EXTERNAL REFERENCE SPECIFICATION (ERS)

COMMUNICATIONS CONTROL INTERCOM 3.0 COMMUNICATIONS SUBSYSTEM FOR NOS/BE INTERCOM 5

74872890

NETWORK PROCESSING UNIT

34843-053-502

ORIGINATOR				
E. P. Lamoureux	J. W. Townsend	P. W. Hagen	D. Marek	R. Kerr

REVISION RECORD

REV.	E.C.O.	PAGE	PARAGRAPH	DESCRIPTION	APVD.	DATE
A	06577			Initial Release		
B				Major Changes		
C				Changes to BISYNC & TTY TIP, Name Change to CCI		
D				Major Changes		

TABLE OF CONTENTS

TABLE OF CONTENTS (Con't)

TABLE OF CONTENTS (Con't)

TABLE OF CONTENTS (Con't)

TABLE OF CONTENTS (Con't)

TABLE OF CONTENTS (Con't)

LIST OF TABLES

APPENDIX A

1.0 INTRODUCTION

1.1 Scope

This External Reference Specification (ERS) provides the functional level description and the detailed interface specifications for the Communications Control INTERCOM. The functional level is primarily covered in Sections 1 and 2 which describe the objectives, general approach, and features to be added that are different from the Standard Product INTERCOM 4.5/CCP 1.0 Communications Subsystem. The detailed specifications cover the interfaces between INTERCOM and CCI such that it will be the programming reference document for this implementation.

1.2 Objectives

The primary communications requirements are to:

1) Support existing INTERCOM/CCP 1.0 features under NOS BE.

2) Add 2780/3780 point to point and HASP multileaving workstation support.

3) Be implemented in 6 to 9 months elapsed time.

4) Provide throughput capabilities for supporting up to 25 RJE terminals at line speeds up to 19.2K bps.

5) Reduce host resources used for communications (central memory and PPU time).

All of the above requirements must be met by the project.

The following secondary requirements will be considered in the design:

1) The special software produced should be a logical step toward spooling batch traffic directly to the host disk.

2) The design should be compatible with the standard product NAM and CCP 3.1 wherever possible to utilize the software already developed for CCP 3.1.

3) The resultant INTERCOM driver will support two or three 255X per host channel but multiple 255X's can be driven from different channels and PPUs.

4) CCI will run in the 2550-2 and support memory size 65K words or greater. Since the CCI software will be developed using the same Base System Software as CCP 3.1 uses (CCP BDL002), it is possible to run 2552 hardware.

5) Although an asynchronous terminal batch mode feature would be valuable for this implementation, it will not be considered a requirement.

6) The Mode 4 job stream must be compatible with existing standard products.

1.3 Overview

1.3.1 Background

The CCI Program is intended to provide a more efficient base for enhanced RJE terminal support. The philosophy of this project is to move the protocol and buffering responsibilities for RJE data streams into the NPU. This will simplify the interface to the host and permit the NPU to expand as the network grows with very little impact on the host system. The RJE data interface between the NPU and host will be disk formatted Physical Record Units (PRU). The host will pass output PRUs to the NPU for delivery to the output devices in the network and the NPU will pass input PRUs to the host for recording on Rotating Mass Storage (RMS). The interactive data paths will be the same as they are currently for CCP 1.0/INTERCOM 4.5. The software provided will support the currently supported protocols, Mode 3 and Mode 4 plus 2780/3780 Binary Synchronous Communications (BSC) and HASP multileaving workstations.

1.3.2 Data Flow

The data flow in the system, from the network to the host system, can be divided into three classes: interactive input and output, RJE input, and RJE output (Figure 1).

1.3.2.1 Interactive Data

The interactive input and output streams are functionally the same as INTERCOM/CCP 1.0. Formats are changed to be CCP 3.1 compatible. The data is passed between the NPU and host as line images coded in ASCII code. Interactive input is accumulated by the NPU until a line is completed and then passed to the host system. It is the responsibility of the host system to process the message or to deliver it to the application running under INTERCOM as required. Interactive output is passed from the host system to the NPU in ASCII coded line images. It is the responsibility of the NPU to deliver these messages to the proper device at the terminal.

1.3.2.2 RJE Input Data

RJE input is received by the NPU, transformed to a standard format, and accumulated as PRU images. The PRU blocks are then transferred by the NPU to the host. The host system is responsible for writing the data to RMS. End of Record (EOR) and End of Information (EOI) blocks are transferred as short PRU blocks. The number of physical PRUs transferred as a block across the coupler is a system assembly parameter.

