BLK

The BLK element is used to transfer non-last blocks of input or output files. The size of the block upline is determined by the terminal. It is a host's responsibility to ensure that the size of the downline block will not exceed the terminal

8.2.1 Block Protocol Usage (cont.)

buffer size after the protocol envelope has been added. The TIP will attempt to deliver all blocks to the terminal. The effect of delivering a too large block differs according to terminal type.

MSG

The MSG element is used to transfer the last or only block of an input or output file. An upline MSG block is generated whenever an end-of-information (EOI) is encountered in the card stream. The EOI will be designated by the <END OF INFORMATION> sequence. A downline MSG block designates the end of a Cyber file.

BACK

The BACK element is used to acknowledge delivery or processing of BLK, MSG or CMD elements for purposes of flow control.

BRK

The BRK element is used to temporarily stop the data flow when an operator action or printer not ready condition is detected. The application is responsible for restarting the flow. The BRK is specifically sent when the TIP receives a block which does not conform to BVT.

STP

The STP element is used to stop the data flow when the end device becomes inoperative or otherwise incapable of accepting more data. The source process is required to protect all data which has not been BACKed and issue no new data.

STRT

The STRT element is used to cancel the effect of the STP element. The source process must respond with a RST and may then resume sending data.

RST

The RST element is used to indicate the point at which a BRK or STRT was actioned. A destination process issuing a BRK or STP discards all unacknowledged and all new BLK, MSG and CMD's until a RST is received. Additional BRK's or STP's may not be issued while the RST is outstanding.

CMD

A command (CMD) is used to cause a change of mode in the other process. A CMD which is to affect data in the opposite direction will not take effect until all data in the same direction ahead of it has been processed. A CMD which is to affect data in the same direction will be actioned at the point in the data stream that it was issued.

8.2.1 Block Protocol Usage (cont.)

Usage is defined in the Terminal Interface Package sections and summarized in Appendix A.

8.2.2 Block Content

Table 8-1 defines the content of message blocks to the level of detail required for BVT processing. Table 8-2 provides a Forms Control definition and Table 8-3 shows the Data Control Byte definitions.

Table 8-1 BVT/Host Interface

<MESSAGE> = [<CONTROL> / <DATA MESSAGE>]

<CONTROL> = [<DOWNLINE CONTROL> / <UPLINE CONTROL>]

<DOWNLINE CONTROL> = <NETWORK HDR> <COMMAND> [<STOP INPUT> / <START INPUT>]

<UPLINE CONTROL> = <NETWORK HDR> <COMMAND> <INPUT STOPPED><REASON CODE>

<NETWORK HDR> = <NETWORK ADDRESS> <PRI> <BSN>

<NETWORK ADDRESS> = <DN> <SN> <CN>

<PRI> = [0 / 1] Priority

<BSN> = [0 / 1 / 2 / . / 7] Block Sequence Number } SEE BLOCK PROTOCOL SECTION

<COMMAND> = 4

<STOP INPUT> =
<REASON CODE> =
<START INPUT> =
<INPUT STOPPED> =
} See Table A-11 in Appendix A for values

<DATA MESSAGE> = [<BLKBLOCK>]$_{0-n}$ <MSGBLOCK>

<BLKBLOCK> = <NETWORK HDR> <BLK><DBC><DATA>

<MSGBLOCK> = <NETWORK HDR> <MSG><DBC><DATA> (<END OF INFORMATION>)

<BLK> = X'01
<MSG> = X'02
} See Block protocol

<DBC> = [<UPLINEDBC> / <DOWNLINEDBC>]

<UPLINEDBC> = 0

<DOWNLINEDBC> = 0

<DATA> = [[(<MODECHANGE>) (<FORMSCONTROL>)
(<COMPRESSEDDATA>) (<ENDOFMEDIA>)]$_{1-n}$
<ENDOFRECORD>]$_{1-n}$

A single <MODECHANGE> is allowed ahead of a physical record. <FORMSCONTROL> is required ahead of each print line. <COMPRESSEDDATA> may be elided, e.g., <FORMSCONTROL> without print. <ENDOFMEDIA> is required at the end of each physical record. <ENDOFRECORD> and <ENDOFINFORMATION> are required to indicate logical record or file boundaries.

<MODECHANGE> = [<DEFAULT> / <29> / <26>] {Affects all data following}

<DEFAULT> = X'FF00 {Default data mode. Always sent following EOI.}

<29> = X'FF02 {Columns 79 and 80 contain '29'}

<26> = X'FF03 {Columns 79 and 80 contain '26'}

<OTHER> = X'FF09 {Columns 79 and 80 contain other than '26', '29'}

<MODECHANGE> reports the presence in an input card stream of card punch indicators. Columns 79, 80 of a Job Card or EOR Card may contain indications of whether the following cards were punched as an 026 or 029 punch. This information is passed to the host to allow additional translation to be performed.

TABLE 8-1 BVT/HOST INTERFACE (Con't)

<FORMSCONTROL> = X'FF [X'E0 | X'E1 | • | • | • | X'FD | X'FE]
{Forms control associated with each print line. See table 8-2 for definition Forms controls which are not supported by a specific device results in a single space. See individual TIP actions for implementation.}

<COMPRESSEDDATA> = [<COMPRESSEDZEROES> | <COMPRESSEDBLANKS> | <REPLICATIONCOUNT> <BYTE> | <STRINGLENGTH> <STRING> | <STRINGINDICATOR> <STRING>] 1-n

<COMPRESSEDZEROES> = X'FF [X'32 | X'33 | • | • | • | X'3E | X'3F]
{Represents 2 thru 15 zeroes}

<COMPRESSEDBLANKS> = X'FF [X'12 | X'13 | • | • | • | X'2E | X'2F]
{Represents 2 thru 31 blanks} Note that trailing blanks may be totally elided.

<BYTE> = (0..255) {Eight bit byte}

<REPLICATIONCOUNT> = X'FF [X'42 | X'43 | • | • | X'8E | X'8F]
{Number of times the byte following the count is to be repeated. Value may range from 2 (X'42) to 79 (X'8F). Upline compression is determined by terminal, full compression capability should be assumed. Not used downline.}

TABLE 8-1 BVT/HOST INTERFACE (Con't)

<STRINGINDICATOR> = X'FF90

{Indication that the following byte string consists of uncompressed data of indeterminate length. The string is terminated by the first non-data X'FF encountered. Used upline only. Any data X'FF patterns must be doubled up by the TIP and the added X'FF must be deleted by the host}

<STRING> = $[BYTE]_{1-n}$

{n is limited by physical record length of terminal device}

<STRINGLENGTH> = X'FF [X'91, X'92, ..., X'CE, X'CF]

{Indication that the following Byte string consists of uncompressed data of length 1 (X'91) thru 63 (X'CF). This method of representing uncompressed data is always used downline but is used upline only when a count is provided by the Terminal, e.g., HASP.

{the following three elements allow file structure to be retained during transfer}

<ENDOFMEDIA> = X'FF0A

{Represents end of physical record e.g., card, print line}

<ENDOFRECORD> = X'FF0B

{Represents end of logical record and may occur at other than block boundary}

<ENDOFINFORMATION> = X'FF0C FF0AFF00

{Occurs only in a MSG block as the last six characters}

TABLE 8-2 FORMS CONTROL DEFINITION

<FORMSCONTROL> (HEX)	Action Before Printing	Action After Printing
E0+	Space 1	No Space
E1+	Space 2	No Space
E2+	Space 3	No Space
E3+	Suppress Space	No Space
E4+	Skip to Channel 1*	No Space
E5	Skip to Channel 12**	No Space
E6	Skip to Channel 6	No Space
E7	Skip to Channel 5	No Space
E8	Skip to Channel 4	No Space
E9	Skip to Channel 3	No Space
EA	Skip to Channel 2	No Space
EB	Skip to Channel 11	No Space
EC	Skip to Channel 7	No Space
ED	Skip to Channel 8	No Space
EE	Skip to Channel 9	No Space
EF	Skip to Channel 10	No Space
F0	No Space	Skip to Channel 1*
F1	No Space	Skip to Channel 12**
F2	No Space	Skip to Channel 6
F3	No Space	Skip to Channel 5
F4	No Space	Skip to Channel 4
F5	No Space	Skip to Channel 3
F6	No Space	Skip to Channel 2
F7	No Space	Skip to Channel 11
F8	No Space	Skip to Channel 7
F9	No Space	Skip to Channel 8
FA	No Space	Skip to Channel 9
FB	No Space	Skip to Channel 10
X'FC-X'FE	RESERVED	

* Page eject
\+ Supported on all devices
** Bottom of page

TABLE 8-3 DATA CONTROL BYTE DEFINITIONS

HEX Value	Use
FF00-FF09	Data Modes
FF0A-FF0F	Information Separators
FF10-FF2F	Compressed Blanks
FF30-FF3F	Compressed Zeros
FF40-FF8F	Compressed Data
FF90	Uncompressed String Terminated by X'FF
FF91-FFCF	Uncompressed String of Length 1 thru 63
FFD0-FFDF	Not Used
FFE0-FFFE	Forms Control
FFFF	Data Character FF

TABLE 8-4 CYBER JOB STREAM CARD INPUT EXAMPLES

Terminal Input	BVT
Jobcard:	
JOBNAME.... 29 26 xx (2)	(1) FF90 JOBNAME.... 29 26 xx (3) FF0A FF 02 03 09 (4)
End of Record Card:	
7 8 9 nn (6) /* EOR nn.... 29 26 xx	(5) FF0B nn FF0A FF 02 03 09
End of Information card:	
6 7 8 9 /* EOI	(7) FF0C (8) FF0A FF00

1. Uncompressed stream terminated by FF flag.
2. Columns 79/80 of JOB card may contain 26/29 code sequence.
3. End of physical record sequence.
4. Data mode (mode change) caused by contents of column 79/80 of JOB or EOR card.
5. EOR sequence.
6. EOR card may contain octal logical level number following EOR designator.
7. EOI sequence.
8. Data mode is always reset to FF00 on EOI.

9.0 Interactive Virtual Terminal (IVT)

This section describes the interface between application programs in a CYBER host and Terminal Interface Programs (TIP's) in a 255X supporting interactive terminals. Details of the user interface and virtual-to-real transforms are described in the appropriate TIP section.

9.1 Overview

The variety of real terminals which may be used to access interactive processes causes an obvious problem of incompatibility. This problem is of greater concern on the output side where the use of format effectors produces undesirable and unintended effects. The concept of an Interactive Virtual Terminal (IVT) is introduced to solve this problem.

An IVT may be defined such that an application may expect compatible input from a terminal and issue output to a terminal with confidence that the intended results will occur. The function of the IVT is to provide the necessary transforms between selected types of real terminals and the virtual terminal; and further to provide a measure of variation of these transforms to widen the variety of real terminals which may be accomodated.

The choice of functions to be provided by the IVT is deliberately restricted to ensure that even when transforming to the real terminal with the lowest capability significant intelligence will not be lost. Similarly, consideration must be given to compatibility between existing software, which drives real terminals and the IVT. Here the application is required to provide additional transforms. Where the application requires to use features not provided by IVT, but known to exist on the connected real terminal, this may be achieved in one of two ways:

- The application may embed appropriate control characters in the output text or, conversely, scan for significant control characters in the input text. Due regard must be made for the control characters which are significant (and, therefore, possibly transformed by) the IVT.
- By transferring data in transparent mode, the transforms are inhibited and the application then has direct access and responsibility for all real terminal features. The transparent mode is separately selectable in each direction.

9.2 Virtual Terminal Characteristics

An IVT will always have an input device and an output device. The input device is typically a keyboard, but may well be a paper tape reader or cassette reader. When not a keyboard, IVT will normalize reader input to appear identical to keyboard input (exception - transparent mode). The output device is typically a printer or display, but may well be a paper tape punch or cassette recorder. The application will be unaware of the output media in general. Optional additional equipment supported is a paper tape reader/punch, where use of <XON><XOFF> is required to control reading of input. In this case, an explicit switch to paper tape mode is necessary before operation of paper tape equipment.

The IVT does not provide a method of switching between display and printer, but assumes local hard copy facilities may exist. IVT device parameters as seen by a host application are:

Line Width	-	Infinite (subject to block limit)
Page Size	-	Infinite
Parity	-	None (set to zero)
Code Set	-	ASCII - 128 characters available
Format Effector Delays	-	None

IVT Format Effectors (FE) are an optional feature of downline data blocks. A flag in the Data Block Clarifier determines whether Format Effectors are present or not. If the flag is set, FE's are not present and each logical line will be pre-print single spaced on output and the first character will be printed. If the flag is zero, FE's are assumed as the first byte of each logical line of text. Undefined FE's will result in pre-print single space. The interpretation of FE's is as follows:

(a)

Action takes place pre-print	F.E in ASCII
Single Space	<SP>
Double Space	0
Triple Space	-
Position to start of current line	+
Position to top of form or home cursor	*
Home cursor and clear screen	1
No action	,

9.2 Virtual Terminal Characteristics (cont.)

(b) Action takes place post-print	F.E. in ASCII
Single Space	.
Position to start of current line	/

IVT additional controls which may be passed downline in text are:

Carriage Return	- <CR>
Line Feed	- <LF>

Other potential control characters or control character sequences are translated one for one. Thus the application can detect special input control sequences and transmit special output control sequences by taking note of the translation performed for a specific terminal. Idle fill will only be inserted for <CR> and <LF> however. IVT operational controls which may be passed via flags in the DBC of downline blocks are:

Auto-input	- Return this output with next input (only actioned on a MSG type block)
Transparent	- Inhibit IVT transform for this output
FE	- FE's present or absent

IVT operational controls which may be passed via flags in the DBC of upline blocks are:

Transparent	- This block was not transformed by IVTIP
Parity error	- This block had one or more parity errors
Cancel	- Cancel the message of which this MSG block is a part.

IVT mode control which may be effected by downline synchronous commands are shown in Table 9-1 under the description of a <DOWN-LINE CONTROL>. IVT will use upline synchronous commands to communicate 'input stopped'.

9.3 Host Interface

The TIP resident in a 255X communicates with the application in the CYBER host. All communications between the two is subject to processing by an intermediate process in the host. The Network Access Method (NAM) is the intermediate process and exists to provide a common logical interface to the communications network.

The IVT interface to the host is necessarily described at two levels -- the interface to NAM and the overlying interface to the interactive application. The interface to NAM is the Block Protocol and an understanding of this is a pre-requisite for the reader of this section.

9.3.1 Block Protocol Usage

The IVT uses the Block Protocol, both up-line and down-line, for the transfer of information and control. Each element of the Block Protocol used, and its method of use, is described in turn.

BLK Element

The BLK, or non-last segment of a message, is used for transferring data both up-line and down-line. When a message is greater than "n", then the message is divided into blocks of size "n" bytes. All non-last segments are sent as BLKS. "n" has a maximum value of 2043, but would normally be a lower value to conserve 255X resources.

Upline, a character mode block is a partial logical line, (typically a physical line), sent at the convenience of the TIP. See the TIP section for a discussion of upline blocking. A transparent mode block consists of a system-defined number of bytes.

The optimum block size for the IVT is a small number of physical lines for the specific terminal. For special applications such as graphics or paging the optimum block size is a single display.

MSG Element

The last or only segment of a message is sent as a MSG block. For transparent down-line data, if page wait is selected, the MSG block indicates the end of the page.

9.3.1 Block Protocol Usage (cont.)

BACK Element

The BACK element is used for flow control. A BACK is sent by the receiving process (NAM/TIP) when it has delivered, or otherwise disposed of a BLK, MSG or CMD.

CMD Element

The CMD provides a means of passing control information synchronously with the data stream, but apart from the BLK's and MSG's, which constitute the data stream. The CMD is used to perform the functions specified in Table 9-1.

BRK Element

The TIP will send the BRK for the following reasons:

(a) User Break 1 received from terminal (typically means Abort Queue)

(b) User Break 2 received from terminal (typically means Abort Job)

(c) Downline block does not conform to IVT

In all cases, the TIP will discard all locally queued output data and all newly arriving data until a RST is detected. The use of BRK downline is not anticipated.

Note: Data discarded includes synchronous CMDs.

STP Element

The TIP may send a STP to the application requesting suspension of output.

STRT Element

The STRT element cancels the effect of the STP element.

RST Element

The RST is sent by a process when it has actioned a BRK or STRT. It indicates the point in the data stream that the action occurred. A further STP or BRK must not be issued when a RST is outstanding.

9.3.2 Host Interface Definition

Table 9 -1 defines the interface between the TIP and the host. The following points are emphasized:

(a) All up-line Character mode messages consist of zero or more BLK blocks and a single MSG type block, each block containing typically a single physical line and the whole comprising a single logical line.

(b) Down-line Character mode messages may be multiblock and each block may contain multiple logical lines. Logical lines may not cross block boundaries.

(c) For down-line Character mode messages, a flag in the DBC indicates whether Format Effectors are present. If so, all logical lines are preceded by a Format Effector byte. Multiple logical lines in a MSG block are separated by a <US> (X'1F).

(d) A logical line may contain any of the 128 ASCII character set, except for <US>.

(e) In Character mode, all ASCII characters consist of seven bits, right-justified, in an eight-bit byte with the parity set to zero.

(f) All bytes of a Transparent mode block may contain any of the 256 possible bit combinations, except that, for an up-line block if a terminator character is defined, this will not appear. Note that terminal or CLA configuration may restrict the significant number of bits in the byte to less than eight.

(g) All blocks contain a Data Block Clarifier (DBC) byte immediately preceding the data. This byte contains flags, which provide additional information concerning the data in the block. Certain flags are used only in up-line or down-line blocks.

(h) BLKS, MSGs and CMDs sent to a TIP which are incorrectly formatted for IVT will result in an upline BRK.

Table 9-1

Virtual Terminal/Host Interface

<MESSAGE> = [<CONTROL> <DATA MESSAGE>]

<CONTROL> = <NETWORK HDR> [<COMMAND> <BREAK>] [<DOWNLINE CONTROL> <UPLINE CONTROL> <REASON CODE>]

<NETWORK HDR> = <DN> <SN> <CN> <PRI> <BSN> — See Block Protocol Section

<COMMAND> = 4

<BREAK> = 5

<REASON CODE> — See Block Protocol Section

<DOWNLINE CONTROL> = [<TERMINAL CONTROL> <TERMINAL PARAMETERS>]

<TERMINAL CONTROL> = [<STOP INPUT> <START INPUT>]

<STOP INPUT> = <PFC> <SFC>

<START INPUT> = <PFC> <SFC>

See Table A-11 in Appendix A for values of PFC, SFC.

<UPLINE CONTROL> = <INPUT STOPPED>

<INPUT STOPPED> =

See Table A-11 in Appendix A for values of PFC, SFC.

<TERMINALPARAMETERS> = <PFC><SFC>

(See Table A-11 for values)

TC = [1 2 . . . 14 15]

PW = <NNN>

PL = <NNN>

PA = [Z O E N]

CN = <SELECTED CHAR>

BS = <SELECTED CHAR>

CT = <SELECTED CHAR>

CI = [CA <NN>]

LI = [CA <NN>]

AP = [N Y S]

DL = (X<HH>)(,C<NNNN>)(,TO)

IN = [KB XK PT XP]

OP = [PR DI PT]

EP = [Y N]

PG = [Y N]

AL = <SELECTED CHAR>

B1 = <SELECTED CHAR>

B2 = <SELECTED CHAR>

MS = <TEXT>

All commands shown are text strings. See Section 9.4 for details.

Table 9 - 1 Virtual Terminal/Host Interface (Con't)

<NNNN>	=	(0..4095) {One to Four Decimal Digits}
<NNN>	=	(0..255) {One to Three Decimal Digits}
<NN>	=	(0..99) {One to Two Decimal Digits}
<SELECTEDCHAR>	=	{ ASCII Representation of Selected Character}
<HH>	=	(x'00..x'FF) {Selected Bit Pattern as Sent by Terminal}
<TEXT>	=	{One through fifty ASCII characters message }

<DATAMESSAGE>	=	[<TRANSMODEMSG> <CHARMODEMESSAGE>]
<TRANSMODEMSG>	=	[<BLKBLOCK>] 0-n* <MSGBLOCK>
<BLKBLOCK>	=	<NETWORKHDR> <BLK> <TRANSBLKCONTENT>
<MSGBLOCK>	=	<NETWORKHDR> <MSG> <TRANSBLKCONTENT>
<BLK>	=	x'01
<MSG>	=	x'02
<TRANSBLKCONTENT>	=	[<DBCUPLINE> <DBCDOWNLINE>] [BYTE] 0-n *
<BYTE>	=	(0..255) {CLA mode or terminal may not support full range. If terminator is specified, then that value will not appear upline}
<DBC DOWNLINE>	=	<SPARE> <SPARE> <SPARE> <SPARE> <FEUSAGE> <TRANSPARENT> <NOTUSED> <AUTOINPUT> {Binary flags}

* the value of n is defined later

Table 9-1 Virtual Terminal/Host Interface (Con't)

<AUTOINPUT> = [0 This output is not autoinput / 1 This output to be returned ahead of next input]

<FEUSAGE> = [0 Format effectors used / 1 Format effectors not used]

<DBCUPLINE> = <SPARE> <SPARE> <SPARE> <NOTUSED> <NOTUSED> <TRANSPARENT> <CANCELLED> <PARITYERROR> {Binary Flags}

<CANCELLED> = [0 No Action Required / 1 Cancel any incomplete upline message]

<TRANSPARENT> = [0 Block is in char mode / 1 Block is in transparent mode]

<PARITYERROR> = [0 Parity errors not detected / 1 Parity errors detected]

<CHARMODEMESSAGE> = [<CHARMODEBLK>]$_{0-n*}$ <CHARMODEMSG>

<CHARMODEBLK> = <BLKADDRESS> <BLK> [<UPLINE CONTENT> / <DOWNLINE CONTENT>]

<CHARMODEMSG> = <BLKADDRESS> <MSG> [<UPLINECONTENT> / <DOWNLINECONTENT>]

<UPLINECONTENT> = <DBCUPLINE> [<PHYSICALLINE> / <LOGICALLINE>]

<PHYSICALLINE> = [<128ASCIICHARSETWITHPARITYSETTOZERO>]$_{0-n*}$ {Blocking may occur at physical line boundary or logical line boundary. See individual TIP for a discussion of upline blocking.}

<LOGICALLINE> = [<128ASCIICHARSETWITHPARITYSETTOZERO>]$_{0-n*}$ {Except total of all bytes in a Block must not exceed n. n is a system parameter separately declarable upline and downline with a maximum of 2043}

<DOWNLINECONTENT> =

<DBCDOWNLINE> [[<FE><LOGICALLINE><US>]$_{0-n*}$ [<FE><LOGICALLINE>] / [<LOGICALLINE><US>]$_{0-n*}$ [<LOGICALLINE>]]

<US> = x'1F {<US> must not appear in a downline logical line}

* the value of n is defined later

Table 9 -1 Virtual Terminal/Host Interface (Con't)

```
<FE>  =  [ <SINGLESPACEPRE>
           <DOUBLESPACEPRE>
           <TRIPLESPACEPRE>
           <STARTOFCURRENTLINEPRE>
           <FORMFEEDPRE>
           <HOMEANDCLEARPRE>
           <NULL>
           <SINGLESPACEPOST>
           <STARTOFCURRENTLINEPOST> ]
```

```
<SINGLESPACEPRE>          = <SP>   }
<DOUBLESPACEPRE>          = '0'    }
<TRIPLESPACEPRE>          = '-'    }  {Preprint Format Effectors}
<STARTOFCURRENTLINEPRE>   = '+'    }
<FORMFEEDPRE>             = '*'    }
<HOMEANDCLEARPRE>         = '1'    }
<NULL>                    = ','    {No Action}
<SINGLESPACEPOST>         = '.'    }
<STARTOFCURRENTLINEPOST>  = '/'    }  {Postprint Format Effectors}
```

9.3.3 Communication Supervisor (CS) Interface

The following Service Messages are used by IVT.

9.3.3.1 Terminal Class, Page Width, Page Length

Parameters

Port

Subport

Cluster Address

Terminal Address

Device Type

Originator (user or application)

New Terminal Class (1 thru 15)

New Page Width (0-255)

New Page Length (0-255)

Purpose

To make available to host applications the current Terminal Class, Page Width or Page Length. Sent by the NPU to CS.

Stimulus

One of the specified parameters is changed by user or applications.

9.3.3.2 Operator Message

Parameters

Port

Subport

Cluster Address

Terminal Address

Device Type

Text (50 characters or less)

9.3.3.2 Operator Message (cont.)

Purpose

To allow a user at a terminal to communicate with the local operator. Sent by the NPU to CS.

Stimulus

Receipt of the command <CTL> MS = text at an interactive terminal.

9.3.3.3 Host Broadcast

Parameters

Logical Link ID or Terminal ID
DBC
Text (50 characters or less)

Purpose

To deliver informative messages to interactive user(s). Sent by host to NPU.

Stimulus

Local operator wishes to broadcast a message or send a message to a specific user.

9.4 Commands for Terminal Parameterisation

Host software may send downline control messages as defined in Table 9-1 to change the mode of operations and parameterisation of a terminal. Each control message consists of a synchronous command (see Block Protocol) with a single command embedded as an ASCII text string. All control messages from host software to the TIP may also be entered by a terminal user. Three of the commands, namely Terminal Type, Page Width and Page Length when entered by the terminal user or issued by host software are reported to the Communication Supervisor. All terminal user entered commands result in an acceptable/unacceptable response to the user. Host commands which are invalid or illegal will not be actioned, but will be rejected with a BRK, and will be printed on the NPU console (see Table 9-2 for commands acceptable by each TIP).

9.4 Commands for Terminal Parameterisation (Cont'd)

Terminal Class (TC)

Establishes a class for the terminal with default values for all parameters as defined in Table A-10. A TIP will not obey the command if the class is not supported.

Page Width (PW)

To establish the physical line width in characters for output. For non-transparent blocks, the TIP inserts the character sequence defined for the terminal class to move the carriage or cursor to the next line at the point where the number of characters to be transmitted equals the page width. The parameter NNN varies between 0 and 255, where 0 means "new line" is never inserted.

Page Length (PL)

To establish the number of physical lines in a page for output. The TIP inserts the character sequence defined for the terminal class to advance the carriage or cursor to the next page when the number of physical lines transmitted equals the page length. Also, if the Page Wait feature is selected, the TIP will wait for an operator input before continuing. The parameter NN varies between 0 and 255, where 0 means no paging.

Parity Selection (PA)

To inform the IVTIP which type of parity to expect on input and generate on output. See description of Parity in the Asynchronous TIP section.

Cancel Character (CN)

To establish the character to be used to cause the logical input line in process to be deleted.

Backspace Character (BS)

To establish the character to be used to cause the previous input character to be deleted from the input buffer in process.

Control Character (CT)

To establish the character to be used to enter operational control messages.

Carriage Return Idle Count (CI)

To establish the number of idle characters to be inserted in the output stream following a CR. Theuse of CI-nn overrules the default known to the TIP and CI-CA restores the default.

9.4 Command for Terminal Parameterisation (cont.)

Line Feed Idle Count (LI)

To establish the number of idle characters to be inserted in the output stream following a LF. The use of LI=NN overrules a default and LI=CA restores the default.

APL Code Set (AP)

To switch to/from the APL code set from/to the user's normal code set. (Valid only for certain terminal types.) In APL normal mode, input processing is changed to allow normal action on special characters; e.g., backspace, cancel input line, etc. In APL special mode BS and cancel characters are sent upline as data.

Transparent Text Delimiter (DL)

To establish the transparent text delimiter for input. The delimiter may be a character, a character count or a timeout of 300 ± 100ms. One or more of the delimiters may be active simultaneously. Default values are shown in Table A-10.

Input Device (IN)

To specify the input device as a keyboard or paper tape reader in character or transparent mode. Note that paper tape input is allowed in keyboard mode, but that the TIP does not send the <X-ON> characters to start the paper tape reader.

Output Device (OP)

To specify the output device as printer, CRT display, or paper tape punch. Printer and CRT display are functionally equivalent. The user may punch a paper tape in any mode, but the TIP will only provide the <X-OFF> character if OP=PT and if data is not transparent.

9.4 Command for Terminal Parameterisation (cont.)

Echoplex mode (EP)

To specify where input character echoing will take place. EP=N implies the terminal is doing its own input echoing and EP=Y causes the TIP to set the CLA to provide character echoing.

Page Wait (PG)

Selects Page Wait on or off. Allows the user to control output by demanding each page explicitly when the previous page has been assimilated.

Abort Output Line Character (AL)

Selects the character which, when input followed by a CR will result in the current output line being discarded.

User Break 1 (B1)

Selects the character which, when input followed by a CR will result in an upline BRK with reason code 'User Break 1". Conventionally thisis used to Abort Queue.

User Break 2 (B2)

Selects the character which, when input followed by a CR will result in an upline BRK with reason code "User Break 2". Conventionally this is used to Abort Job.

Command	MD4	HASP	ASYNC
TC	AR**	B	AR
PW	AR	AR	AR
PL	AR	B	AR
PA	B	B	A
CN	A	A	A
BS	B	B	A
CT	A	A	A
CI	B	B	A
LI	B	B	A
AP	B	B	A/B*
TM	B/A***	B	A/B*
DL	B	B	A/B*
IN	B	B	A
OP	B	B	A
EP	B	B	A
PG	A	B	A
AL	B	B	A
B1	A	A	A
B2	A	A	A
MS	C	C	C
Other or invalid parameters	B	B	B

A = Action
AR= Action and Report to CS
B = No Action and Send BRK or ERR..
C = Valid only from User

* These commands are only valid for certain terminal classes. TM and DL are not valid commands for terminal class 4 (IBM 2741). The AP command is only valid for terminal classes 3 (Memorex 1240) and 4. A BREAK will be sent to the application if any of these commands are received for a terminal in a class which does not support the command.

** An error will occur for any attempt to change the mode from 4A to 4C or vice versa.

*** Transparent mode is legal for mode 4C devices only.

Table 9-2 Action on IVT Commands

10.0 HASP MULTI-LEAVING TIP

10.1 Multi-Leaving Protocol Definitions

The Multi-Leaving Protocol consists of the bidirectional transmission of information blocks between an NPU and a HASP Multi-Leaving terminal in a transparent or non-transparent mode. The informational blocks are defined to be the following types of blocks:

1. Control Blocks
2. Data Blocks

Control blocks contain BSC control characters.

Data blocks contain data records which are character strings and their associated character string control bytes. Each data record in the data block is associated with a specific peripheral device. In order to facilitate identification, a stream number and a device type are assigned to the data record via a record control byte (RCB). Each record control byte has a sub-record control byte (SRCB) associated with it to provide additional information about the data record.

A data block may consist of several data records, all of which may or may not be from the same device. In order to control the flow of data from or to any particular device, a function control sequence (FCS) is added to each data block.

To facilitate error detection, a block control byte (BCB) is added to each data block.

10.2 Multi-Leaving Protocol Operations Description

The following narrative is a general description of how the Multi-Leaving protocol operates:

> The terminal software is loaded and the communications line is initialized. After the SIGN-ON command is transmitted, the NPU and the terminal transmit idle blocks until a function is desired.
>
> When a function other than a console message or console command is desired, the process desiring to initiate the function transmits a request to initiate function transmission RCB. The receiving process then transmits a permission to initiate function transmission RCB if the data from the requesting process can be processed. If the data cannot

10.2 (Con't) be processed, or the function is now in process, the request to initiate a function transmission RCB is ignored.

When a permission to initiate a function transmission RCB is received, the requesting process begins transmitting data blocks to the other process. Data blocks can be transmitted until an EOF is encountered. In order to transmit more data blocks on the same device stream, the request to initiate a function transmission RCB sequence must be initiated again. If a request to initiate a function transmission is not received before data blocks are received, the data blocks are ignored.

Data blocks are transmitted one block at a time. Before another block can be transmitted, the receiving process must transmit a positive response. A positive response is an acknowledge control block or a data block.

Console functions (operator messages/commands) do not have to follow the request to initiate - permission to initiate sequence A console function may be initialized any time the wait-a-bit in the FCS is not set, and remote console bit is set.

10.3 Multi-Leaving Block Descriptions

10.4 Control Blocks

Four types of control blocks are used in the Multi-Leaving protocol. These control blocks are:

1. Acknowledge block
2. Negative acknowledge block
3. Enquiry block
4. Idle block

A description of the blocks and the block usage are contained in the following subsections. See table 10.4 - 1 for a description of the significant EBCDIC Characters.

10.4.1 Acknowledge Block (ACKO)

The acknowledge block (ACKO) consists of the following control characters: SYN, SYN, SYN, DLE, ACKO, PAD

Where: SYN = Synchronization control character
DLE = Data link escape control character
ACKO = Affirmative acknowledgement control character
PAD = Pad control character (all 1 bits)

The ACKO block is transmitted to indicate that the previous block was received without error and no data is available for transmission.

TABLE 10.4.1 HASP Significant EDCDIC Characters

Char	Hex	Octal	Decimal	Meaning
SOH	01	001	1	Start of Header
STX	02	002	2	Start of Text
DLE	10	020	16	Data Link Escape
ETB	26	046	38	End of Block
ENQ	2D	055	45	Enquiry
SYN	32	062	50	Synchronize
NAK	3D	075	61	Not Acknowledge
ACKO	7D	160	112	Acknowledge
PAD	FF	177	255	Pad

NOTE: ACKO only has significance in the sequence DLE ACKO (as the entire message) since ACKO is not a protocol character.

10.4.2 Negative Acknowledge Block (NAK)

The negative acknowledge block (NAK) consists of the following control characters: SYN, SYN, SYN, NAK, PAD

Where: SYN = Synchronization control character
NAK = Negative acknowledgement control character
PAD = Pad control character (all 1 bits)

The NAK block is transmitted to indicate that the previous block was received in error and a retransmission is necessary. A NAK block is never transmitted as a response to a NAK block.

10.4.3 Enquiry Block (ENQ)

The enquiry block consists of the following control characters: SYN, SYN, SYN, SOH, ENQ, PAD

Where: SYN = Synchronization control character
SOH = Start of leader control character
ENQ = Enquiry control character
PAD = Pad control character (all 1 bits)

The enquiry block is transmitted to establish communications with HASP and the NPU. The enquiry block is only used at system loading time.

10.4.4 Idle Block (ACKO)

The idle block is an ACKO block which is used to maintain communications and avoid unprogrammed time-out when neither process has any data to transmit. The idle block is transmitted at least every two seconds.

10.5 Data Block Control Bytes

The control bytes, that are part of each data block, are described in the following subsections.

10.5.1 Block Control Byte (BCB)

The block control byte bit representation is as follows:

```
BIT NO.   0       7
          ØXXXCCCC
```

Where: Ø = 1 (must always be on)
XXX = 000 = Normal block
= 001 = Ignore sequence count
= 010 = Reset expected block sequence count to CCCC
= 011 -111 = Not used in this implementation
CCCC = Module 16 block sequence count

10.5.2 Function Control Sequence (FCS)

The function control sequence bit representation is as follows:

```
BIT NO. 0       78        F
        ØSRRABCDØTRRWXYZ
```

Where: Ø = 1 (must always be on)
S = 1 = Suspend all stream transmission (wait-a-bit)
= 0 = Normal state

Note - for the following bits
-a bit = 1 = Continue function transmission
-a bit = 0 = Suspend function transmission

T = Remote console stream identifier
R = Not used in this implementation
ABCDWXYZ = Various function stream identifiers

10.5.2 (Con't) These stream identifiers are oriented to the recipient. An FCS from the NPU to the terminal represents card reader function stream identifiers according to the following:

Card Reader No.	1 = A
Card Reader No.	2 = B
Card Reader No.	3 = C
Card Reader No.	4 = D
Card Reader No.	5 = W
Card Reader No.	6 = X
Card Reader No.	7 = Y
Card Reader No.	8 = Z

An FCS from the terminal to the NPU represents punch and printer function stream identifiers according to the following:

Printer No. 1 = A = Punch No. 8
Printer No. 2 = B = Punch No. 7
Printer No. 3 = C = Punch No. 6
Printer No. 4 = D = Punch No. 5
Printer No. 5 = W = Punch No. 4
Printer No. 6 = X = Punch No. 3
Printer No. 7 = Y = Punch No. 2
Printer No. 8 = Z = Punch No. 1

10.5.3 Record Control Byte (RCB)

The record control byte bit representation is as follows:

BIT No. 0 7
ØIIITTTT

Where: Ø = 0 = End of transmission block (IIITTTT = 0)
1 = All other RCB's
III = Stream identifier if TTTT ≠ 0
= Control information if TTTT = 0 (control record)
= 000 = Not used in this implementation
= 001 = Request to initiate a function transmission*
= 010 = Permission to initiate a function transmission*
= 011 - 101 = Not used in this implementation
= 110 = Bad BCB on last block received
= 111 = General control record*
TTTT = Record type identifier
= 0000 = Control record
= 0001 = Operator message display request (downline)
= 0010 = Operator command (upline)
= 0011 = Card Input record
= 0100 = Print record
= 0101 = Punch record
= 0110 - 111 = Not used in the implementation

*The RCB for these functions is contained in the SRCB.

10.5.4 Sub Record Control Byte (SRCB)

The sub record control byte bit representation is as follows:

BIT No. 0 7
ØSSSSSSS

Where: Ø = 1 (must always be on)
SSSSSSS = Additional record information dependent upon record type (RCB)

.RCB = General control record
SSSSSSS = 10000001 = Initial terminal sign-on
.RCB = Request or permission to initiate a function transmission
SSSSSSS = Stream identifier and record type identifier as described in RCB
.RCB = Bad BCB on last block received
SSSSSSS = Expected block sequence count
.RCB = Print record
SSSSSSS = MCCCCCC

Where: M = 0 = Normal carriage control
1 = Not used in this implementation
CCCCCC = Carriage control information
= 1000NN = Space immediately NN spaces
= 11NNNN = Skip immediately to channel NNNN
= 0000NN = Space NN spaces after print
= 01NNNN = Skip to channel NNNN after print
= 000000 = Suppress space

.RCB = Punch record
SSSSSSS = MMBRRSS

Where: SS = Punch stacker select information
B = 0 = Normal EBCDIC card image
= 1 = Not used in this implementation
MM = 00 = SCB count units = 1
= 01 - 11 = Not used in this implementation
RR = Not used in this implementation

.RCB = Input record
SSSSSSS = MMBRRRR

Where: MM = 00 = SCB count units = 1
01 - 11 = Not used in this implementation
B = 0 = Normal EBCDIC card image
= 1 = Not used in this implementation
RRRR = Not used in this implementation

10.5.5 String Control Byte (SCB)

The string control byte bit representation is as follows:

BIT No. 0 7
OKTCCCCC

Where: O = 0 = End of record (KTCCCCC = 0)
= 1 = All other SCB's
K = 0 = Duplicate character string
T = 0 = Duplicate character is a blank
= 1 = Duplicate character is non blank (character follows SCB)
CCCCC = Duplication count
K = 1 = Non-duplicate character string
TCCCCC = Character string length

If KTCCCCC = 0 and O = 1, SCB indicates record is continued in the next transmission block. This feature is not supported by HASP and is shown for completeness only.

10.6 Data Block Description

Data blocks consist of data records, the control byte described in the previous sub-sections and the following text control characters:

SYN = synchronization control character
DLE = data link escape control character
SOH = start of header control character - used only if non-transparent mode
STX = start of text control character
ETB = end of transmission block control character
CRC-16 = cyclic redundancy checking control characters (2 bytes)
PAD = pad control character (all 1 bits)

A typical data transmission block is shown in figure 10.1.

10.7 Short Block Descriptions

There are several blocks that appear to be data blocks but are really special case data blocks. These short blocks are:

- Operator console blocks
- End of file blocks
- FCS change blocks
- Sign-on blocks
- BCB error blocks

TABLE 10.1

Typical Multi-Leaving Data Transmission Block

SYN	
SYN	- Synchronization Characters
SYN	
DLE	- BSC Leader (SOH if no transparency feature)
STX	- BSC START-OF-TEXT
BCB	- Block Control Byte
FCS	- Function Control Sequence (2 bytes)
RCB	- Record Control Byte for record 1
SRCB	- Sub-Record Control Byte for record 1
SCB	- String Control Byte for record 1
D A T A	- Character String
SCB	- String Control Byte for record 1
D A T A	- Character String
SCB=0	- Terminating String Control Byte for record 1
RCB	- Record Control Byte for record 2
SRCB	- Sub-Record Control Byte for record 2
SCB	- String Control Byte for record 2
D A T A	- Character String
SCB=0	- Terminating String Control Byte for record 2
RCB=0	- Transmission Block Terminator Record Control Byte
DLE	- BSC Trailer (SYN if no transparency feature)
ETB	- BSC Ending Sequence
CRC-16	- Cyclic Redundancy Checksum (2 bytes)
PAD	- All 1 Bits

The sign-on blocks are described in the terminal start-up section 10.9 and the BCB error blocks are described in the Error Conditions Section 10.8 .

10.7.1 Operator Console Blocks

Blocks which contain operator console messages or commands are special in that no additional records are packed into the data block following a console record.

A request to initiate a transmission function is not required to transmit console records. The only restriction is that the WAIT-A-BIT is not set in the FCS, and the remote console bit is set.

10.7.2 End of File Blocks (EOF)

Blocks which contain end of file are special in that no additional records from the same device stream are packed into the data block following an EOF. Data blocks which are terminated by an EOF contain a final record which is as follows (for reader no. 1):

RCB = 10010011 - Card reader stream no. 1
SRCB = 10000000 - SCB count units = 1, EBCDIC card images
SCB = 00000000 - EOF
RCB = 00000000 - Transmission block terminator (BSC trailer)

In order to transmit more records for a device stream that contains an EOF, the request to initiate a function transmission must be transmitted again. If another device stream contains data for transmission and has permission available to transmit, the last RCB in the above example would be a device stream RCB followed by data instead of a transmission block terminator.