1.3.2.3 RJE Output Data

RJE output is retrieved from RMS by the host system and tranferred across the coupler to the NPU as PRU images. End of Record and End of Information are indicated in the message header. The start of a new output file is indicated by the host system as a banner message and the NPU may (user option) generate a banner block for the terminal from the file name provided in the banner message. The NPU is responsible for blocking line images from the PRU block record and resolving the logical format control in the data record to the physical characteristics of the output device on the terminal.

CCI -- PHASE 1

DATA FLOW

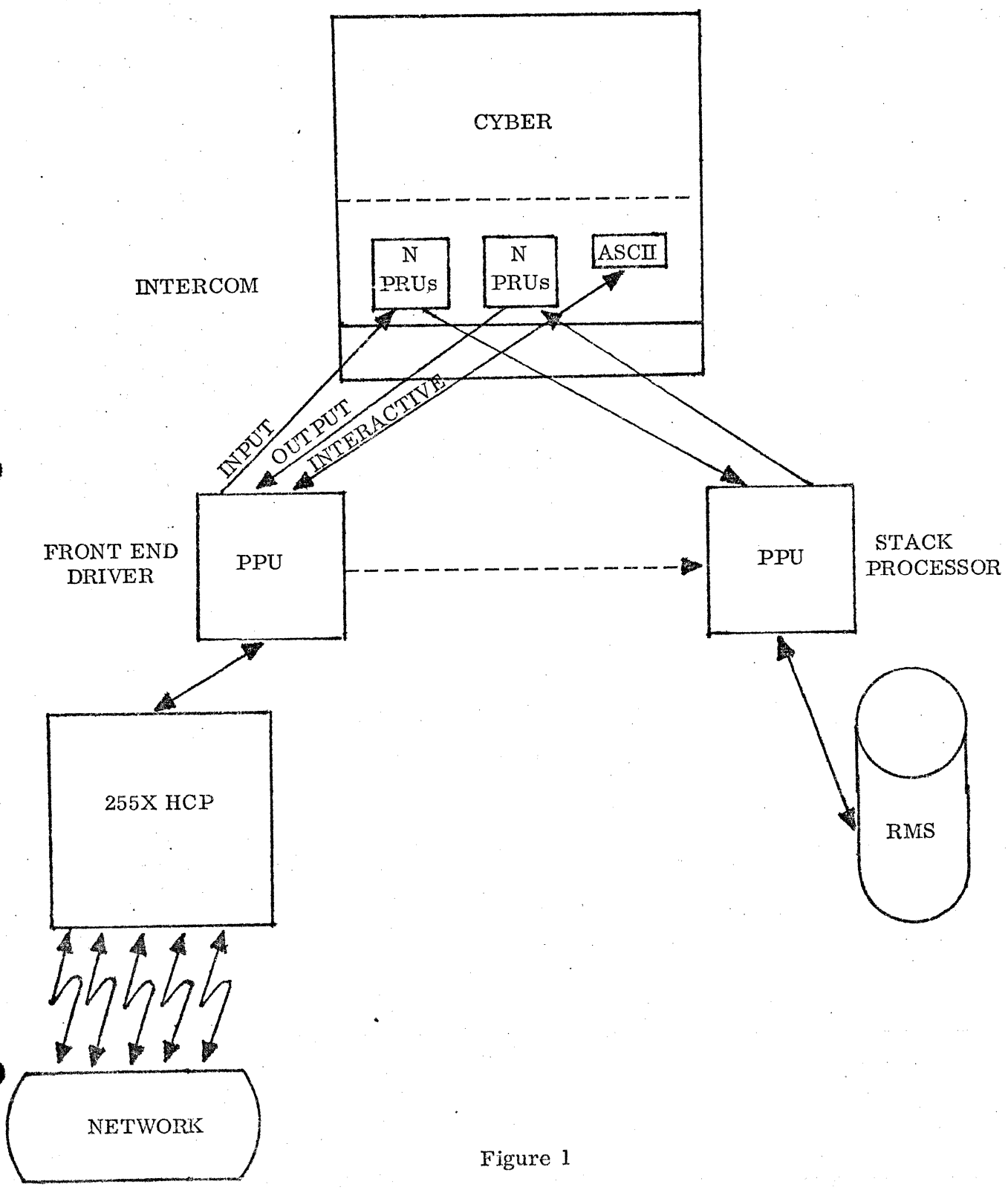


Figure 1

1.3.3 Data Flow Control

Data flow will be controlled by connection numbers. Each terminal will have at least one full duplex connection number, the interactive data stream. The interactive is always defined for an active terminal. This data stream can be considered a full duplex channel by which the terminal operator can control the operations of his terminal. Because of hardware restrictions of some terminals, the use of this connection number may be mutually exclusive with all batch data flow. For Mode 4A and 2780/3780, interactive data or batch data will be active but not both simultaneously. For HASP workstations and Mode 4C, both batch and interactive data streams may be active simultaneously.

1.3.3.1 RJE Input Control

Each RJE input stream will have its own connection number and will be initiated by a command to the NPU from the host system. The command will specify the mode of the transmission; either display code or transparent.

There are two types of input transmissions supported. The user command "READ, FILENAME, MODE" permits data to be transferred from the input device on the terminal to a local file on the host system. Only a single data file may be transferred with this command. The data will consist of all data up to the EOI. An EOI is defined as an EOI card for display code transmissions and as the end of the input data stream for transparent transmissions. When the EOI is received by the NPU, the final data block is sent to the host system.

The second type of input transmission is job stream input. A job stream input is inherently a display code transmission. A job within the stream is terminated by an EOI card. When the EOI is received by the NPU, the final data block for that job is sent to the host. The host is responsible for completing that job file and submitting the input job to the input queue of the host.

The facility to translate input records according to the 026/029 Code Set option is supported for BSC (2780/3780/HASP) by having the NPU examine columns 79 and 80 of Job cards and End of Record cards. The default code (026/029) is set by the Host. In addition, the last record of a job deck stream can be in transparent mode by using columns 79 and 80 to specify TR for transparent. In that case, the EOI for that job deck would be determined by receiving an end of data (ETX) indication from the remote terminal. At the start of each job the code set is set to the initial value established by the Host, unless changed by the Job card. If there is no code set specified on an EOR card, it remains the same as before (default or as set by the Job card).