10.7.3 FCS Change Blocks

The FCS change block is transmitted when the status of one or more of the streams has changed, and there is no data to transmit. The FCS change block is as follows:

(BSC Header)

BCB
FCS - Changed FCS
RCB = 00000000 - Transmission block terminator

(BSC Trailer)

10.8 Error Conditions

The error conditions that are seen by the HASP TIP are:

- CRC-16 error
- Illegal block make-up
- Unknown response
- Time out
- BCB error

10.8.1 CRC-16 Error (Cyclic Redundancy Checking)

Cyclic redundancy checking only occurs on data blocks.

If a CRC-16 error occurs, the receiving process transmits a NAK block to the transmitting process which informs the transmitting process that a retransmission of the last block is required. If the retransmitted block is correct, the processing continues.

10.8.2 Illegal Block Make-Up Error

A data block must end with an ETB control character. If the data block does not, then an illegal block make-up error occurs. The receiving process transmits a NAK block to the transmitting process which informs the transmitting process that a retransmission of the last block is required. If the transmitted block is correct, the processing continues.

10.8.3 Unknown Response Error

An unknown response error occurs when the response received from the transmitting process is not one of the following:

- A data block beginning with DLE, STX control characters in transparent mode
- A data block beginning with SOH, STX control characters in non-transparent mode
- An ACK block
- A NAK block

If an unknown response error occurs, the receiving process transmits a NAK block to the transmitting process which informs that transmitting process that a retransmission of the last block is required. If the retransmitted block is correct, processing continues.

10.8.4 BCB Error

Every data block contains a BCB byte and in each BCB byte is a block sequence count. The data blocks are transmitted in sequentially ascending order unless an ignore or reset BCB byte is transmitted. If the block sequence count in the data block is not equal to the block sequence count expected by the receiving process a BCB error occurs.

If a BCB error occurs and the block sequence count is a duplicate of a block sequence count previously received (expected block sequence count minus received block sequence count ≤ 2), the data block is ignored and processing continues as if an FCS change block or ACK block was received.

If a BCB error occurs and the block sequence count is not a duplicate block count as described in the previous paragraph, a BCB error block is transmitted from the receiving process to the transmitting process. The BCB error block informs the other process that a block sequence count error has occurred, and that the transmitting process is to back up the file to the missing block or is to transmit a reset BCB byte. The

10.8.4 (cont.)

format of the BCB error block is:

(BSC Header)

BCB = 1001XXXX = Ignore sequence checking, XXXX = received block sequence count

FCS
RCB = 11100000 - Bad BCB on last block
SRCB = 1000YYYY - expected block sequence count
SCB = 00000000 – end of record
RCB = 00000000 - Transmission block terminator

(BSC Trailer)

For BCB errors detected by the HASP TIP for any input or output device stream, no recovery by the HASP TIP will be attempted. The HASP TIP will send a Line Status Service Message with Line Inoperative.

10.9 Terminal Start-Up and Termination

Terminal start-up is accomplished via a three-step process:

- Terminal initialization
- Communication line initialization
- Sign-On

10.9.1 Terminal Initialization

The terminal software is loaded and put into execution. The loading can be by paper tape, cards, magnetic tape or mass storage depending upon the terminal hardware. The initialization processor establishes I/O buffers and other necessary parameters. After all the buffers are set, a card is read from the card reader. If the card is blank, the default sign-on parameters are used (default sign-on parameters are assembled into the terminal software). If the card is a /*SIGNON card, the parameters on the /*SIGNON card are used instead of the default.

10.9.2 Communication Line Initialization

After the terminal is initialized, the communication line is initialized. The line is initialized by the HASP TIP upon receipt of a Configured Line service message from the host. When communications are established with the line, communications between the HASP TIP in the NPU and the terminal are established via the following procedure:

- An ENQ block is sent from the terminal process to the HASP TIP.
- The ENQ is ignored by the HASP TIP until an INIT arrives from the host process for the console stream. The HASP TIP then ACK's the ENQ.
- If the ACK block is received by the terminal, a buffer is constructed and the sign-on record is queued for transmission to the HASP TIP.
- If I/O errors occur or the ACK block is not received, the above starting step is repeated.
- After the sign-on record is transmitted and a positive response is received (ACK), the terminal is ready to do normal processing.
- As each individual batch device stream is then configured by the CYBER Host, and the INIT is received, the HASP TIP will allow processing of output streams. For input streams, processing will not begin until Start Inputs are received for the input device stream.

10.9.3 Sign-On Block

A sign-on block is transmitted to the HASP TIP from the HASP Work station. The data portion of the sign-on block is the sign-on record. The format of the sign-on record is:

```
Column 1        16      25
       /*SIGNON REMOTEnn  password
```

Where: NN = a one or two digit number which may be used to correlate this remote terminal with information about it in the host computer

Password may be blank

The Sign-On Block Format is:

(BSC Header)

BCB = 1010XXXX - Reset count to XXXX
FCS
RCB = 11110000 - General control record

10.9.3 Sign-On Block (cont.)

SRCB = 11000001 - Initial Sign-On
Sign-On Record
RCB = 0000000 - Transmission block terminator

(BSC Trailer)

The Sign-On record will not be sent to the Cyber Host since the Cyber Host requires a login procedure via the operator's console.

10.9.4 Sign-Off Block

The /*SIGNOFF card, when transmitted to the HASP TIP as a record in the data block, has the same effect as an EOF block. The HASP TIP will convert the Signoff Record to a BVT EOI and send it to the Host as a MSG data block.

10.10 Host Interface

10.10.1 Connection Configuration and Initialization

Once the line becomes operational, the HASP TIP will allow the Sign-On block to be sent from the HASP Work station. The Sign-On block will be acknowledged back to the HASP Work station but will not be delivered to the Cyber Host.

The Cyber Host, upon receipt of a Line Operational Service Message for an NPU port designated to service HASP Work station, will issue a Configure Terminal Service Message to the HASP TIP for the HASP Work station's operators' console. Following the INIT, an output message or start input command received by the HASP TIP will prime the HASP TIP to allow input from the operators' console. The console connection is to handle operator input and output messages to/from the connected host process.

In addition to the console, the batch device streams will be configured by the Cyber Host. Start input commands from the Cyber host will be sent to allow the HASP TIP to grant permission to input device streams for the reading of cards. Output device streams will be initiated by the HASP TIP as soon as data arrives. It is important to note that terminals configured as device type Plotter will use card punch output streams.

10.10.2 Traffic Handling

Once the necessary initialization and configuration is complete, traffic can flow between the terminal and the host. During this traffic handling period, the HASP TIP is involved in the following functions:

- Code conversion - upline and downline
- Format conversion, HASP to BVT upline, BVT to HASP downline
- Flow control upline and downline
- HASP error recovery procedures
- IVT interface for the console stream
- Input/Output streams, to/from a HASP terminal

10.10.2.1 Code Conversion

Code conversion is performed by the HASP TIP from EBCDIC to ASCII upline and from ASCII to EBCDIC downline.

Since EBCDIC is a larger code set, only the 128 - character subset equivalent to the ASCII code set is converted from EBCDIC to ASCII. All EBCDIC characters not convertable to ASCII equivalents will be converted to the ASCII blank character.

For downline data, the HASP TIP will append the transparent/non-transparent BSC envelope around the data as per the BSC envelope received from the HASP Work station.

10.10.2.2 Format Conversion To/From BVT

10.10.2.2.1 Compressed Data

10.10.2.2.1.1 Upline

The HASP TIP will convert the SCB's, Section 10.5.5, to BVT compressed format.

SCB's containing blank compression will be converted directly to BVT blank compression codes. Trailing blanks will be stripped.

The HASP terminal makes no distinction of compressed zeros as BVT does; therefore, zeros will be treated the same as any other repeated character by the HASP TIP as per conversion to BVT.

10.10.2.2.1.2 Downline

The HASP TIP will convert BVT compressed format to SCB format.

BVT compressed zeros will be expanded to the SCB format.

10.10.2.2.2 EOI/EOR BVT Codes

As defined by the Block Protocol, all data blocks between the CYBER host and the HASP TIP will be BLK blocks, except for the last block which is a MSG block representative of the end-of-information block implying no more data transmissions to follow.

10.10.2.2.2.1 Upline

The end-of-input block (MSG block) will be sent by the HASP TIP when an end-of-file block is received from a card reader stream. Contained within the MSG block will be a BVT EOI. /*EOI cards received from the card reader stream will cause the HASP TIP to send the MSG block as well. Consecutive /*EOI cards will be deleted by the HASP TIP.

/*EOR cards received from the card reader streams will cause the HASP TIP to convert it to its special BVT equivalent as well as obtaining the level number from the card and passing it also.

In addition to the preceeding, the first card in from an input device stream as well as the card after a /*EOI (assumed to be a job card) and the /*EOR card will be scanned in columns 79-80 for code conversion (26, 29). An appropriate BVT mode change will be passed upline in the data stream.

10.10.2.2.2.2 Downline

The MSG block will cause the HASP TIP to send on the associated output device stream an end-of-file block.

BVT EOR/EOI's will be converted to /*EOR's (with the appropriate level no.) and /*EOI's. For output device other than the card punch, they will be deleted and therefore be ignored.

10.10.2.2.3 Uncompressed Data

The HASP TIP will convert uncompressed SCB's to BVT uncompressed control codes and uncompressed BVT control codes to uncompressed SCB's.

10.10.2.2.4 Forms Control Codes

The BVT forms control codes will be converted to SRCBs on printer streams. For each possible BVT code, there is an equivalent pre or post print SRCB.

10.10.2.3 Flow Control

10.10.2.3.1 Downline

The FCS fields control flow on each of the streams by the use of the bits assigned to control each stream. The FCS sent by the terminal to the the TIP controls the TIP's downline delivery of records related to each stream.

The TIP will correlate the FCS bits with the applicable connection numbers and if a bit is set to the suspend transmission state, the TIP will send an upline STP block on the related connection after a timeout occurs. In subsequent upline blocks from the terminal to the TIP, a change of state of an FCS bit from suspend transmission to continue transmission for any downline stream will cause the TIP to send a STRT block upline on the related connection number to allow the host to resume delivery of downline traffic for that stream to the TIP.

For the first downline output record to a stream, if a request to initiate function transmission, transmitted from the HASP TIP is denied by the terminal, a STP block will be sent upline over the streams CN after a timeout occurs. If permission is granted, a STRT is sent.

10.10.2.3.2 Upline

The TIP, by inserting the appropriate stream control byte in the FCS field in the downline blocks to the terminal, can select which devices can input from the terminal. This is done in response to NPU regulation levels and/or host flow control on the CN's related to the terminal stream. If the TIP receives a STP or BRK block on the CN related to input from a card reader, the appropriate FCS bit will be set to suspend in the next downline data block for the terminal. If no downline data block exists to send the FCS, the TIP will send an FCS change block to the terminal, Section 10.7.3.

In response to the Stop Input command, the TIP will send an Input Stopped command to the host. If data continues to arrive from the terminal, it will be discarded. No permission to transmit will be granted by the TIP.

If the host becomes unavailable or the NPU reaches buffer saturation, the TIP will stop all input from the terminal by setting the WAIT-A-BIT in the FCS control byte.

10.10.2.4 HASP Error Recovery Procedures

The basic method of informing the receiving process of an error is through the use of the NAK block. The TIP must remember the last downline data block delivered to a terminal so that it can retransmit it should the need arise. Retransmission of a given data block is attempted three times following the initial NAK. For output blocks undeliverable to the terminal, the HASP TIP will force a Line Status SM with Line Inoperative.

The HASP TIP, after receiving the same block incorrectly four times from the terminal, will take the same action as above.

10.10.2.5 IVT Console Interface

The operator console used for HASP Workstation control will be handled as an interactive console via the IVT interface. HASP compressed data will be expanded to IVT. The following subset of IVT commands will be allowed:

- CT
- B1
- B2
- MS
- CN
- PW

In addition, line folding past the column specified in the PW command or the default value will be performed by sending multiple output lines. Auto input will be allowed. Any attempt by the application to send transparent data will force the HASP TIP to send a BRK.

HASP Downline IVT Format Effector

Pre-Print	F.E.	Action
Single Space	Space	Generate a blank line
Double Space	0	Generate two blank lines
Triple Space	-	Generate three blank lines
Start of Current Line	+	Ignore
Term Feed	'*'	Ignore
Have and Clear	1	Ignore
Null	,	Ignore

Post Print	F.E.	Action
Single Space	.	No action required
Start of Current Line	/	Ignore

No parsing of any of the characters within the output line will be performed.

10.10.2.6 Input/Output Streams To/From a HASP Terminal

The HASP TIP will never send to the HASP terminal multiple stream data within a block. The host must send to the HASP TIP the approximate desirable length of a burst of stream output for each active output stream.

The HASP TIP will insure that for output data blocks which contain a format effector only, a blank character will be inserted into the output stream so as not to confuse the HASP Workstation into passing a format effector only data block into an end-of-file block.

HASP Workstations multi-leave their input device streams. The HASP TIP will parse the input stream, sorting out each physical record to its associated CN, and send the data to the CYBER Host accordingly.

11.0 ASYNCHRONOUS TIP (ASYNCTIP)

The Asynchronous TIP supports dedicated and dial-up asynchronous lines servicing free-wheeling terminals operating at standard speeds in the range 110bps to 9600bps. The TIP provides software support for teletype, 2741 and teletype-compatible CRT's operating in an interactive mode with Cyber host applications. The interface between the host and the TIP is defined in the Interactive Virtual Terminal (IVT) section; this section concentrates on the terminal and user interface.

11.1 Terminals Supported

The ASYNCTIP is designed to provide a real-to-virtual transform for a selected group of real terminals. Additionally, a method by which the user at a terminal or a connected application may vary parameters and operating modes provides potential expansion of service to terminals which differ in detail from the real terminal types explicitly supported. The Terminal Class table in Appendix A provides a full list of default parameters for each type of terminal; the following table summarizes the terminals supported by the ASYNCTIP.

Terminal Class	Manufacturer	Model Number
1	Teletype	M33,35,37,38
2	CDC	713-10
3	Memorex	1240
4	IBM	2741
5	Teletype	M40
6	Hazeltine	2000
7	CDC	LIAT 751
8	Tektronix	4014

11.2 Operational Characteristics

The ASYNCTIP supports dedicated or dial-up lines which may be physically two or four wire, HDX or FDX. The TIP is logically HDX with contention normally being resolved in favor of input. The line speed is set up explicitly or may be determined by auto-recognition (see later). The terminal type supported is supplied by the user at the terminal or by the host software, as are any further variable parameters or modes. The TIP is prepared to receive input at all times and will attempt to deliver output whenever available, unless input is currently active, page wait is in force or an auto input input block has been output. When input is detected during output, the TIP will suspend output and will repeat this output later from the beginning of the logical line, unless the input was one of the user break commands. All input and output in the character mode is transformed between the real and virtual terminal characteristics, e.g. code conversions, format effectors, format effector delays, line delimiters, special character recognitions, etc. The transparent mode is available to suppress this transform where desired.

11.3 Auto-Recognition

The user of a 110 to 300 Baud Terminal on an auto-recognition line must depress the carriage return key after connection is established. The TIP programs the CLA to sample the line at 300 baud. The first character received by the TIP indicates the speed of the terminal as follows:

Character Sent	Character Received	Terminal Speed (baud)
X'8D	X'9C or X'8C	110
X'8D	X'E6	150
X'8D	X'8D or X'0D	300

The TIP then reports the current terminal type in the Line Operational service message and resets the CLA to sample and transmit according to the proper speed. All speeds above 300 bps require the dial-in to be made to ports designated to the appropriate speed.

11.3.1 IBM 2741 Auto-Recognition

When a user with an IBM 2741-compatible terminal dials into an auto-recognition port, the first character, a D (X'16), is sent by the terminal automatically. The CLA, sampling this at 300 bps, will cause the TIP to see X'C0 as the first character received. Since this is different than the characters received by the TIP for teletype, the TIP will set the CLA to sample and transmit at 134.5 bps and wait to sample the next character to determine if the terminal code is Correspondence Code or EBCD.

Users of a Correspondence Code terminal must depress the "C" key causing the TIP to sample X'74. Users of EBCD Code terminals must depress the "E" key causing the TIP to sample X'6B. From this, the TIP selects the code to use with this particular 2741 terminal. After depressing either key, the user must depress the ATTN key to put the terminal into a receive state.

11.4 Block Protocol Interface

The ASYNCTIP will support the following CMD blocks: IVT command, Stop Input, Input Stopped, Start Input. Refer to Table A-11 for a description of the format of these commands.

The host sends Stop Input and Start Input. The TIP responds with a BACK and, in the case of Stop Input, an Input Stopped is also sent. The host responds to an Input Stopped with a BACK.

11.5 Real Terminal Transforms

The following section describes the relationship between the Virtual Terminal and Real Terminals supported by the ASYNCTIP.

11.5.1 Parity Options

Non-transparent mode output: The host will send down 8-bit character. Except for virtual characters, the upper bit (bit 7) can be anything and will be ignored. Namely, before the tip translates the character, if translation is needed, the bit 7 will be suppressed. Likewise, if translation not needed, the bit is suppressed before going to the CLA for outputting. This is true for all parity settings. If by accident the upper bit is set in the two virtual characters, they will not be recognized.

Transparent mode output: The host can send down anything for bit 8 of all parity settings except "none". The seventh bit is set zero for parity = zero, and is set correctly for parity = odd or even. For parity = none, the entire 8 bits as sent from the host is sent through the CLA to the terminal.

Non-transparent mode input: The TIP will send the host 8 bit characters with the 8th bit always set = 0. This is true for all 4 parity settings.

Transparent mode input: For parity = none, the host receives the 8 bits just as they came in on the line. For the other three parity settings, the 8th bit is set to zero.

11.5.2 Character Mode Input Processing

Logical/Physical Lines

A physical line of input is defined to be an input line that ends with the terminal's line feed/new line delimiter or current page width reached. A logical line of input is defined to be an input line that ends with the terminal's carriage return/EOT delimiter. TIP processing of logical/physical lines will be described later.

Type Ahead

The TIP will always operate in the Type Ahead mode. Output will only be started if input pauses at the end of a logical line for longer than 200 ± 100 ms. If input clashes with output or starts at any time that output is active, the output will be halted and input may proceed. In the case of a true HDX line, the input may have been garbled and must be cancelled and repeated. If the user is in Auto-input mode (see later), he has the responsibility for not typing ahead.

Default Block Mode Support

The TIP, by default, will be capable of Block Mode terminal support. This means that input then has priority over output. At the end of logical/physical lines, a 300 millisecond timer will be started by the TIP. If any new input arrives from the terminal, the output side of the TIP will be locked out. Output data from the host will stay queued for the terminal; any canned responses, e.g., echoing carriage return to line feed sequences, will be discarded.

11.5.2.1 Keyboard Input

Parity Checking/Stripping

The TIP will service the input data stream in the default mode for the terminal class. For no parity checking, as data characters arrive from the terminal, the parity bit will be stripped. The user can cause parity checking to be done by the TIP using the <CTL> PA command. The CLA will be set to the terminal's currently defined parity. As input characters arrive, the CLA will automatically check and strip parity from the data characters. If a parity error occurs, the TIP will store the bad character in the input data buffer, and then mark the DBC that a parity error exists within the data block.

NUL(s)/DEL(s) Processing

The TIP will always strip (NUL(s) and DEL(s) from the input data stream as it receives them.

Character Translation

The TIP will convert the real terminal's input characters to 7 bit ASCII (parity bit = 0) as it receives them.

Backspace Processing

The TIP will be capable of detecting the terminal's currently defined backspace character. Input characters will be discarded by the TIP as per the number of backspace characters. Backspacing to the beginning of a line causes the line to be deleted. Backspacing through the beginning of line causes the backspaces to be ignored. Since the TIP may ship physical lines to the host prior to the end of logical line, all references to beginning of line in the preceding should be understood to refer to physical lines. Additionally in the special case that the current page width is reached prior to the end of a physical line indicators, backspacing is not permitted into the block which will have been released at that point. When using the APL code set, the backspace will only be actioned if the APL special mode is not in effect. Otherwise the backspace will be treated as data.

Auto Input

The TIP will be capable of auto input; e.g., capable of inputting into a previous output data block. Logically, the output data block will be chained to the front of newly arriving input data and will be sent to the host as part of the input data stream. Auto input only applies to downline MSG blocks and will be ignored if specified in a BLK block.

11.5.2.1 Keyboard Input (cont.)

After the auto-input block has been delivered, the TIP will not deliver any more output until an input has been received. Otherwise the input from the terminal may not be synchronized with the correct auto input. Only the first twenty characters of output data will be returned. Note that format effectors are stripped prior to return of data. When the user wishes to override the auto-input and substitute his own input, this may be achieved by entering a cancel input line character followed by a CR/EOT; this will cancel any data entered by the user and the auto-input. The TIP will remain in input mode until a subsequent non-cancelled input is received.

Line Feed/New Line Processing (Physical Line)

The TIP will discard the line feed or new line character. In the case of line feed, a carriage return sequence will be sent back to the user. The currently assembled block will be sent to the host as a BLK block containing a single physical line.

Carriage Return/EOT Processing (Logical Line)

The TIP will discard the carriage return or EOT character. A line feed sequence or new line sequence, respectively, will be sent back to the user if the mode permits. The currently assembled block will be sent to the host as a MSG type data block. Null logical lines will be discarded only if they are used as a page turn.

Start of Text Processing

The STX character will be discarded when it occurs as the first character of a logical line.

Cancel Processing

The TIP will detect the terminal's currently defined cancel character preceding the end-of-logical-line indicator. The TIP will discard the current input logical line. If any part of the logical line has already been dispatched, then a cancel MSG block will be sent to the host. When using the APL code set, the cancel character will be recognized only if not in APL special mode. Otherwise, the cancel character will be treated as data.

Upper/Lower Case Shift Processing

For the 2741 terminal, the TIP will keep track of shifts between lower and upper case to ensure correct translation to ASCII. The TIP will assume the lower case condition at the beginning of each input logical line.

11.5.2.1 Keyboard Input (cont.)

Maximum Line Width

If the current line width is reached prior to detection of a physical line terminator, the partially assembled physical line will be sent to the host as a BLK block. In the case where the line width is zero, the maximum line width will be 140 characters.

11.5.2.2 Paper Tape Input Character Mode

The TIP will be capable of reading a paper tape of input data without forcing the user to specifically enter into a paper tape mode. To accomplish this, <X-OFF> characters should not exist on the paper tape or, alternatively, the user must turn the reader back on after each <X-OFF>.

For those users who have paper tape with <X-OFF> characters on the paper tape, paper tape input should be declared. In both keyboard and paper tape modes, the TIP will detect end of physical/logical lines and process them accordingly. It will then check the next character which arrives; NUL's/DEL's are always stripped. If the character following a carriage return/EOT delimiter is a line feed/new line, it will be discarded and not actioned by the TIP. Similarly, if following a LF/NL delimiter the TIP inputs a CR/EOT character, it will be discarded and not actioned.

In keyboard mode all <X-OFF> characters will be treated as data. In paper tape mode an <X-OFF> will be treated as data unless it is at the end of a logical line. In this case, the <X-OFF> will be discarded. If the <X-OFF> stopped the tape, whether or not it was at the end of the logical line, the TIP will send an <X-ON> after a MSG block from the application has been processed and there is no further output queued for this terminal.

11.5.3 Transparent Mode - Keyboard/Paper Tape Input

Input data received by the TIP will be sent to the host with no character translation performed. Data will be sent as BLK type blocks when system default block size is exceeded, until any one of the user transparent delimiters is reached. The data will then be sent as a MSG type block. In transparent paper tape mode where a special character or character count is specified, receipt of an <X-OFF> which stops the tape will result in an <X-ON> sent to the terminal. The <X-OFF> character and all previously input data will be sent to the host in a BLK block. When an <X-OFF> is input and it is the Special Character or a time out is the delimiter, and the tape stops, then the previous input will be sent in a MSG block, transparent mode will be terminated and no <X-ON> will be sent. If the input does not stop at the end of the transparent input, the remaining data will be processed in character mode with the possibility that some of the initial data might be lost. The number of significant bits received from the terminal may range from six

11.5.3 Transparent Mode - Keyboard/Paper Tape Input (cont.)

to eight depending on terminal type and parity setting. When parity checking is set to mode N, the parity bit will be passed as data. All information is passed right justified in the byte with non-significant high order bits set to zero.

When transparent mode ends, the TIP returns to keyboard/character mode.

11.5.4 Character Mode - Output Processing

Output delivered to the TIP can have multiple logical lines within the data block. End of logical line deli niters, as well as certain embedded format characters, will be translated to the Real Terminal's format sequence where possible. See Table 11-1 for a description of embedded format effectors to real terminal transforms. The TIP will monitor for input or break during output to allow the user to stop the output and perform certain actions.

During automatic line folding or end of logical line processing, the TIP will insert into the output stream the terminal's currently defined number of NUL characters as time fill.

When providing paging of output, any input will cause the TIP to reset the page count to the top of page. Therefore, the user is responsible for inputting data which may cause subsequent output to be improperly positioned on following pages.

Where FE's cause the terminal to be positioned over a page boundary a new page sequence only will be output. This feature can be disabled by setting the page length to zero.

11.5.4.1 Printer Output

Character Translation

All output data delivered to the TIP from the host application will be in ASCII. The TIP will convert the ASCII data to the real terminal's character set.

Format Effectors/Line Folding

Each logical line of output may contain a format effector as the first character. A bit in the Data Block Clarifier defines whether or not these format effectors are present. Pre-print single spacing is assumed if the format effectors are not present or are not defined. The format effectors cause pre or post format control. See Table 11-2 for the list of format effectors. The TIP will convert the format effectors to the real terminal's format sequence. Where applicable, the TIP will do automatic line folding by outputting the terminal's line feed/carriage return sequence with the appropriate number of NUL's.

2741 Upper/Lower Case

Current upper/lower case will be retained by the TIP for output. Appropriate upper/lower case shift characters will be inserted by the TIP as a function of ASCII code translation to the 2741 terminal's character set.

11.5.4.2 CRT Output

CRT output will be processed like printer output except that the TIP will do page waits when the option is selected. After a page wait, the user enters a null line to obtain the next page. The TIP will discard the null input line in page wait situations. If a non-null input line is typed by the user, it will be treated as a page turn and will be passed to the host or processed by the TIP if it is a command.

When the page wait option is selected, the page output will be one line less than the current page length to allow space for the user input. On hard copy devices or when the current page length is zero, the page wait option has no effect. If a top of form is received in the output stream and the page is not full, the text OVER.. will be output to tell the user to enter a page turn.

11.5.4.3 Paper Tape Output

When the output device is explicitly paper tape, the TIP will insert into the output stream an <X-OFF> character (DC 4) followed by three NUL's, at the end of each logical line sequence if it contains post print format effectors. Line folding will be performed as for printer output.

11.5.5 Transparent Mode - Output Processing (Printer/CRT/Paper Tape)

Transparent mode is provided to allow the user application to inhibit the TIP transforms. In this mode the user application is responsible for all data formatting.

The application can permit page waits by sending a synchronous command to the TIP. The TIP will do page waits at the end of every MSG type data block. Page wait responses are interpreted the same as Character mode page waits by the TIP.

11.5.6 Logical Line Aborting - Character/Transparent Mode - Keyboard/CRT/Paper Tape Output

During output, the TIP will monitor for a break or input. The user can cause the current logical output line to be terminated by entering the abort line character followed by a CR/EOT. Output will then continue with the next logical output line.

11.6 User Interface - Data

11.6.1 User Input

input = [(<STX>) (<logical line>)<LLDLM>(<X-OFF>)
(<STX>) <CTL><Command><LLDLM>
<transparent data> <DLM>(<X-OFF>)]

The terminal user enters input as the basic unit that he wishes processed by the computer. If page mode is in effect input may be treated as a request for next page.

<logical line> = [<physical line><LF> (<CR>)]$_{0-N}$ <physical line>

Character mode inputs are <logical lines>.

<LLDLM> = [<CHARSEQ>
<EOT>]

<CTL> = <Control Character>
{This is defined by terminal type and may be changed by user.}

<Command> = <Terminal Definition Commands>
{See Section 11.7}

<Transparent Data> = [[<byte>]$_{1-n_1}$(X-OFF)]$_{1-n_2}$
Where n_1 and n_2 are positive integers

<DLM> = [200 msec. timeout
character count
delimiter byte]
{Any of these may be specified by the user and they may be used in combination.}

<physical line> = [<Character>]$_{1-m}$
{Where "m" is the terminals physical line width defined by user}

11.6.1 User Input (Cont'd)

<LF> = x'0A
<CR> = x'0D
{When translated from the user's code set to no parity ASCII}

<CHARSEQ> = <¬LF> [<DEL>]$_{0-m}$ <CR>

<EOT> = x'04
{When translated from the user's code set to no parity ASCII}

<byte> = <bit pattern>
{Any bit pattern capable of being received from the terminal.}

<character> = <128 ASCII CHAR SET>
{When translated from user's code set to no parity ASCII.}

<DEL> = {idle fill}

<X-OFF> = {a character which turns the paper tape reader off}
Meaningful only when input device is paper tape (IN = PT or XP)

11.6.2 User Output - This section describes the user output sequences in the format that is sent to the terminal (i.e., after the IVT transforms, paging, etc., have been performed).

<output> = <page>
<transparent data>

$$\text{<page>} = (\text{<FF>})\left[(\text{<PREFE>})\ \text{<physical line>}\left[\begin{array}{l}\text{<LF>}[\text{<idle>}]_{0-L}\ \text{<CR>}\ [\text{<idle>}]_{0-M} \\ (\text{<POSTFE>})\ (\text{<X-OFF>}[\text{<idle>}])\end{array}\right]_{1-N}\right]_{3}$$

{Where L and M are the line feed and carriage return idle counts defined by terminal class or Terminal Definition Commands and "N" is the Page Length in physical lines. If the Page Length is set to zero, no form feeds <FF> will be inserted by the TIP and the Page Wait feature will have no effect.}

<transparent data> = [<byte>]$_{1-N}$
{Where "N" is an installation parameter "maximum block size"}

11.6.2 User Output (Cont'd)

<FF>	=	{"Home and Clear"}	{The "Home and Clear" will differ depending on terminal class. The FF will not be sent if the Page Length is zero.}
<PREFE>	=	[Single Space Double Space Triple Space Start of Current Line Home Home and Clear]	{These pre-print format effectors will differ depending on the terminal class}
<physical line>	=	[<character>]$_{0-N}$	{Where "N" is defined by the Page Width}
<LF>	=	{x'0A translated from no parity ASCII to the user's code set causes the cursor or platten to move down one line.}	
<idle>	=	[<DEL> <NULL>]	
<POSTFE>	=	[Single Space Start of Current Line]	These post print format effectors will differ depending on terminal class.
<X-OFF>	=	{A character that turns the paper tape reader off}	Used only when the output devices is paper tape (OP=PT) and if data is not transparent.
<byte>	=	{Any bit pattern capable of being received by the terminal.}	Depending on the parity option selected, it may be seven bits plus parity or all eight bits as received from the host.

11.7 User Interface - Control

When the connection between the user and the host is initially established, the terminal is configured with a set of default parameters. Host software may modify these parameters at any time. The user has the further capability to modify the configuration of the terminal, its operational modes, and the management of the up-line and down-line data streams.

The IVTIP will detect each operational control message and parse it for correctness. If the user input is correct, the TIP will change the appropriate parameter(s). The TIP responds to the user with a canned <CR><LF>. If the user input is incorrect, the TIP will respond with the canned message <CR><LF> ERR.. <CR><LF>. To enable the TIP to detect operational control messages, each message must start with the defined Control Character and be contained in one logical input line. Most commands concerning output do not affect any output blocks in progress, but become effective on subsequent blocks.

Terminal Class

Command: ⟨CTL⟩ ⟨TC⟩ = [1 2 3 4 5 6 7 8] ⟨CR⟩

Function: To establish the terminal class and default parameters as defined in the Terminal Class table of Appendix A.

Page Width

Command: ⟨CTL⟩ PW = NNN ⟨CR⟩

Function: To establish the physical line width in characters for output and maximum character block size on input. For non-transparent output blocks, the TIP inserts the character sequence defined for the terminal class to move the carriage or cursor to the next line at the point where the number of characters to be transmitted equals the page width. The parameter NNN varies between 0 and 255, where 0 means "new line" is never inserted.

11.7 User Interface - Control (Cont'd)

Page Length

Command: <CTL> PL = NN <CR>

Function: To establish the number of physical lines in a page for output. The TIP inserts the character sequence defined for the terminal class to advance the carriage or cursor to the next page when the number of physical lines transmitted equals the page length. Also, if the Page Wait feature is selected and OP=CRT, the TIP will wait for an operator input before continuing. The parameter NNN varies between 0 and 255 where 0 means no paging.

Check Parity

Command: ⟨CTL⟩ PA = [Z O E N] ⟨CR⟩

Function: To inform the IVTIP which type of parity to expect on input and generate on output. See description under Section 11.5.1.

Cancel Character

Command: ⟨CTL⟩ CN = a ⟨CR⟩

Function: To establish the character to be used to cause the logical input line in process to be deleted.

Backspace Character

Command: ⟨CTL⟩ BS = a ⟨CR⟩

Function: To establish the character to be used to cause the previous input character to be deleted from the input buffer in process.

11.7 User Interface - Control (Cont'd)

Abort Output Line

Command: <CTL> AL = a <CR>

Function: To establish the character to be used to cause the rest of the present output logical line to be discarded.

User Break 1 Character

Command: <CTL> B1 = a <CR>

Function: To establish the character to be used to generate an upline BRK with a "User Break 1" code.

User Break 2 Character

Command: <CTL> B2 = a <CR>

Function: To establish the character to be used to generate an upline BRK with a "User Break 2" code.

Control Character

Command: <CTL> CT = a <CR>

Function: To establish the character to be used to enter operational control messages.

Carriage Return Idle Count

Command: <CTL> CI = [CA / nn] <CR>

Function: To establish the number of idle characters to be inserted in the output stream following a CR. The use of CI = nn for these terminals overrules the default known to the TIP and CI = CA restores the default

11.7 User Interface - Control (cont.)

Line Feed Idle Count

Command: <CTL> LI = [CA / nn] <CR>

Function: To establish the number of idle characters to be inserted in the output stream following a LF. The use of LI=nn overrules the default known to the TIP and LI = CA restores this default.

APL Code Set

Command: <CTL> AP = [Y / N / S] <CR>

Function:

Where Y = APL code set
N = terminal code set
S = APL special code set

To allow the user to switch to/from the APL code set from/to the user's normal code set. (Valid only for certain terminal types. See Table 11-1.) When the APL special code set is chosen, input processing is changed to inhibit action on backspace and cancel input, which are normally treated as control characters in the APL code set.

Transparent Text Delimiter

Command: <CTL> DL = (Xhh), Cnnnn), (TO) <CR>

Where hh are two hexadecimal digits representing the terminal-originated character selected as a delimiter, "nnnn" is a character count (1 to 4096 decimal), and TO is input character timeout. Each field is optional, but at least one must appear. Parameters may be entered in any order and trailing commas may be deleted.

Function: To establish the transparent text delimiter set. The timeout value is 300 ± 100 msec.

11.7 User Interface - Control (Cont'd)

Input Device

Command: ⟨CTL⟩ IN = [KB / XK / PT / XP] ⟨CR⟩

Function: To allow the user to specify the input device as a keyboard or paper tape reader and whether or not transparent mode is in effect. Note that paper tape input is allowed in keyboard mode, but that the TIP does not send the <X-ON> characters to start the paper tape reader.

Output Device

Command: ⟨CTL⟩ OP = [PR / DI / PT] ⟨CR⟩

Where PR = Printer,
DI = Display (CRT),
PT = Paper Tape Punch.

Function: To allow the user to specify the output device as printer, CRT display, or paper tape punch. Printer and CRT display are functionally equivalent except for page wait. The user may punch a paper tape in any mode, but the TIP will only provide the <X-OFF> character if OP = PT and if data is not transparent.

Echo Mode

Command: ⟨CTL⟩ EP = [Y / N] ⟨CR⟩

Function: To allow the user to specify where input character echoing will take place. EP = N implies the terminal is doing its own input echoing and EC = Y causes the TIP to set the CLA to provide character echoing.

Operator Message

Command: ⟨CTL⟩ MS = text ⟨CR⟩

Function: To allow the user to send message text to the local operator. The maximum number of text characters accepted is the current line width or 50 characters, whichever is less. Any text entered in excess of the maximum will be truncated.

11.7 User Interface - Control (cont.)

Page Wait

Command: PG = [Y / N]

Function: Selects page waiting on or off. Allows the user to control output to display one page at a time. Note: this command has effect only for OP = CRT.

11.8 Input/Output Control

The three special characters which may be defined to abort output line and cause user Break 1 and user Break 2 must all be followed by a CR to be effective. For FDX lines, the special characters may be entered during output; for HDX lines, a break must first be entered to cause output to stop and the special character to be recognized. When a break is entered the user may enter data or commands. At the first opportunity after a downline reset has been received, the TIP will resume output unless commanded to do otherwise.

In certain situations, the NPU is forced to reject input. This occurs when the NPU runs low on buffers, the network block limit is exceeded, a Stop Input command is received, or the NPU loses contact with the host. If the reason for rejecting input is because the NPU lost contact with the host, then at the time the condition is detected in the NPU, each connected console terminal will be sent a canned message to inform the user of the situation. The canned message is:

<X-OFF> <NUL><NUL> <CR> <LF> <BELL< >BELL< >IDLES> N
INPUT STOPPED <user text> <CR> <LF> <IDLES>$_N$

<User text> is defined as up to 30 characters of user supplied informative text which may be changed via the 255X console. Default for user text is 'HOST UNAVAILABLE'. If input is received after the user has been notified of a loss of contact with the host or if any of the other reasons for rejecting input are detected, the input will be discarded and the user will be notified with the following canned message:

<X-OFF> <NULL> <NUL> <CR> <LF> <BELL> <BELL> <IDLES>$_N$
REPEAT... <CR> <LF> <IDLES>$_N$

This message is repeated every time any further input is attempted until the situation is relieved. At the time communication with the host has been restored, the user will be notified via the following canned message:

<CR> <LF> <IDLES>$_N$
INPUT RESUMED <CR> <LF> <IDLES>$_N$

11.9 Modem/Line Control

Receive Carrier - The terminal turns on the receive carrier on the first keystroke and keeps it on until the terminal outputs its end of message delimiter.

Transmit Carrier - The TIP will turn transmit carrier on (RTS) for the duration of an output message delivery to the terminal, and turn it off immediately following the last character outputted, or in response to a BREAK received from the terminal.

Received Break - A break received from a terminal will manifest itself in the following ways:

> Line Speeds < 600 bps: Spacing condition on receive data line while output is being delivered for at least 200 ms.
>
> Line Speeds > 600 bps: Spacing condition on supervisory receive channel, while output is being delivered for at least 200 ms.

Transmitted Break - A break transmitted to a terminal will be sent using the same channels in the downline direction as specified for the line types above. A spacing condition 200 - 400 ms on these channels will specify a break.

(Downline Only)

TERMINAL CLASSES

	Virtual Interface	1 TTY 33, 35, 37, 38	2 CDC 713-10	3* Memorex 1240	4* IBM 2741	5 TTY 40	7 CDC 751-1	6 Hazeltine 2000	8 Tektronix 4014
Carriage Return	<CR>	CR	CR	CR	NL	CR	CR	CR	CR
Line Feed	<LF>	LF	LF	LF	LF	LF	LF	LF	LF

*Supports APL Code Subset

Table 11-1 Virtual to Real Terminal Embedded Format Effector Transforms

PRE PRINT	* *	TTY 33, 35, 37, 38	CDC 713-10	Memorex 1240	IBM 2741	TTY 40	CDC 751-1	Hazeltine 2000	Tektronix 4014
Position to start of next line	SPACE	CR LF	CR LF	CR LF	NL	CR LF	CR LF	CR	T
Position to start of current line	+	CR	CR	CR	BS (N) *	ESC G	CR	-	
Position to top of form (cursor home)	*	CR 6 LF's ***	EM	CR 6 LF's ***	6xNL ***	ESC H	EM	~DC2	B
Home cursor and clear screen	1	CR 6 LF's ***	CAN	CR 6 LF's ***	6xNL ***	ESC R	CAN	~FS	
Null	,	-	-	-	-	-	-	-	D
Double space	0	CR 2 LF's	CR 2xLF	CR 2xLF	2xNL	CR2xLF	CR2xLF	-	
Triple space	-	CR 3 LF's	CR 3xLF	CR 3xLF	3xNL	CR3xLF	CR3xLF	-	

POST PRINT									
Single space	.	CR LF	CR LF	CR LF	NL	CR LF	CR LF	-	T
Return to start of current line	/	CR	CR	CR	BS (N) *	CR	CR	-	B

* The number of backspaces is a function of current cursor position

** Any other format effector will be treated as space

*** When PL ≠ 0, the IVTIP will calculate the difference between end of page and current print position and space forward the appropriate number of lines

Table 11-2 FORMAT EFFECTORS FOR REAL TERMINAL TRANSFORM

12.0 MODE 4 TERMINAL INTERACTIVE PACKAGE

12.1 Introduction

The Terminal Interface Package (TIP) for the Mode 4 terminals provides a set of procedures for the interchange of data between a host processor and a Mode 4 terminal. This TIP is used for both interactive display devices and remote batch card readers and printers.

12.2 Definitions

Considerable differences exist in the terminology associated with Mode 4 devices. The following conventions will be used:

Nomenclature used in this specification	Mode 4 Nomenclature	Mode 4C Nomenclature
NPU	data source	control station
cluster address	site address	terminal address
cluster controller	equipment controller	station
terminal address	station address	device address

12.3 Overview

The Mode 4 TIP interfaces with a host process using the CCP R4 protocol. The data contained within the interchanged blocks will conform with the Interactive Virtual Terminal (IVT) interface or the Batch Virtual Terminal (BVT) interface.