Input stream errors such as "Input Device Not Ready" will be reported upline if the terminal hardware permits by use of an asynchronous upline command stating that the input has stopped and the reason. The NPU is responsible for maintaining any accumulated data for the stream. The host will either restart or terminate the input stream with a downline command as the result of a command interchange with the terminal operator.

The NPU initiates the initial transfer of an RJE input stream. Subsequent blocks must wait for a downline acknowledgement from the host.

1.3.3.2 RJE Output Control

Each RJE output stream will have its own connection and output will occur when available for the stream and the terminal is in a state to permit reception of the data.

The output data will be preceded by a file identification record containing the name of the file. The NPU will convert the file name to a banner message that is consistent with the output device at the terminal. The file may be either transparent or display code as specified in the block protocol header.

For print files, the NPU will be responsible for blocking line images to the terminal and processing the carriage control characters consistent with the print device on the terminal. The detection of a print line with "PM" as the first two characters requires that the NPU direct that line to the interactive output device and notify the host with an upline command that the print stream has been stopped by a PM message. Data compression will be the responsibility of the NPU.

Output stream errors such as output device not ready will be reported upline, if the terminal hardware permits, by use of an upline command stating that the output stream has been stopped and the reason. The NPU is responsible for maintaining any undelivered data for the stream. Depending on the terminal characteristics, either the NPU will restart the transmission if the terminal signals the end of the error condition, or the host will either restart or terminate the output stream as a result of a command interchange with the terminal operator.

The NPU controls the flow of output data by use of the upline acknowledgment. The upline acknowledgment in effect requests the next output block for that data stream.

2.0 COMMUNICATION SUBSYSTEM FEATURES

The standard product INTERCOM 4.5/CCP 1.0 features will be supported by this implementation with exceptions and expansions as noted in this section. The resultant INTERCOM driver will support only the 255X front-end (not 667X's or LCC's). Multiple 255X's may be driven from different channels and PPU's.

Generally it can be assumed, unless otherwise specified, that INTERCOM Interactive and Remote Batch features will be functionally the same as the standard INTERCOM 4.5/CCP 1.0 release, although the method of implementation will be different. Specifically, job stream command formats and terminal operations for terminals supported by the standard product must be compatible.