The interface to the terminals complies with the Mode 4A or Mode 4C standards. However, not all features of the Mode 4 protocols nor all features of supported terminals will be used in this implementation.

The TIP will be insensitive to line speeds and will support synchronous lines operating at up to 9600 baud. These lines may be dedicated (with or without a tranceiver) or switched (dialup) with a modem. Further, the lines will be considered half duplex, i.e. the TIP is either transmitting to the line or receiving from the line but not both simultaneously.

Each line may have more than one cluster and each cluster may have more than one terminal. Lines with multiple clusters must be dedicated. Where multiple terminals are on a line, the TIP services each terminal in sequential order without priority.

The TIP will perform auto-recognition when requested by the host. This procedure will determine the code set of the terminal (ASCII or External BCD) and mode (Mode 4A or Mode 4C).

12.3 Overview (cont.)

In addition, the cluster address, terminal address, device type, and limited terminal class information will be reported. This information will enable the host to correctly configure the cluster detected by the auto-recognition procedures. Multi-cluster auto-recognition is not supported.

The Mode 4 TIP will support the remote batch terminals as separate but dependent devices. The dependencies are reported to host on demand when a conflict has or will be likely to occur.

All terminal polling is performed by the TIP. The host requests that input be accepted but the TIP controls the actual polling for data.

The TIP performs recovery for line or terminal errors. All errors from which an immediate recovery is not possible are reported to the host.

12.4 Host Interface

The host interacts with the TIP via the block protocol and the IVT or BVT interface. The written text in each section provides a description of the NPU-host interface.

The TIP processes each line as independent data channels. Each terminal on a line is checked for work in the order the terminals were configured. This method allows each terminal to be processed in order without priority. The card reader and printer of the 200UT are treated as separate terminals in this scheme.

12.4.1 Interactive Interface

The Interactive Virtual Terminal interface to the Mode 4 TIP will provide support of displays attached to synchronous lines. The configuration may be multi-cluster and each cluster may be multi-terminal. The 200UT display is also supported by the IVT interface but several additional features will exist for control of the card reader and printer.

The terminals are activated either by delivery of an output message or a START INPUT command. For a Mode 4A terminal a START INPUT command will invoke a write to the terminal to reset the cursor to the left most character position. This write is intended to clear the 200 UT transmission buffer of a previous card or print block.

Polling for input continues until the terminal is deleted, an error occurs, buffer regulation occurs, logical link regulation occurs, or a STOP INPUT command is received. INPUT STOPPED command is sent in response to the STOP INPUT command.

A STP is sent whenever a communication error is detected.
The subsequent STRT is sent when the error condition is resolved.

For the 200UT, the use of the display will cause the card reader and printer connections to send STPS. These interchannel interactions are intended to signal the use of the 200UT transmission buffer which is shared by the display, card reader and printer. The host will receive STRTS on the card reader and printer to signal the end of the interactive transactions when the STOP INPUT command is received on the console connection.

INTENTIONALLY LEFT BLANK

12.4.2 Card Reader Interface

The Mode 4A card reader is activated by sending a command to the TIP to start accepting input. The TIP sends card reader data, transformed to the BVT specifications, to the host. Each block of data is reported to the host as a BLK block until an EOI card (6/7/8/9 punch in column one) is detected. The block containing the EOI is reported to the host as an MSG block. Upon sending an MSG block, no further input is accepted and an input stopped command is sent to the host.

The MSG block will contain the data up to and including the EOI card. Subsequent EOI cards are discarded until the first non EOI card is sensed. The data following the last EOI is considered part of the next message. (It is possible that a single block from a Mode 4 device will contain more than one job. Each job will be reported as a MSG block).

The input stopped command will be sent following the last data sent to the host from any block containing an EOI. An input stopped command is also sent if no further cards are present in the input hopper. A reason code for the input stopped command to distinguish the two cases is supplied.

An upline STP will occur on the card reader channel indicating that down line data or commands must not be sent or must be repeated if already sent and not acknowledged with a BACK. This STP is invoked by the host or the terminal operator whenever the display is in use.

A STP will also occur when the TIP detects a communication error with the terminal. A subsequent STRT will occur when the error is resolved.

INTENTIONALLY BLANK

12.4.3. Printer Interface

The Printer may be activated whenever the host chooses by sending down line data. The printer connection is considered active until a MSG block is sent by the host or until the display is used. The TIP sends the data, transformed from the BVT specifications to the printer. Each block when delivered correctly to the terminal is acknowledged to the host with a BACK.

A STP will be sent for the printer whenever data or commands cannot be actioned because the display is in use. This stop occurs either when the host sends data to the display or remote operator interrupts a batch operation. The host must prepare to resend any data or commands not acknowledged with a BACK.

A STP will also occur whenever an irrecoverable error is detected on the device.

A BRK will be sent to the host whenever the printer is found to be not ready while attempting to deliver output.

12.4.4 Block Protocol Interface

The Mode 4 TIP will utilize four CMD block types in its interface to the host. These are the IVT command, Start Input, Stop Input and Input Stopped. (See Table A-11.)

12.5 Real Terminal Transforms

The Mode 4 TIP will transform the data according to the specifications for the IVT or BVT interfaces. These transforms will produce the requested effects wherever terminal features allow.

12.5.1 Interactive Virtual Terminal Transforms

12.5.1.1 Downline IVT Transforms

The TIP will transform the downline IVT data for the terminals described in Table 12-9. This data will be processed as specified in Tables 12-10 and 12-11. Additional information is provided in the DBC field of the data blocks. This information consists of the following:

Downline Flags	Purpose
auto input	the down line data is returned to the host
transparent	the data is not to be transformed by the IVT interface
FE	Format Effectors are not present in the data

Upline Flags	Purpose
transparent	the data has not been transformed by the IVT interface

The cursor is returned to the left margin following each output of a logical line. If more than one logical line exists in a Block, the logical separator <μs> is actioned as a carriage return. This assures that output data is compatible whether logical lines are blocked or not. The fact that the cursor is returned to the left margin after each output is taken into account when actioning the format effectors. Any format effector which is undefined is actioned as a preprint position to start of next line.

Any ASCII control character will be replaced with blanks, and for those terminals with fewer than 96 characters, lower case will be folded into upper case.

If an error is detected in the IVT data, a BRK is sent to the host.

12.5.1.1 Downline IVT Transforms (cont.)

The actioning of the preprint format effectors to position to top of form (*) or to home cursor and clear screen must be the first of a transmission to the terminal. Thus, if more than one logical line exists in a block from the host, it will be fragmented into as many separate transmissions as necessary.

12.5.1.2 Upline IVT Transforms

The input from the terminal may include multiple logical lines separated by CR's. Each logical line is sent to the host as an individual MSG block. No other transforms are performed on the data except that escape codes are not counted in the calculation of the cursor position.

12.5.1.3 Auto Input

The TIP will deliver output to the terminal and retain the data buffers whenever the auto input flag is set in the DBC of a MSG block. The subsequent input from the terminal is attached to the end of the saved data and returned to the host. The Format Effector is deleted from the auto input if present. If more than one logical line is present in a MSG block specifying auto input, a BRK is sent to the host. If more than one logical line is received from the terminal, the first is appended to the saved auto input. All subsequent logical lines are transmitted to the host as received.

The saved auto input data may be cancelled by the terminal operator by entering a logical line ending with the cancel control character (CN). The cancel request must be the first logical line of the transmission and subsequent logical lines are sent to the host as received.

An input logical line other than an IVT command must be received to satisfy the auto input request before a subsequent output will be sent to the terminal. The cancelled line is not sufficient to satisfy the auto input request.

12.5.1.4 Transparent

The IVT transform is not performed on transparent data. However, the Mode 4 frame control is added to the data but no code conversion is performed. The parity bit for each character is also added before the data reaches the line.

Auto input and page wait are supported for transparent data. However page length calculations are not supported; page wait occurs following each MSG block only.

12.5.1.4 Transparent (cont.)

Format effectors are not supported and each output is assumed to be a write with an E4 terminator. The clear write and reset write features of the Mode 4 protocol are not supported.

Transparent input applies only to the first input transaction following selection of the feature. The Mode 4 frame control characters are removed but no other translation occurs. The cursor is not repositioned to the left margin following each input or output and the keyboard is not unlocked. Since any further polling would result in retransmission of the previous data, polling ceases. The host must request that polling resume by sending output or a Start Input command.

Transparent mode for a Mode 4A terminal will be illegal and a BREAK will be sent if detected.

12.5.1.5 User Break One and Two

The IVT interface allows the terminal operator to request a BRK to signal the user break one or two condition. This BRK is invoked by entering a logical line with either the user break one or two character as the only data. The interpretation of these BRK's is application development.

12.5.1.6 Page Wait

The page wait feature of the IVT interface provides a method of assuring that output will be delivered at a readable rate. The data sent from the host is displayed on the screen until the end of page is reached.

The calculations for page size are based on the page width (PW) and page length (PL) parameters. These parameters are assumed to be the actual size of the terminal. In particular it is assumed that the hardware provides an automatic carriage return at the page width boundary.

Page calculations take into account line folding and a folded logical line may span a page boundary. The clear write/reset write format effectors terminate a previous page. If the previous page is not full, the message OVER.. is sent to the terminal. A page is full whenever the page length less one line is filled.

Page turning is accomplished whenever the terminal operator enters an input line. Any page turn prompt consisting of a nul line or a line consisting of only the control character is not sent to the host.

12.5.1.7 Code Conversions

In character mode, all IVT data is code converted whether the terminal is ASCII or External BCD. The ASCII Mode 4A translation includes folding lower case into upper case and substitution blanks for any control codes. The Mode 4C translation provides the substitution of blanks for the control codes but allows the transmission of the lower case codes.

Table 12-10 Downline IVT Format Effector Transforms

PREPRINT DESCRIPTION	Effector Code	Transform (all devices)
Position to start of next line	SPACE	nul
Position to start of current line	+	nul
Position to top of form (3) (cursor home)	*	C (2)
Home cursor and clear screen (3)	1	12 (2)
Null	'	nul
Double space	0	CR (1)
Triple space	-	CR, CR (1)

POST PRINT

Single space	•	nul
Return to start of current line	/	nul

1) The symbol CR represents the two code pair 1B, 41 (in External BCD it is 3E, 41)

2) These values are the MTI codes of the Mode 4 protocol. (See Section 12.5.1.1). Where C = reset write and 12 = clear write.

NOTE: The cursor is returned to the left margin following each input and each output of a logical line. Thus, the format effectors are actioned with the assumption that the cursor is positioned at the left hand margin of the next line.

Table 12-11 Downline IVT Transforms

IVT Interface		Transform (all devices)
Carriage Return	<CR>	CR(1)
Line feed	<LF>	nul
Logical Line Separator	<US>	CR(1)

(1) The Symbol CR represents the two code pair 1B, 41 (for External BCD it is 3E, 41).

INTENTIONALLY LEFT BLANK

12.5.1.8 Cursor Control

The TIP will return the cursor to the left most character on the next line following the end of each input or output line. A blank line will occur on the screen if the ETX symbol (◁) from an input request is in the last column or if the output ends in the last column. This is required to allow positioning of the send index for the next line.

Whenever the send index terminator is detected as the first two characters (an escape, control code pair) it is deleted before sending the message to the host.

Cursor positioning to the left margin will be accomplished in one of three ways depending on terminal class. Terminal class is initially configured and can be changed with the TC IVT command from the terminal user or application. For the 214 and 200 UT terminals, each input is replied with an output of a sufficient number of blanks to move the cursor to the left hand margin of the next line. Each output is padded with blanks to also move the cursor to the left hand margin of the next line.

For 731/732 and 734 terminals, each input is replied with only a line clear to unlock the keyboard and each output is terminated with a line clear.

Finally, for the Mode 4C devices (711 and 714) each input is replied to with a carriage return, backspace sequence. Similarly, each output is terminated with a carriage return, backspace sequence. In either case the intent is to place the cursor at the left hand margin of the next line.

12.5.1.9 Message Type Indicators (MTI)

The codes below are in hexadecimal notation, exclusive of parity. The type of MTI code affixed to output data is a function of the format effector in character mode only. For transparent mode the MTI is always write.

12.1.5.9 Message Type Indicators (MTI)(cont.)

MTI In Transmitted Block X'	MTI In Received Block			
	REJECT X'18	ACK X'06	ERROR X-15	READ X-13
05 Poll	X		X	X
12 Clear Write	X	X	X	
0C Reset Write	X	X	X	
11 Write	X	X	X	
07 Alert		X	X	
31 Configuration	X		X	X

POLL, ALERT, REJECT, ACK and ERROR transmission blocks are non-data blocks, and have the following format.

SYNC	SOH	CA	TA	MTI	ETX	LPC

12.5.1.10 E-Codes

Device selection is performed by E-codes which are appended to the output by the TIP. Similarly, E-codes coming from the terminal indicate the responding device and also report status. Received E-codes are stripped from the input data by the TIP. The codes below are in hexadecimal notation, exclusive of parity.

E-CODE	X'CODE	WRITE (Output)	READ (Input)
E1	42	to CRT (text)	from CRT (text)
E2	20	to PRINTER (text)	from PRINTER (no text): indicates possible error in printing last block. from CARD READER (text): indicates that card reading has stopped.
E3	21	to CARD READER (no text): enables transfer of card buffer to CRT buffer	from PRINTER (no text): indicates that last block correctly printed. from CARD READER (text): normal card data.
E4	22	to CRT (text): position to START INDEX	Not possible.

12.5.1.11 Features Not Supported

The following features of Mode 4 devices are not supported by the TIP:

- Status Request
- Alert
- Diagnostic Write
- Receipt of Initialization Request

12.5.1.12 Error Correction and Load Regulation

The TIP performs short-term recovery for both input and output. The TIP retains three error counters, as follows:

Error Counter	Type of Error
1	No response: after transmitting to the terminal, a response timeout occurs - SOH is never received.
2	Bad response: CA or TA does not correspond to terminal addressed by transmit block Invalid MTI Invalid or missing E-code ETX missing (over length block or premature drop of Data Carrier Detect) Character parity error or LPC error Test in block which should not have text
3	ERROR response (indicates transmit error)

Whenever any error occurs, the TIP increments the appropriate counter, and retries the output-input sequence. If any counter reaches a build-time-parameter-defined threshold in an attempt to complete a single transaction with the terminal, the TIP performs the long term error recovery procedures.

If the TIP is unable to acquire sufficient buffers for an input block or when the host is down, it discards the partial block and repolls the terminal later when the condition is cleared. No error counter is incremented by this operation. However, a counter is incremented in the NPU statistics block to indicate the number of times that regulation has taken place.

12.5.1.13 Long Term Error Recovery

Should the TIP discover an error in communicating with a terminal, the host is sent a STP. In the case of a Mode 4A terminal, a STP is sent on all connections on the cluster. The terminal is then polled every 10 seconds until the problem is resolved. If a read response is ever detected on the terminal, the host is sent a STRT. A terminal status service message is sent each time a change in terminal status is noted.

When all terminals on a switched line are in the long term error recovery mode, the host is sent a line inoperative service message. The host should disconnect this line.

12.5.1.14 Handling of Errors for CDC 711 Terminal

The toggle bit received from the 711 terminal is always the same as appeared in the previous WRITE or POLL. This makes it impossible to determine whether data was correctly received by the 711 if the ACK or REJECT is garbled by transmission line noise, Therefore, the toggle bit of a POLL (which is ignored by all other Mode 4 terminals) is set to the value opposite to that which the terminal is expected to have, assuming that the last WRITE was correctly received by the terminal. Thus, when polling a 711 for toggle, the TIP will receive a bad toggle (not the expected toggle state) and will therefore repeat the WRITE in question. This makes duplication of output on a 711 inevitable - there is no way the TIP can compensate for the loss of status information. However, no output data is lost.

12.5.1.15 Duplication of WRITE Data on CRT

Those terminals which do not have separate CRT and transmission buffers (such as the 200 UT) write output data directly to the CRT screen as it is being received. If the terminal detects an error in the block, it will send an ERROR response, causing the TIP to resend the output. But because the cursor is not is the same place as it was when the original WRITE was performed, the output block will appear two (or more) times on the CRT screen. This is not a problem with RESET WRITE or CLEAR WRITE which home the cursor before displaying the output data, and thus overwrite the bad block.

12.5.2 IVT Commands

The following list of commands will be actioned by the IVT interface for Mode 4:

Command	Definition
TC	Terminal Class
PW	Page Width
PL	Page Length
CN	Cancel Character
CT	Control Character
IN	Input Device for Transparent Mode (Mode 4C only)
PG	Page Wait
B1	User Break One
B2	User Break Two
MS	Operator Message

Each command, when entered from the terminal, must be preceded by the control character and followed by a carriage return or an end of message. The default values for the commands are given in Table A-10. Commands sent by the host are contained within CMD blocks and are not preceded by the control character. Furthermore, only one IVT command may be sent in a CMD block.

If an error is detected in a command from the host, a BR is sent in reply. When errors are detected in a request from the terminal, the message ERR.. is sent.

12.5.3 Batch Virtual Terminal Transforms

The Mode 4 TIP will convert downline data from BVT specifications to the Mode 4 protocol. This conversion is limited to the actual features of the 200UT printer as described in Table 12-13.

Any BVT code pair beginning with X'FF will be considered an error if not supported by the Mode 4 Transform. Any sequence of characters not preceded by a legal BVT code pair will also be considered a host error. All such errors are reported by sending a BRK to the host.

Upline data is translated from the Mode 4 protocol to the BVT specifications as described in Table 12-14. Each card other than EOR, EOI is scanned for spaces and trailing spaces will be removed. Each card is terminated with the end of media indicator. Sequences of uncompressed data will be preceded by the string indicator.

Special processing will occur for EOI, EOR and JOB cards (first card following an EOI card) to transform them to the form specified in Table 8-4. The TIP will always report a data mode of "other" for the JOB and EOR cards. For the

12.5.3 Batch Virtual Terminal Transforms (cont.)

EOI card, the data mode will always be "default."

The card data will be transmitted as read from the terminal. For External BCD the data is converted to ASCII as specified by the system code conversion table.

Table 12-13 Downline BVT Transforms

BVT Interface		Conversion	
		ASCII	BCD
<mode change>	X'FF00 to x'FF09	nul	nul
<forms control>	X'FFE0	X'20	X'50
	X'FFE1	X'4A	X'4A
	X'FFE2	X'4A1B4020	X'4A3E5050
	X'FFE3	X'50	X'30
	X'FFE4	X'41	X'41
	X'FFE5 - X'FFFE	X'20	X'50
<compressed zeros>	X'FF32	X'3030	X'4A4A
	X'FF33	X'1B43	X'3E43
	X'FF34	X'1B44	X'3E44
	⋮	⋮	⋮
	X'FF3F	X'FF4F	X'3E4F
<compressed blanks>	X'FF12	X'2020	X'5050
	X'FF13	X'1B23	X'3E23
	X'FF14	X'1B24	X'3E24
	⋮	⋮	⋮
	X'FF2F	X'1B3F	X'3E3F
<end of media>	X'FF0A	X'1B40	X'3E50

Table 12-14 Upline BVT Transforms

Mode 4 Interface	BVT Interface	
following each JOB or EOR card	<mode change>	X'FF09
following each EOI card	<mode change>	X'FF00
beginning of uncompressed data	<string indicator>	X'FF90
end of card	<end of media>	X'FF0A
(1) esc X'57 in column 1 (1)	<end of record>	X'FF0B
(1) esc X'56 in column 1 (1)	<end of information>	X'FF0C

(1) the code (esc) indicates escape; X'1B for ASCII and X'3E for BCD.

12.6 Auto-Recognition

The host may request auto-recognition for Mode 4 lines. This will invoke a procedure for determining the cluster address and terminals that exist on the line. When the host configures the line the TIP will respond with the line enable response. If the line is dedicated auto-recognition begins; for a switched line the TIP waits until the ring indicator is present.

Auto-recognition begins with a cluster poll to determine the cluster address of the caller. The first four polls are done at cluster address X'70 to allow the caller to hear the audible tone and to allow the modem time to stabilize after the modem data switch is depressed. All cluster address are attempted at least twice before a failure is declared. The time out for a nonexistent cluster is 1/2 to 1 second.

Once the cluster address has been determined, the TIP checks for receipt of a read message. The read message contains an escape code which determines the code set in use by the terminal. Polling continues until the read message is received. If the terminal is BCD then auto-recognition is complete. For an ASCII terminal, the configuration poll is sent to determine the configuration. If there is an error response or no response, the terminal is assumed to be Mode 4A. If a read response is detected the terminal is assumed to be Mode 4C.

The line status (operational) service message is sent to the host for normal completion of auto-recognition. This service message contains the following:

Field Name	Description	
TT	Terminal type (See Table A-2)	
CA	Cluster address	
TA	Terminal address	} For each terminal
DT	Device type	} For each terminal

(See Appendix A for more details)

For all terminals the appropriate terminal type (TT) will be reported indicating one of the following: Mode 4A BCD, Mode 4A ASCII or Mode 4C. The actual cluster address (CA) is also reported in the range 70-7F.

For the Mode 4A BCD or Mode 4A ASCII three terminals are reported describing the console, the card reader and line printer. The terminal address for all three terminals will be X'60.

12.6 Auto-Recognition (cont.)

The configuration request will be used for the Mode 4C terminals to determine the actual terminal address (TA) and actual device types (DT). Either a console or a line printer may be reported.

To complete auto-recognition during the dial up procedures the remote operator must press the send key on at least one of the displays in the cluster. This allows the code set of the terminal to be recognized.

APPENDIX A

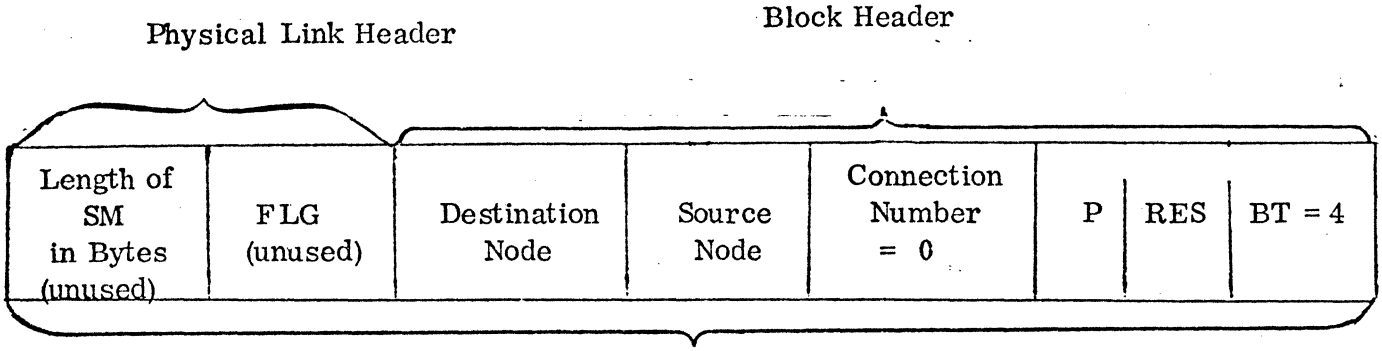
SERVICE MESSAGES

&

TABLE FORMATS

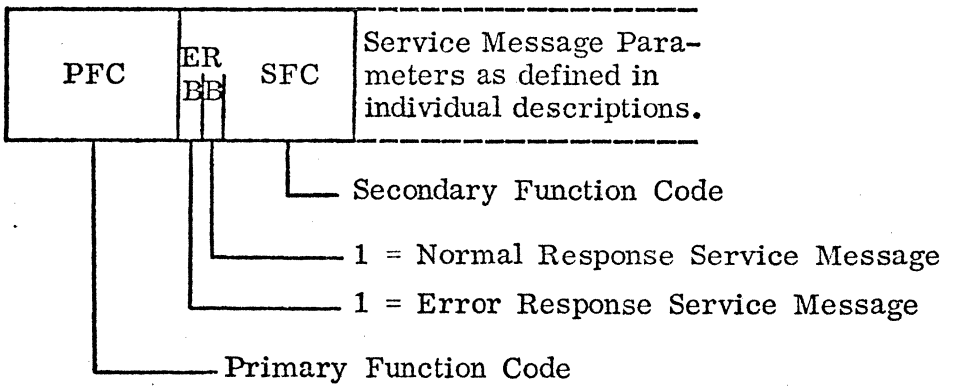
A.0 Service Message General Format

All service messages described within this Appendix are prefixed by the header information shown below. (This information is omitted in the individual descriptions to conserve space.) Each of the major subdivisions in the header diagram is one eight bit byte in length.



P = Priority flag
RES = Unused

The general format of the service message body is shown below. Each of the major subdivisions in the body is also one eight bit byte in length.



X'00 - X'3F = Reserved for Network Use
X'40 - X'7F = Reserved for Intra-Host Use

Service Message Parameters

The following table defines abbreviations used in the individual service message descriptions.

Abbreviation	Meaning
BN	Block Number - used in the overlay load SM to insure delivery of all overlay program blocks.
BSN	Block Serial Number - part of the block protocol. See Section 2.1.1.
BT	Block Type - SM's are always of type CMD. See also Section 2.1.3.
CA	Cluster Address - part of a terminal's physical identification (see Section 4.3.1).
CFS	Configuration State - state of the line as known by the service module (see Table A-4 for values).
CN	Connection Number - part of the block address. In the address of a SM, the CN is always zero. When used as data in a SM, the CN may be nonzero.
DN	Destination Node ID - part of the block address. (See Section 2.1.2.1.)
DT	Device Type - part of the Terminal Type (see Table A-2).
EB	Error Bit in SM response.
FN	Field Number - used in line and terminal configure SM's to describe a field in the LCB or TCB (see Tables A-7 and A-8 for values).
FV	Field Value - used in line and terminal configure SM's as the value to be put in the field. (See Tables A-7 and A-8).
HO	Host Ordinal - a value (0-15) that is included in all SM's that refer to control structures.
ID1	Node ID1 - used to identify the destination node in SM's dealing with logical links.
ID2	Node ID2 - used to identify the source node in SM's dealing with logical links.
LBN	Last Block Number - used in the overlay load SM to insure delivery of all overlay program blocks.

LRN	Link Remote Node - node ID of the neighbor node at the other end of a trunk.
LT	Line Type - used to describe the transmission capabilities of the line (see Table A-3).
NBL	Network Block Limit - the number of blocks allowed to be outstanding for any one terminal at a given time.
NL	Number of Lines - the number of configured lines belonging to a particular CS.
NT	Number of Terminals - the number of terminals configured on a line.
P	Port - the CLA addressed used for a communications line.
PFC	Primary Function Code - used to delineate the class of SM (see Table A-1).
RB	Response Bit in SM response.
RC	Response Code - used in SM responses to indicate the requested action has taken place or an error has occurred.
SFC	Secondary Function Code - used to indicate a particular SM within a class of SM's (see Table A-1).
SN	Source Node - part of the block address (see Section 2.1.2.1).
SP	Subport - used in general to further describe the communications line, but in this release it must be zero.
TA	Terminal Address - part of the terminal's physical identification (see Section 4.3.1).
TOT	Total Number of Status SM's to be sent for this request. Used by the requestor to verify all responses have arrived.
TC	Terminal Class - used to describe the common characteristics of a set of terminals (see Tables A-2 and A-10).
TT	Terminal Type - the combination of DT and TC.

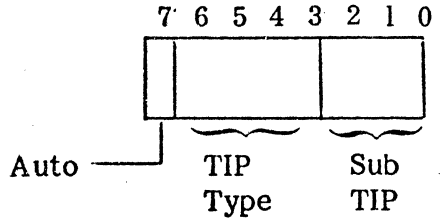
Table A-1

Service Message Summary

Service Message Name	PFC	Mnemonic	SFC	Mnemonic
Load Request	1	D8LOAD	0	D9RQ
Force Load			1	D9FRC
NPU Initialized			2	D9INIT
Configure Logical Link	2	D8LINK	0	D9LLCNF
Delete Logical Link			1	D9LLDLT
Configure Trunk/Line	3	D8CONFIG	0	D9LNCNF
Delete Line			1	D9LNDLT
Configure Terminal			2	D9TMLCNF
Reconfigure Terminal			3	D9TMLRCNF
Delete Terminal			4	D9TMLDLT
Overlay Program Block	4	D8OVLOAD	0	D9OVLBLK
Terminate Overlay			1	D9OVLTMT
Overlay Data	5	D8OVLDATA	0	D9DATA
Logical Link Status Request	6	D8STATUS	0	D9LLSTAT
Trunk Status Request			1	D9TNKSTAT
Line Status Request			2	D9LNSTAT
Terminal Status Request			3	D9TMLSTAT
Line Count Request			5	
NPU Statistics	7	D8COUNTS	0	D9NPUCNTS
Trunk/Line Statistics			1	D9CNTLN
Terminal Statistics			2	D9CNTML
Enable Trunk/Line	8	D8LINE	0	D9ENABLE
Disable Trunk/Line			1	D9DISABLE
Disconnect Trunk/Line			2	D9DISCONNECT
CE Error	X'A	DSEVENT	0	D9CE
Message to Network Operator			9	D9ALARM
Host Broadcast One	X'C	D8USER	0	D9BRDI
Host Broadcast All			1	D9BRDCST
Operator Message			2	D9OPMSG
Terminal Class			3	D9TMCL
Page Width			4	D9PGWD
Page Length			5	D9PGLN

Table A-2. Terminal Type (TT)/Device Type (DT)

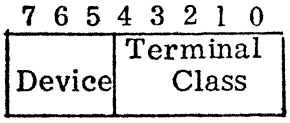
Terminal Type (TT)



TIP Type =	1 ASYNC	2 Mode 4	3 Hasp
Sub TIP			
= 1	ASCII - 110	M4A/BCD	
= 2	ASCII - 150	M4A/ASCII	
= 3	ASCII- 300	M4C	
= 4	2741 EBCD		
= 5	2741 Corres.		

Auto = Auto Recognition Flag

Device Type (DT)



Class	Terminals Supported (By Device)				
	0 Console	1 Card Reader	2 Line Printer	3 Card Punch	4 Plotter
1	M33, etc.				
2	713				
3	M1240				
4	2741				
5	M40				
6	H2000				
7	751-1				
8	T4014				
9	HASP	HASP	HASP	HASP	HASP
10	200 UT	200UT	200UT		

* When the DT byte is sent in a downline SM to identify a particular TCB, the TC field need not match the field in the TCB as the latter can change at any time.

Table A-2. Terminal Type (TT)/Device Type (DT) (cont'd)

Class	Terminals Supported (By Device)				
	0 Console	1 Card Reader	2 Line Printer	3 Card Punch	4 Plotter
11	214				
12	711-10				
13	714				
14	731				
15	734				

Table A-3

Line Types (LT)

Line Type (Value)	Transmission Facility	CLA Type	Modem Type	Answer Mode	Carrier Type	Circuit Type	Turn-Around Required	Turn-Around Delayed	Transmission Mode
(1)	HDX	2560-1	RS232-201A/208B Compatible	Switched	Controlled	2 Wire	YES	NO	Synchronous
(2)	FDX*	2560-1	RS232-201B/208A Compatible	Dedicated	Controlled	4 Wire	YES	NO	Synchronous
(3)	FDX	2560-1	RS232-201B/208A Compatible	Dedicated	Constant	4 Wire	NO	NO	Synchronous
(4)	RESERVED →								
(5)	RESERVED →								
(6)	FDX	2561-1	RS232-103E/113 Compatible	Switched	Constant	2 Wire	NO	NO	Asynchronous
(7)	FDX	2561-1	RS232-103E Compatible	Dedicated	Constant	2 Wire	NO	NO	Asynchronous
(8)	RESERVED →								
(9)	RESERVED →								
(X'A)	FDX	2563-1	RS232-201B Compatible	Dedicated	Constant	4 Wire	NO	NO	HDLC
(X'B)	RESERVED →								

*Operating with HDX Protocol.

Table A-4

Configuration States

Value	Significance
0	LCB Not Configured
1	LCB Configured Not Enabled
2	Enable Requested to TIP
3	Line Operational, No TCB's
4	Line Operational, TCB's Configured
5	Disable Requested to TIP
6	Line Inoperative, No TCB's
7	Line Inoperative, TCB's Configured
8	Disconnect Requested to TIP

INTENTIONALLY BLANK

Table A-7
Line Control Block
Field Number/Field Value (FN/FB) Assignments

Field Number	NPU Mnemonic Name	Description	Mode 4 TIP	ASYNC	HASP
5	BZOWNER	Node ID of Owning CS/NS	1-255*	1-255*	1-255*
21	BZLNSPD	Line Speed Index	-	0-8**	-

* Required for Configuration
** Required if Auto Recognition not specified

Line Speed Index Table

Index	Baud Rate
0	110
1	134.5
2	150
3	300
4	600
5	1200
6	2400
7	4800
8	9600

This field only required if Auto Recognition not specified.

Table A-8

Terminal Control Block

Field Number/Field Value (FN/FV) Assignments

Field Number	NPU Mnemonic Name	Description	VALUES		
			MODE 4 TIP	ASYNC TIP	HASP
5	BSTTYP	Terminal Class	10-13	1-9	14
12	BSOWNER	Node ID of Owning CS	1-255*	1-255*	1-255*
13	BSCN	Connection Number	1-255	1-255	1-255
14	——	Destination Node	0-255	0-255	0-255
15		Source Node	0-255	0-255	0-255
16	BSNBL	Network Block Limit	0-7 *	0-7 *	0-7 *
19	BSIPRI	Input Priority	1-3	1-3	1-3
28	BSPGWIDTH	Page Width	0-255	0-255	0-255
29	BSPGLENGTH	Page Length	0-255	0-255	

* Required for Configuration

Table A-8 (cont.)

Field Number	NPU Mnemonic Name	Description	Values		
			ASYNC TIP	Mode 4 TIP	HASP TIP
30	BSCANCHAR	Cancel Character	0-127	0-127	0-127
31	BSBSCHAR	Backspace Character	0-127		
32	BSCNTRLCHAR	Control Character	0-127	0-127	0-127
33	BSCRIDLES	Carriage Return Idle Count	0-99	-	-
34	BSLFIDLES	Line Feed Idle Count	0-99	-	-
35	BSCRCALC	Calculate CR Idle Count	0-1 (No-Yes)	-	-
36	BSLFCALC	Calculate LF Idle Count	0-1 (No-Yes)		
37	BS APL	APL Code Set	0-2 (No-Yes-Special)		
38	BSXPARENT	Transparent Input Mode	0-1 (No-Yes)	0-1 (No-Yes)	-
39	BSXCHM	Transparent Char. Count Delimiter (Most Sign. Bits)	0-15 (Most Signif. 4 bits)	-	-
40	BSXCHL	Transparent Char. Count Delimiter (least Sign. Bits)	0-255 (least signif. 8 bits)	-	-
41	BSXCHAR	Transparent Char. Delimiter	0-255	-	-
42	BSXTO	Transparent Time Out Delimiter	0-1 (No-Yes)	-	-
43	BSINDEV	Input Device	0-1 (KB, PT)	-	-
44	BSOUTDEV	Output Device	0-2 (PR, DI, PT)	-	-
45	BSECHOPLX	Echoplex Mode	0-1 (No, Yes)		
46	BSPGWAIT	Page Wait	0-1 (No, Yes)	0-1 (No, Yes)	-
47	BSPARITY	Parity Mode	0-3 (Zero, Odd-Even, None)		-
48	BSABTLINE	Abort Output Line Char.	0-127	-	-
49	BSUSR1	User Break 1 Char.	0-127	0-127	0-127
50	BSUSR2	User Break 2 Char.	0-127	0-127	0-127
51	BSCODE	TIP Code Set	4-5**	1-3**	

* Required for Configuration

** Required only if not Auto Recognition

1 = Mode 4 A BCD
2 = Mode 4 A ASCII
3 = Mode 4 C
4 = 2741 EBCD
5 = 2741 Correspondence
(Note: same numerical values as for SUB TIP)

INTENTIONALLY LEFT BLANK

TABLE A-10

Default Parameters for Terminal Classes

Terminal Class	1	2	3	4	5	6	7	8
Terminal Supported	M33, M35 M37, M38	CDC 713-10	Mem 1240	IBM 2741	M40	Hazel-tine 2000	CDC 751-1	Tek-tronix
Page Width	72	72	72	130	74	74	70	72
Page Length	0	0	0	0	0	0	0	0
Parity	Even	Even	Even	Odd	Even	Even	Even	Even
Cancel Input Line Char.	$	$	$	$	$	$	$	$
Back Space	BS	BS	BS	BS	N/A	BS	BS	BS
Control Char.	%	%	%	%	%	%	%	%
Carriage Return Idle Cnt.	2	0	CA*	CA*	1	0	0	0
Line Feed Idle Count	1	0	CA*	1	3	3	0	0
APL Mode	N/A	N/A	No	No	N/A	N/A	N/A	N/A
Transparent Mode	No	No	No	No	No	No	No	No
Transparent Delimiter	CR/ 2043	CR/ 2043	CR/ 2043	CR/ 2043	CR/ 2043	CR/ 2043	CR/ 2043	CR/ 2043
Device Mode In/Out	KBD/ PTR	KBD/ CRT	KBD/ CRT	KBD/ PRT	KBD/ CRT	KBD/ CRT	KBD/ CRT	KBD/ CRT
Echo Mode	No	No	No	N/A	No	No	No	No
Page Wait	No	No	No	No	No	No	No	No
Abort Output Line	CAN	CAN	CAN	?	CAN	CAN	CAN	$
User Break 1	:	:	:	:	:	:	:	:
User Break 2	)	)	)	)	)	)	)	)

* Calculated by TIP

Table A-10 (cont.)

Terminal Class	9	10	11	12	13	14	15
Terminals Supported	HASP	200UT	214	711-10	714	731	734
Page Width	80	80	80	80	80	80	80
Page Length	N/A	13	13	16	16	13	13
Cancel Input Line Char.	$	$	$	$	$	$	$
Control Char.	%	%	%	%	%	%	%
Transparent Mode	N/A	N/A	N/A	No	No	N/A	N/A
Device Mode In/Out	N/A	KBD/ CRT	KBD/ CRT	KBD/ CRT	KBD/ CRT	KBD/ CRT	KBD/ CRT
Page Wait	N/A	Yes	Yes	Yes	Yes	Yes	Yes
User Break 1	:	:	:	:	:	:	:
User Break 2	)	)	)	)	)	)	)

Table A-11

Commands (CMD Blocks) Sent Over Logical Connections (CN+0)

Start Input

PFC =	SFC =
C1	05

Stop Input

PFC =	SFC =
C1	06

Input Stopped

PFC =	SFC =	RC
C1	07	

0 = Stop Input response
1 = Input device not ready
2 = Card slip error
3 = EOI input

Define Terminal Characteristic (Terminal Parameter)

PFC =	SFC =	String
C1	04	

See Table 9-1

Individual Service Messages

A.1.0 Load Request

PFC = 1	SFC = 0	LRN	P	SP

Line over which load will be performed

Node ID of element to load

Response

NONE

A.1.1 Force Load

PFC = 1	SFC = 1

Response

NONE

A.1.2 NPU Initialized

PFC = 1	SFC = 2	CCP Version	CCP Cycle	CCP Level

Describes the current software running in the NPU

Response

NONE

A.2.0 Configure Logical Link

PFC = 2	SFC = 0	ID1	ID2	HQ

Nodes forming Logical Link; ID1 to be used as the Local ID at the NPU

Normal Response

PFC = 2	SFC = X'40	ID1	ID2	HO	RC

RC: 0 = Configured

Error Response

PFC = 2	SFC = X'80	ID1	ID2	HO	RC

RC: 1 = ID1 invalid
2 = ID2 invalid

A.2.1 Delete Logical Link

PFC = 2	SFC = 1	ID1	ID2	HO

ID1, ID2: Nodes forming Logical Link; ID1 to be used as the Local ID at the NPU

Normal Response

PFC = 2	SFC = X'41	ID1	ID2	HO

Error Response

PFC = 2	SFC = X'81	ID1	ID2	HO

SFC: Logical Link does not exist

A.3.0 Configure Line

PFC = 3	SFC = 0	P	SP	HO	LT	TT	FN1	FV1	•••	FN_n	FV_n

(See Table A-7)

(See Table A-2)

(See Table A-3)

Normal Response

The normal response is a Line Enabled Normal Response SM. See Section A.8.0.

Error Response

PFC = 3	SFC = X'80	P	SP	HO	LT	TT	RC	FN	FV

(FN/FV Pair returned if RC = 1)

(See Table A-3)

(See Table A-2)

1 = Invalid FN/FV
2 = Invalid Line Number
3 = Line Control Block already Configured (Configure)
4 = Invalid Line Type
5 = Invalid Terminal Type
6 = Diagnostics In Progress

A.3.1 Delete Line

PFC = 3	SFC = 1	P	SP	HO

Normal Response

PFC = 3	SFC = X'41	P	SP	HO	RC = 0

Error Response

PFC = 3	SFC = X'81	P	SP	HO	RC

RC = 2 Invalid Line Number
= 3 Line Not Configured

A.3.2
A.3.3 Configure/Reconfigure Terminal

PFC = 3	SFC = 2	P	SP	HO	CA	TA	DT	HO	FN1	FV1	•••	FN_n	FV_n

2 = Configure
3 = Reconfigure

(See Table A-2)

(See Table A-8)

Normal Response

PFC = 3	SFC	P	SP	HO	CA	TA	DT	HO	RC=0

X'42 = Terminal Configured
X'43 = Terminal Reconfigured

(See Table A-2)

Error Response

PFC = 3	SFC	P	SP	HO	CA	TA	DT	HO	RC	FN	FV

(FN/FV Pair returned if RC = 1 or 9)

X'82 = Configure
X'83 = Reconfigure

(See Table A-2)

1 = Invalid FN or FV
2 = Invalid Line Number or Terminal Address
3 = Terminal already configured (Configure) or not configured (Reconfigure)
4 = No Buffer for TCB (Temporary)
5 = Invalid DT
6 = Line Inoperative or not Enabled
8 = Logical Link not established
9 = CN in Use

A.3.4 Delete Terminal

PFC = 3	SFC = 4	P	SP	HO	CA	TA	DT	HO

Normal Response

PFC = 3	SFC = X'44	P	SP	HO	CA	TA	DT	HO	RC=0

Error Response

PFC = 3	SFC = X'84	P	SP	HO	CA	TA	DT	HO	RC

2 = Invalid Line Number
3 = Terminal not Configured

A.4.0 Overlay Program Block

PFC = 4	SFC = 0	BN	LBN	Overlay ID	Checksum		• • •	

Words 1-n of Overlay

Complement of Arithmetic Sum of Data Words

Normal Response

PFC = 4	SFC = X'40	BN	LBN	Overlay ID	RC = 0

Error Response

PFC = 4	SFC = X'80	BN	LBN	Overlay ID	RC

1 = Overlay Space in Use
2 = Checksum Error

A.4.1 Terminate Overlay

PFC = 4	SFC = 1

Response

PFC = 4	SFC = X'41

A.5.0 Overlay Data (General Form)

PFC = 5	SFC = 0	Overlay ID	DATA

Normal Response

PFC = 5	SFC = X'40	Overlay ID	DATA

Error Response

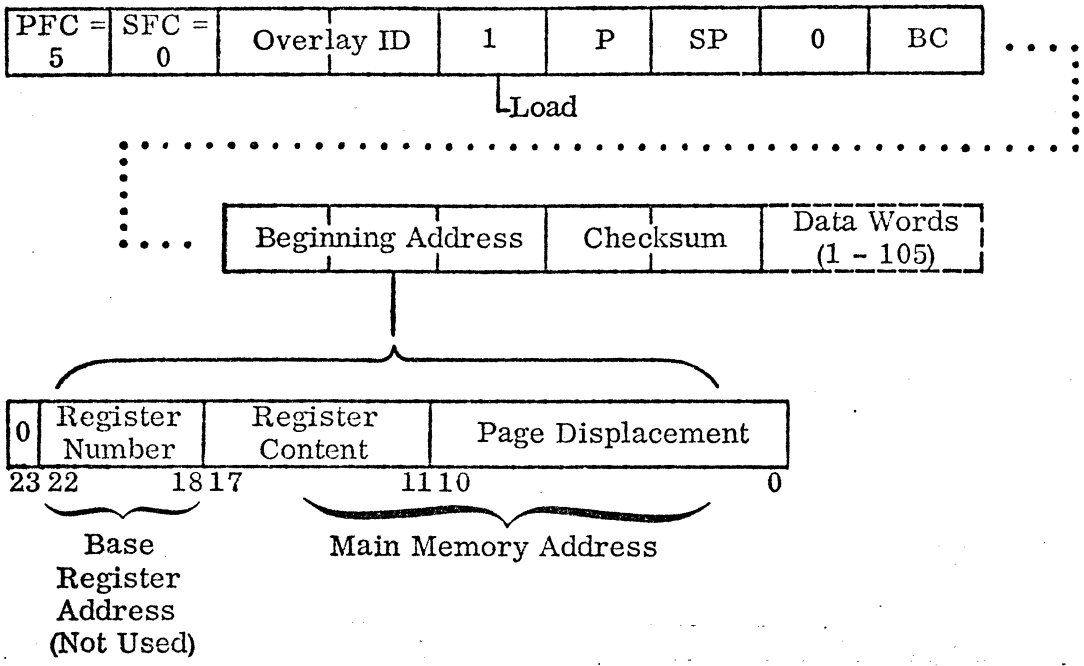
PFC = 5	SFC = X'80	Overlay ID	RC	Overlay ID

(Overlay ID returned if RC = 1)

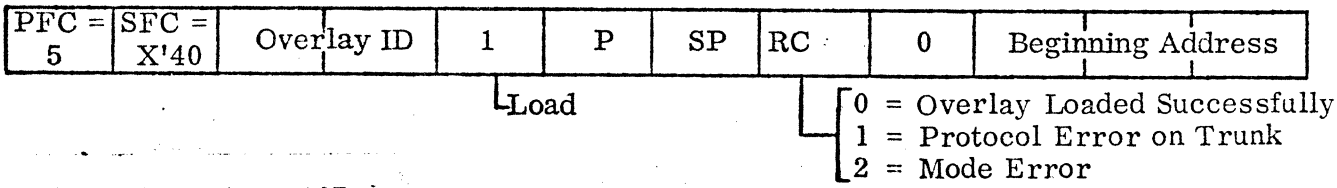
1 = Invalid OVID
2 = No Overlay Loaded

A.5.A Overlay Data (Loading/Dumping)

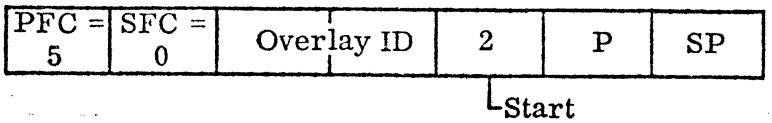
Load Command



Response



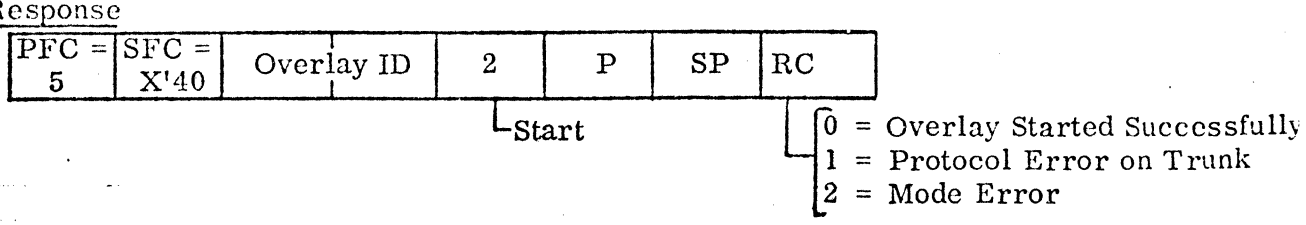
Start Command



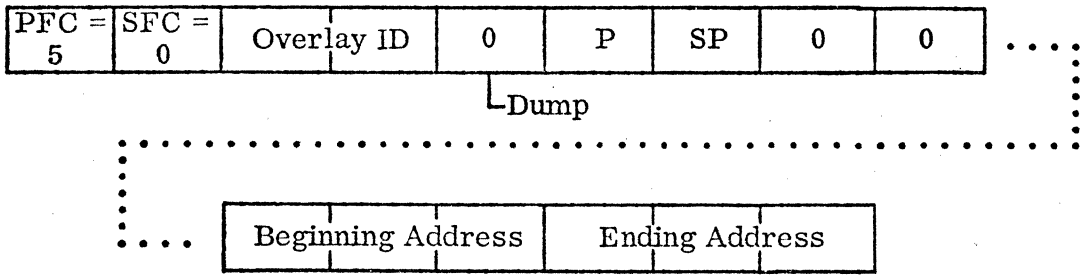
A.5.A Overlay Data (Loading/Dumping) (continued)

Start Command (continued)

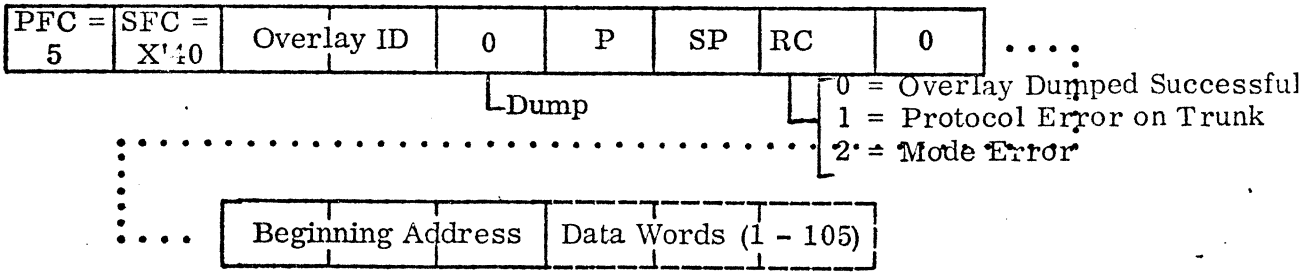
Response



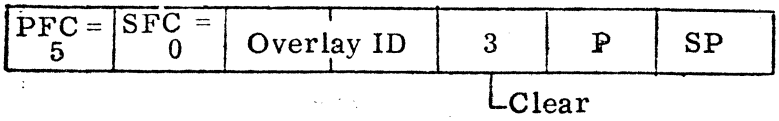
Dump Command



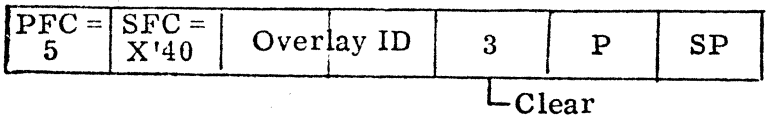
Response



Clear Command



Response



INTENTIONALLY BLANK

A.6.0 Logical Link Status Request

PFC = 6	SFC = 0	ID1	ID2	HO

Node ID's forming Logical Link; ID1 is Node ID of the NPU. If ID1 and ID2 are missing, return status for all Logical Links supported by NPU.

Normal Response

PFC = 6	SFC = X'40	ID1	ID2	HO	RC	RL	INIT	TOT

0 = Logical Link Operational
1 = Logical Link Inoperative
TOT = number of LL in an "all" request
RL = Regulation Level
INIT = 0 - Second and subsequent responses
1 - First unsolicited response.

Error Response

PFC = 6	SFC = X'80	ID1	ID2	HO	RC

2 = Logical Link(s) not Configured
3 = Logical Link Status Request in progress or request not from NS

Note: The normal response may be solicited (SFC = X'40) or unsolicited (SFC = X'0).

A.6.1 Trunk Status Request

PFC = 6	SFC = 1	P	SP	HO = 0

If missing, return status on all trunks

Normal Response

PFC = 6	SFC = X'41	P	SP	HO = 0	RC	LT	CFS	LRN	TOT

(See Table A-3) (See Table A-4)

RC = 0 - Trunk Operational
4 - Trunk Inoperative
5 - No Ring Indicator

Error Response

PFC = 6	SFC = X'81	P	SP	HO = 0	RC	LT	CFS	LRN	TOT

(See Table A-4)

(See Table A-3)

1 = Invalid Line Number or No Trunks Configured Belonging to Requestor
2 = Trunk Status Request in Progress
3 = Illegal Configuration State
4 = Cannot Disable Last Path to NS

Unsolicited Response

NOTE: Normal Responses above may be sent as an unsolicited status message with SFC = 1.

A.6.2 Line Status Request

PFC = 6	SFC = 2	P	SP	HO

If missing, return status on all lines except trunks

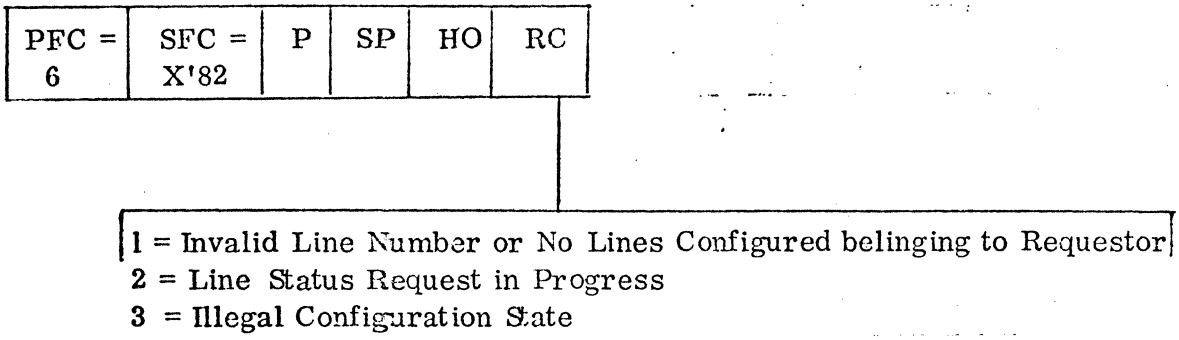
Normal Response

PFC = 6	SFC = X'42	P	SP	HO	RC	LT	CFS	NT

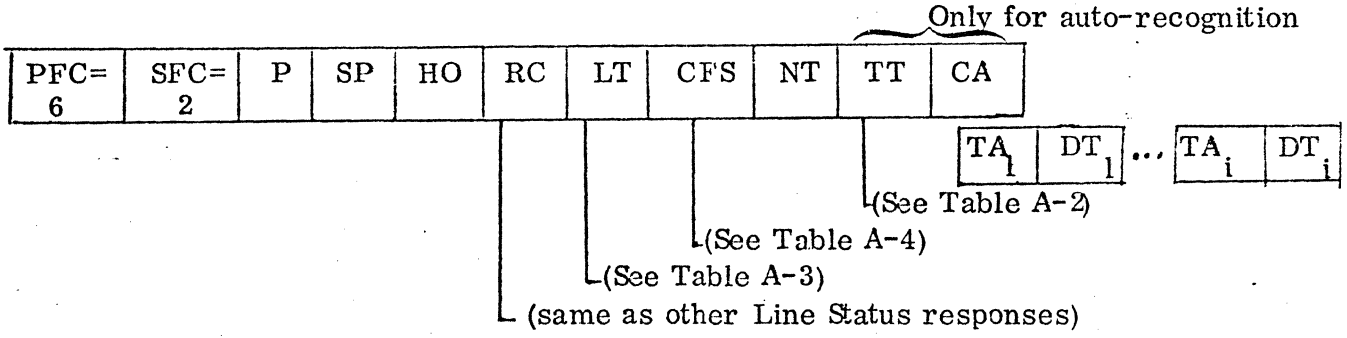
(See Table A-3) (See Table A-4)

RC = 0 - Line Operational
4 - Line Inoperative
5 - No Ring Indicator or Auto Recognition in Progress

Error Response



Unsolicited Response



For Auto Recognition responses, the TA DT pairs are repeated for each terminal that can be detected by the TIP. The ASYNC TIP will only report one TA DT pair. The DT may be either zero to indicate no information or four to indicate the IBM 2741. The Mode 4 TIP may report up to 15 TA DT pairs with the full range of values as shown in Table A-2 for DT. For a Mode 4A cluster, the TIP will report 3 terminals: DT=0, TC=0; DT=1, TC=0; DT=2; TC=0. Mode 4C consoles will be reported as TC=0 as it is not possible to distinguish 711 from 714.

A.6.3 Terminal Status Request

PFC = 6	SFC = 3	P	SP	HO

Normal Response

PFC = 6	SFC = X'43	P	SP	HO	CA	TA	DT	HO	RC	DN	SN	CN	TOT

(See Table A-2)

RC = 0 - Terminal Operational
4 - Terminal Inoperative

Error Response

PFC = 6	SFC = X'83	P	SP	HO	CA	TA	DT	HO	RC

(See Table A-2)

RC
1 = Invalid Line Number or No Terminals Configured belonging to Requestor
2 = No Terminals Configured
3 = Terminal not Configured
5 = Terminal Status Request in Progress

Unsolicited Response

NOTE: Normal Response (above) may be sent as an unsolicited status message with SFC = 3.

A.6.4 Line Count Request

PFC = 6	SFC = 5

Normal Response

PFC = 6	SFC = X'45	NL

A.7.0 NPU Statistics

PFC = 7	SFC = 0	Statistics Words

Word 1 = Service Messages Generated
Word 2 = Service Messages Processed
Word 3 = Bad Service Messages Received
Word 4 = Blocks Discarded due to Bad Address
Word 5 = Packets/Blocks Discarded due to Bad Format

Word 6 = Times at Regulation Level 3
Word 7 = Times at Regulation Level 2
Word 8 = Times at Regulation Level 1
Word 9 = Times at Regulation Level 0

Response

NONE

A.7.1 Trunk/Line Statistics

PFC= 7	SFC= 1	P	SP	HO	LRN	00	Statistics Words 1 - 4

(Trunks only; LRN = 0 for Lines)

Word 1 = Blocks Transmitted
Word 2 = Blocks Received
Word 3 = Characters Transmitted (Good Blocks Only)
Word 4 = Characters Received (Good Blocks Only)

Response

NONE

A.7.2 Terminal Statistics

PFC= 7	SFC= 2	P	SP	HO	CA	TA	DT	HO	Statistics Words 1 - 3

(See Table A-2)

Word 1 = Blocks Transmitted
Word 2 = Blocks Received
Word 3 = Blocks In Error

Response

NONE

A.8.0 Enable Trunk/Line

PFC = 8	SFC = 0	P	SP	HO

Normal Response (Trunk/Line Enabled)

PFC = 8	SFC= X'40	P	SP	HO	RC	LT	CFS	LRN / NT=0

Trunk
Line

(See Table A-4)

(See Table A-3)

RC = 0 - Trunk/Line Enabled and Operational
4 - Trunk/Line Inoperative
5 - Line Enabled - Wait for Ring Indicator or Auto Recognition result.

Error Response (Trunk/Line Not Enabled)

PFC= 8	SFC= X'80	P	SP	HO	RC

(See Trunk/Line Status Request Response Codes)

A.8.1 Disable Trunk/Line

PFC = 8	SFC = 1	P	SP	HO

Normal Response (Trunk/Line Disabled)

PFC= 8	SFC= X'41	P	SP	HO	RC = 0	LT	CFS	LRN / NT	Trunk Line

(See Table A-3)

(See Table A-4)

Error Response

PFC= 8	SFC= X'81	P	SP	HO	RC

(See Trunk/Line Status Request Responses)

A.8.2 Disconnect Trunk/Line

PFC = 8	SFC = 2	P	SP	HO

Normal Response

Normal response is Line Enabled Normal Response SM.

Error Response

PFC = 8	SFC = X'82	P	SP	HO	RC

*SFC = X'80 for RC≥ 4

(See Trunk/Line Status Request Error Response Codes)

A.A.1 Message to Network Operator

PFC = A	SFC = 01	0	Text (0-50 characters)

A.C.0 Host Broadcast One

PFC =X'C	SFC =0	P	SP	HO	CA	TA	DT	HO	TEXT

Text must be 1-50 characters in IVT compatible format.

Normal Response

PFC X'C	SFC X'40	P	SP	HO	CA	TA	DT	HO	RC=0

Error Response

PFC X'C	SFC X'80	P	SP	HO	CA	TA	DT	HO	RC

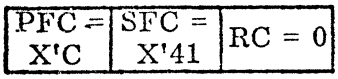
1 = Invalid Line Number
2 = Invalid Device Type
3 = Terminal not Configured
4 = Terminal Inoperative
5 = Host Broadcast in Process

A.C.1 Host Broadcast - All

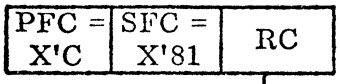
If Zero, Broadcast to Console Terminals supported by NPU

TEXT (50 Characters or Less. in IVT compatible format).

Normal Response



Error Response



1 = No Logical Link Established or Specified Logical Link not Established
2 = Broadcast Already in Progress

A.C.2 Operator Message

PFC= X'C	SFC= 2	P	SP	HO	CA	TA	DT	HO	TEXT (50 characters or less)

A.C.3 Terminal Class

PFC= X'C	SFC= 3	P	SP	HO	CA	TA	DT	HO	ORIG

DT → (See Table A-2)

ORIG { 0 = Terminal User
1 = Application

A.C.4 Page Width

PFC= X'C	SFC= 4	P	SP	HO	CA	TA	DT	HO	ORIG	PW

PW → Page Width in characters per line.

A.C.5 Page Length

PFC= X'C	SFC= 5	P	SP	HO	CA	TA	DT	HO	ORIG	PL

PL → Page Length in lines per page.

Mark S. Lacey

3270 TIPs

3270 TIPs are currently available via the Special Products Library for NOS/BE - INTERCOM 5 - CCI and NOS-NAM R4 - CCP 3.1.

The TIPS are designed for use with the IBM 3270 Information Display System, using IBM's Binary Synchronous Communications (BSC) protocol.

The NOS/BE - INTERCOM 5 TIP is available and is being used at Rockwell International in Seal Beach, California, SUN LIFE in London, and the Danish Post and Telegraph. The INTERCOM 5 TIP has also been sold to SGIO in Australia.

The NOS-NAM variant of the 3270 TIP is also available and is currently being installed at KOCO (Korean Oil Company) and Raytheon in Boston. The NOS-NAM 3270 TIP has also been sold to FERMINAL in Chicago and BPS in France.

Following are the major features of the 3270 TIPs:

- Supported 3270 Control Units and Devices/Stations
  - 3271 Control Unit, Models 1, 2
  - 3274 Control Unit, Model 1C
  - 3275 Display Station, Models 1, 2 (without Dial feature)
  - 3276 Control Unit/Display Station, Models 1, 2, 3, 4
  - 3277 Display Station, Models 1, 2
  - 3278 Display Station, Models 1, 2, 3, 4
  - (NOS/BE only) 3284, 3286, 3288 printers
  - (NOS/NAM only) card reader support (magnetic - id card) is implicitly available in transparent mode

- Configuration

  - The TIPs will support multiple 3270 display devices connected to a 3270 Control Unit and multiple 3270 Control Units on one communication line. The addressing scheme of the 3270 allows up to 32 Control Units on a multi-point line and up to 32 devices per Control Unit. The NOS-NAM TIP will not impose any additional limitation on the configuration. NOS/BE - INTERCOM 5 limits support to 12 Control Units on a line where each may have a maximum of 11 consoles and 8 line printers.

- Line types

  - Basically 3 line types (synchronous) line types are supported by the TIP:

    - Dial-up, with the restriction of the 3270 point-to-point (dial) feature because that is a different protocol (no polling/selecting!).

    - Dedicated, controlled carrier. The TIP will raise and drop the carrier with every transmission (the Control Unit can run with either controlled or constant carrier).

    - Dedicated, constant carrier. The TIP will keep the carrier up (the Control Unit can run with either controlled or constant carrier).

- Line speeds

  - The TIP will not impose a limit on the transmission line speeds, except with respect to timing out the largest possible block sent to the terminal, a lower limit is assumed of:

    - 600 BPS for screensize up to 480 characters
    - 1200 BPS for screensize up to 960 characters
    - 2000 BPS for screensizes over 960 characters

    The upper limit depends on the capacity of the 255x (at least 19.2 KBPS).

- Use of General Polls

  - Input will be solicited by the TIP through issuing General Polls to each of the configured Control Units. Output will be sent to a display device after the TIP has successfully selected the device.

- Code Set, Protocol

  - The TIP will support the EBCDIC versions of the 3270.

- Display Support

  - The TIPs will allow the 3270 terminal user to have CYBER applications interact with the terminal as follows:

    - NOS/BE - INTERCOM 5

      - Interactive (line) mode - 96 char ASCII

    - NOS - NAM

      - Virtual, through use of IVT

      - Transparent. This mode allows the user to make use of the special 3270 Data-Protect features.

The recommended sell price for the 3270 TIP is $12,500 per copy. For additional information, please contact either

Mort Goldstein MNA02E
(612) 830-6901

or

Lynn Peterson STAOPS

(714) 630-2022, x2313

3270 TIP ERS

For NAM/CCP

May, 1979

PUB0131S

TABLE OF CONTENTS

TABLES

1.0 BSC_MULTIPOINT_TIP_FOR_3270

This section describes a TIP for use with the IBM 3270 Information Display System when remotely attached to a 2550. The protocol used is IBM's Binary Synchronous Communications (BSC) operating as multipoint data links.

2.0 FEATURES

- 3271 Control Unit, Models 1, 2
- 3274 Control Unit, Model 1C
- 3275 Display Station, Models 1, 2 (without Dial-feature)
- 3276 Control Unit/Display Station Models 1, 2, 3, 4
- 3277 Display Station, Model 1, 2
- 3278 Display Station, Models 1, 2, 3, 4
- EBCDIC code set
- BSC protocol
- up to 32 controllers (clusters) on a line where each cluster may have a maximum of 32 Display Stations (devices)
- data protect is supported in transparent mode (IVT)
- card reader support (magnetic-id card) is implicitly available in transparent mode
- line printers are not supported
- auto recognition not performed

3.0 LINE_CONTROL

The TIP commences to service a line as soon as the line is enabled.

The TIP reports line operational if the modem and CLA signals are present, otherwise, line inoperative. If during normal operation the line becomes inoperative, the TIP will suspend activity on the line and report line inoperative.

Under normal conditions the TIP will issue one general poll per second to each cluster and then, if needed, issue specific polls to all devices on that cluster.

The input process will terminate when the cluster controller indicates it has no more traffic.

A device will be selected as soon as output is available and the device is ready to receive data.

The invitation to input or output will rotate around all the cluster controllers on the line.

5.0 OPERATIONAL OVERVIEW

The TIP will control activity on a 3270 by polling for input and selecting for delivery of output.

The 3270 station is under control of the TIP in either of two modes:

- control mode
- text mode

In control mode, the station is monitoring the line for a valid poll/select sequence. When detected, the station enters the text mode. In text mode, the station is either the master or slave station with the TIP assuming the opposite role.

When the entry into text mode is the result of a poll, the station is the master, the TIP is the slave. When transition is caused by a select, the roles are reversed.

In text mode, blocks of data are transferred from master to slave, one at a time with a positive acknowledgement being required for each block prior to delivery of the next. The master normally determines the end of transfer and uses the "EOT" sequence to cause return to the control mode.

The 3270 generates status information to assist in the correct functioning of its devices. The status will be processed by the TIP.

6.0 CODE CONVERSION

The code set is EBCDIC. EBCDIC will be translated to ASCII prior to delivery to the host. All downline data received from the host is also in ASCII, the TIP will translate to EBCDIC before delivery.

EBCDIC requires standard BSC CRC.

CRC polynomial is: X16+X15+X2+1

7.0 TRANSMISSION_BLOCKS

7.1 INPUT

PAD	SYN	SYN	STX	CU	DA	TEXT	ETB or ETX	CRC	PAD

CU = cluster unit poll address
DA = device address
PAD = x'55' at start of block and x'7F' at end of block. The leading PAD (x'55') is required to start the clock of some older 3270 Controllers. The TIP does not require to receive the leading PAD, it only looks for 2 SYNC's.

7.2 OUTPUT

PAD	SYN	SYN	SYN	STX	TEXT	ETX	CRC	PAD

- periodic "SYN" insertion will be necessary for output blocks whose transmission time exceeds one second.

8.0 INTERNAL_BLOCKS

8.0.1 General_Poll/Poll_Sequence

PAD	SYN	SYN	SYN	EOT	PAD	SYN	SYN	SYN	CU	CU	GP	GP	ENQ	PAD

- first 5 characters clear any cluster controller from text mode (the EOT-sequence).
- GP = general poll code in place of DA for general polls (x '7F' in EBCDIC).

8.0.1.1 General Poll/Poll Responses

- Device Response

  Any device having an input requirement may be internally selected by the 3270 cluster controller. The response may be an input message, a test request message or device status information.

  The cluster controller will start at random a device and input all device messages, sequentially, as long as "ACK"s are received to blocks until all devices have been serviced. The first 2 characters of the first block of each input message identify the responding device.

  The TIP will send blocks terminated by "ETB"s as "BLK" blocks; terminated by "ETX" as "MSG" blocks to the host.

- No Traffic/End Traffic Response

  The cluster controller may respond with an "EOT". The tip moves on to its next phase of line service.

- Time Out or Invalid Sequence

  N number of retries will be attempted. After N retries the device or cluster will be declared inoperative by the TIP. If the cluster is declared inoperative, then by default, all the devices on the cluster will be declared inoperative.

8.0.2 Select Sequence

POLL	SYN	SYN	SYN	EOT	POLL	SYN	SYN	SYN	CU	CU	DA	DA	ENQ	PAD

8.0.2.1 Response to Selection

- ACKO

  The device has entered the text mode and is ready to receive the message.

- WACK

  The device is busy. Output will not be attempted until the device indicates end of busy.

The device has pending status or is unavailable. The TIP terminates the select sequence and specifically polls the device to obtain status.

- Timeout or Bad Response

  Action is the same as for polls.

9.0 DATA TRANSFER PHASE

9.1 MASTER

The TIP enters the data transfer phase in the master role after a successful select sequence. All messages in the queue are delivered to the terminal. Each text message is prepared for output with the appropriate communications envelope and a redundancy checking CRC. The anticipated response to the first message delivered is ACK1 and for subsequent messages, the acknowledgement alternates between ACK0 and ACK1. When the last message is successfully delivered, the TIP returns the selected terminal to the control mode by use of the "EOT" sequence. The TIP limits the number of transmissions blocks sent to the terminal during one selection sequence to 4.

9.1.1 Response to Output Blocks

- ACK0/ACK1

  Alternating acknowledgements to correctly received messages. Action is to prepare and deliver the next message, if any, or return the device to the control mode.

- WACK

  Acknowledges the output message but indicates the device is now busy. The TIP issues an "EOT" causing the device to return to control mode. Some time later, the TIP will again select the device.

- EOT

  Device is unable to perform the operation requested by the transaction. The TIP issues a specific poll to obtain the device status.

- NAK

  The previous message was received in error. After N "NAK"s, the TIP will terminate the selection with an "EOT" sequence and declare the device inoperative.

- Timeout or Invalid Sequence

  Action is the same as for polls.

9.2 SLAVE

The TIP enters this phase as a result of a successful poll.

The 3270 cluster starts a search at some random device and scans all devices in turn. Each device which has data pending (including status) will input. The TIP will acknowledge each block received, with alternating ACK0/ACK1.

The first block of each message identifies the responding terminal by containing the CU and the DA. These characters immediately follow the STX. Status and test request messages carry a SOH and two id bytes prior to the STX.

9.2.1 Received Sequences

- Data Blocks

  The TIP performs the appropriate redundancy check on each data block. For the first block successfully received after a role reversal or poll, an ACK1 is returned. For each succeeding block, the ACK is alternated. Status messages are identified by the characters "%R", immediately following the "SOH."

- Data Block Ending in ENQ

  The 3270 terminates a data block abnormally with an "ENQ" character upon detection of internal errors.

  The TIP "NAK"s the block. The 3270 is expected to respond with an "EOT". The TIP performs a specific poll to obtain status. The status will be such that the device is declared inoperative.

- EOT

  The 3270 sends an "EOT" sequence to end a normal data transfer sequence. The TIP enters control mode and performs the next task in turn for the line.

- Bad Blocks

  The TIP "NAK"s bad blocks. The 3270 does not count "NAK"s. The TIP will count "NAK"s, send out and allow the 3270 for N retries.

- Timeout

  The TIP must timeout a 3270 when operating in the slave mode during data transfers. If the 3270 does not continue a data transfer sequence, the TIP must regain control of the line in order to continue servicing other 3270 lines. The TIP must commence a timeout after each response to a data block.

  If the 3270 does not continue the transfer within the time, T, the TIP must abort the transfer with an EOT sequence. Action at this point is the same as for poll sequences.

  T is 1 second for the data to get started and 3 seconds during transfer of the data.

10.0 UNUSED_BSC_FEATURES

- TTD
- Forward Abort
- Conversational Mode
- 3275 dial-feature

11.0 EXCEPTION_HANDLING

The 3270 has self diagnostic capabilities. This results in presentation of status messages from cluster controllers with devices experiencing abnormal conditions. Some information in such messages requires action by the TIP to modify the communications activity of the terminal, e.g., device busy, not busy, inoperative, operative, etc.

11.1 STATUS MESSAGE FORMAT

SOH	%	R	STX	CU	DA	SS0	SS1	ETX	CRC

- % after "SOH" signifies status message

- SS0 and SS1 are sense and status bytes defined in Table 1.

The status message is returned for a device under the following conditions:

- general poll
  for all conditions other than device busy or unavailable

- specific poll
  all conditions

If error conditions occur during a data transfer phase, status is returned.

The TIP will be forced into performing a specific poll to obtain status in the following cases:

- "RVI" in response to a selection

- "EOT" in response to output of data

- input block terminated by "ENQ".

11.1.1 Handling of Status Messages

During normal operation of the line, BSC will detect various fault conditions, all of which (if they persist) will result in the cluster controller being declared inoperative.

Detection of failure of individual devices on a cluster is only possible by examination of status messages.

The busy condition is signified by a "WACK" response to a select, followed by a status message indicating busy response to a specific poll or more normally, a "WACK" response to an output message. In both of these conditions, further output must not be attempted until a device end (non-busy) status is received in response to the normal general poll/poll sequences.

Since any declaration of a terminal inoperative requires a corresponding method of detecting a return to the operational state, the actions by the TIP are limited to positive failure cases.

The TIP will, therefore,

- detect terminal I/O malfunctions

- monitor device available/unavailable, ready/not ready, busy/not busy

The TIP will not output to a device which is not ready, not available or busy.

A device will be reported as inoperative if

- it malfunctions
- it goes not ready or not available and this has not yet been reported.

A device will be reported as operational if it goes from not ready or not available to ready and available, or when input is received for an inoperative device.

12.0 INOPERATIVE PROCESSING

A controller or line failure will be detected by one of the following methods

- invalid or no response to N successive polls.
- invalid or no response to N successive selects.
- "NAK" response to N successive block retransmissions.
- Timeout of input data phase.

Each of these conditions will result in the cluster controller, or the device which was currently being actioned, being declared inoperative.

The controller will be set to the slow poll mode. In the slow poll mode, a controller is polled once per 10 seconds. Only one attempt is made to establish contact with the controller during the slow poll mode. If the controller responds with a valid sequence other than EOT, the device that responded is declared operational by the TIP and normal polling of the cluster controller is resumed.

If a device is inoperative the TIP will send one specific poll to that device every 60 seconds such that status that can only be obtained through a specific poll can be received by the TIP.

A controller becomes operational as soon as a device on that controller becomes operational. A device can become operational by correctly sending data or status (good status).

As soon as the first device on a controller becomes operational, the TIP will assume normal general-polling (one per second).

12.1 RETRIES

The TIP will attempt the following retries in case of consecutive errors:

- 15 on NAK's or CRC-errors received.
- 7 on ENQ's, Time-outs or Bad-responses.
- 3 on Text-terminated-by-ENQ or ACK0/ACK1 out-of-sequence.

In case of mix consecutive errors the TIP has a retry limit of 31.

13.0 HOST/TERMINAL-USER INTERFACES

13.1 NORMAL INTERACTIVE MODE

In this mode all the standard IVT features (page-wait, auto-input, line-folding, CR/LF in text, format-effectors) are supported by the TIP. In addition, the TIP will after having sent all the available output to the terminal, send a blank line to the screen. This line will unlock the keyboard and give the terminal-user space to enter a next line of input.

13.1.1 Page-wait

The TIP will declare a terminal in page-wait as soon as a screen-full of output is sent to a terminal without any reception of input inbetween. Any input received afterwards will take the terminal out of page-wait and output transmission will continue.

The TIP will display a message (OVER..) at the bottom of the screen, reflecting the page-wait condition and the fact that more output is available.

The recommended way to take a terminal out of page-wait is by hitting the CLEAR key. This will cause the screen to clear and the next line of output to start at the top-of-screen.

13.1.2 Auto-input

After a block of auto-input is sent to the terminal, the TIP will upon reception of a text-block from the terminal, send the first 20 characters of output and the received text as one block to the host-application that requested this auto-input.

13.1.3 IVT Commands

The TIP allows the following IVT commands to be sent by the host-application or the terminal-user:

- PW Page width (default - 80, range 20-255)
- PL Page length (default - 24, range 12-255)
- PG Paging on/off (default - ON range ON, OFF)
- CT Control character (default - %)
- TC Terminal class (must enter % TC = 28)

User breaks are supported but not as special alterable input-characters. The TIP interprets the Program-attention keys as breaks. (PA1 = User-break 1, PA2 = User-break 2).

The TIP does not support the Cancel, because the terminal allows the user to locally cancel text before hitting the ENTER (send) key.

13.1.4 Format-effectors

The TIP allows all valid Format-effectors with the exception of the NO-ACTION which is treated as Pre-Print-one.

Home-Cursor and Clear-Screen format-effectors will force the terminal into page-wait unless the last interaction with the terminal was input.

A Post-print FE will be given an additional pre-print-one after input is received, such that output will not overwrite the input on the screen.

13.1.5 Downline CR/LF Inserted

The TIP replaces a CR/LF or LF/CR sequence logically by a Format-effector, Pre-Print-One. CR/LF sequences can force the terminal into page-wait in the middle of a logical line.

13.1.6 Line Folding

If downline logical-lines are larger than the page-width, the TIP will perform line-folding. Folded lines will not force the TIP into page-wait. More than 3 folded lines out of one logical line can cause the TIP to overwrite text on the screen not yet seen by the user.

13.1.7 Cursor-control

In the non-transparent mode, the TIP supplies all cursor-addresses for output. The terminal-user has control over where the next line of output goes after input: After a line of input, the TIP will start output on the screen one line from where the user left the cursor (normally this will be the next line after input).

13.2 TRANSPARENT MODE

In this mode, all the data-protect features of the 3270 display are available to the application.

13.2.1 Downline

The application will have to construct a screen-full of protected/unprotected fields and supply all the desired attribute-characters and screen-buffer-addresses for the fields. The first character of a downline block must be a valid (allowed) command-code (such as clear-screen, etc.) The TIP remains responsible for preceding the block of output by SYNC-characters, Start-of-text, and Escape-char, and attaches ETX, CRC, PAD at the end. The TIP will also translate all downline data ASCII to EBCDIC and perform SYNC-fill.

Allowed Command-codes are:

- Normal-write x'31'
- Erase-write x'35'
- Erase-unprotected x'3F'

Invalid Command-codes are replaced by an x'31' (normal write).

A typical start of a field would be:

- SBA Set-buffer-address x'11'
- BA1 Buffer-address-1 See 3270 manuals,
- BA2 Buffer-address-2 all in ASCII.
- ATT Attribute-char.

The attribute-character determines the characteristics of the field:

- protected
- unprotected
- intensified
- numeric shift
- etc.

The application is also expected to insert the cursor at a desired position.

Once transparent output is delivered to a 3270 terminal, the TIP will assume transparent input until a non-transparent downline block is delivered to terminal.

To protect the integrity of the protocol the TIP will replace certain downline characters by NULL's. The replace characters are:

SOH, STX, ETX, EOT, ENQ, ACK, NAK, SYNC

13.2.2 Upline

After transparent output is delivered, the TIP will send up to the host all the modified unprotected fields received from the terminal inclusive of the SBA and Buffer-address-chars (2) of each field. (The terminal does not send the Attribute characters back to the TIP.)

If the incoming text is larger than one transmission block (boundary is 256 characters), the TIP will send BLK/BLK/../MSG, such that the application can reproduce a full screen.

13.3 SWITCHING BETWEEN INTERACTIVE AND TRANSPARENT MODES

The TIP will receive input in transparent mode as long as the applications sends downline transparent data. So the first transparent output will signal to the TIP to receive input in transparent mode and vice versa. Note the TIP command to interpret input in transparent mode is supported by the TIP (but it does not seem to be to useful).

TABLE 1

REMOTE STATUS AND SENSE BYTE DEFINITION

Bit No.	Bit Definition
	S/S Byte 0:
0,1	Not Used
2	Reserved.
3	Reserved.
4	Device Busy (DB) - This bit indicates that the addressed device is busy executing an operation or that a busy detection was previously made by a command or a print operation, accepting data from the Operator Identification Card Reader, or performing various keyboard operations (Erase Input, Backtab, and Clear). This bit is set with Operation Check when a Copy command is received which specifies a "busy" device with its "from" address. This bit is set with Unit Specify when a command is addressed to a busy device. This can occur by chaining a command to a Write, Erase/Write, or Copy command which started a Printer or by chaining a command to a Specific Poll addressed to a busy device.
5	Unit Specify (US) - This bit is set if any S/S bit is set as a result of a device-detected error, if a command is addressed to a busy device.
6	Device End (DE) - This bit indicates that the addressed device has changed from unavailable to available and not ready to ready, or busy to not busy. This bit is included during a Specific or General Poll but is not considered pending status by a Selection Addressing sequence.

addressed device has pending status and also detects one of the above status changes that warrants a Device End, then the Device End bit is set and preserved along with the other pending status, and an RVI response is made.

7 Transmission Check (TC) - Not used by the 3271. This bit is set when the 3275 detects a BCC error on the TCU transmission.

S/S Byte 1:

0,1 Not Used.

2 Command Reject (CR) - This bit is set upon receipt of an invalid 3270 command (or Copy command if this feature is not installed).

3 Intervention Required (IR) - This bit is set if:

- A Copy command contains a "from" address in its data stream which specifies an unavailable device.

3 Intervention Required (IR) - This bit is set if:

- The 3271 receives a Selection Addressing sequence or a Specific Poll sequence for a device which is unavailable or which became not ready. A general Poll sequence does not respond to the unavailable/not ready indication and proceeds to determine the state of the next device.

- The 3271 receives a command for a device which the 3271 has logged as unavailable or not ready.

4 Equipment Check (EC) - This bit indicates 3271 detected bad parity from the device.

5 Data Check (DC) - This bit indicates the detection of a parity or Cursor check in either the 3271 or a device buffer or in the 3275 buffer, or 3271 detected bad parity from the device.

6 Control Check (CC) - This bit is set with Control Check, Intervention, Data Check, Device Busy, or Data Check with Unit Specify to indicate that the errors that set these sense bits were detected while the 3271 was executing an operation with the "from" device during a Copy command. This bit is set with Unit Specify to indicate that the "from" address on a Copy command specified a device with a "locked" buffer (the device data is secure).