2.1 Feature Expansions

The following sections discuss the exceptions and extensions of INTERCOM features necessary to support the HASP workstation, 2780 and 3780 terminals.

2.1.1 Terminals Supported

The terminals supported are shown in Table 2-1.

2.1.2 Line Types Supported

The line types supported are shown on Table 2-2.

2.1.3 Remote Batch Facilities

EOR/EOI

Only 7/8/9 and 6/7/8/9 end of record (EOR) and end of information (EOI) cards are required by CDC terminals. Since IBM readers treat multi-punches in columns 1 thru 7 as errors, EOI cards for 2780/3780 and HASP workstations are punched as /*EOI in columns one thru five. /*EOS may also be used to indicate end of information for the HASP workstation and additionally terminates the input stream.

A level number placed in columns 2 and 3 of the EOR card will be supported. For punch output, the level number will be punched in columns 2 and 3 of the EOR card.

Binary Cards

Binary card deck input/output is not supported by any terminal.

026/029 Codes

A "26" or "29" punched in columns 79 and 80 of the JOB CARD or EOR cards will change the input code translation for 2780, 3780 and HASP terminals. Output to punched cards will always be 026 Code.

2.1.3 Remote Batch Facilities (Con't)

Transparent Data

8-bit data may be exchanged transparently with 2780, 3780, and HASP workstations having transparent options, but not Mode 4 terminals. Transparent input may be defined by either a READ, FILENAME, MODE command where mode specifies "transparent" or by punching "TR" in columns 79 and 80 of the EOR card. Transparent mode will be turned off on receipt of logical EOI (last card transmitted). /* EOI is not detected in the transparent mode.

Transparent input data will be written to RMS (stored 8-bits in 12 right justified, 5 characters per CYBER word) without performing any code translation or character expansions.

Transparent output is created using parameters on the ROUTE control card. Transparent output is selected to be delivered to the terminal using the DEFINE terminal command (See Intercom reference manual. Transparent output files will be delivered to the terminal without performing any character compression, code translation, carriage control conversions, or print line/card blocking.

Carriage Control

Printer carriage control for batch output terminal printers is controlled by the first character of each line. Only control characters now supported on Mode 4 terminals which have valid equivalents for 2780, 3780, and HASP are converted, all others result in a single space. Table 2-3 describes the INTERCOM carriage control and equivalents for the supported terminals.

The "suppress" print file carriage control command will be supported on all terminal types.

The horizontal and vertical tab features of 2780, 3780 and HASP terminals will not be supported.

Interactive Carriage Control

Standard INTERCOM carriage control characters for the Teletype and Mode 4 CRT are supported. These control characters will be translated to equivalents when interactive output is being delivered to a 2780 or 3780 line printer or HASP console. Control characters and their equivalents for each of the terminal types are shown in Table 2-4.

Note that carriage control characters on interactive data are not interpreted directly in the 2550 TIP. Conversions from INTERCOM carriage control characters are converted by INTERCOM to a Data Block Clarifier (DBC) code which is interpreted by the TIP. See the TIP descriptions for an explanation of the DBC codes.

Table 2-1 Terminal Types

TIP Type	General Description	Specific Terminals Supported
TTY	10 cps, 110 baud	Teletype M33, M35, and M38 CDC 713-10
TTY	15 cps, 150 baud	Teletype M37 CDC 713-10
TTY	30 cps, 300 baud	CDC 713-10
TTY	Automatic recognition of above line speeds	Any teletype compatible terminals specified above
TTY	60 cps, (600 baud) 120 cps (1200 baud) 240 cps (2400 baud) 960 cps (9600 baud)	TTY compatible
MD4	Mode 4A BCD	214, 217 (200 UT), 731-12, 732-12, 734-1
MD4	Mode 4A ASCII	217, 731-12, 732-12, 734-1
MD4	Mode 4C	711-10, 714-10/20
MD4	Automatic recognition of above by repeated poll of Controllers 70, 71, and 72	Any Mode 4 terminals specified above, after address strapping which implies terminal type from controller address
BSC	BSC Point to Point EBCDIC	IBM 2780, 3780, or equivalent
HASP	HASP Multileaving, EBCDIC	IBM 360/25 HASP workstation or equivalent
AUTO BSC/ HASP	Automatic recognition of 2780/3780/HASP	Any of the above 2780, 3780 or HASP terminals

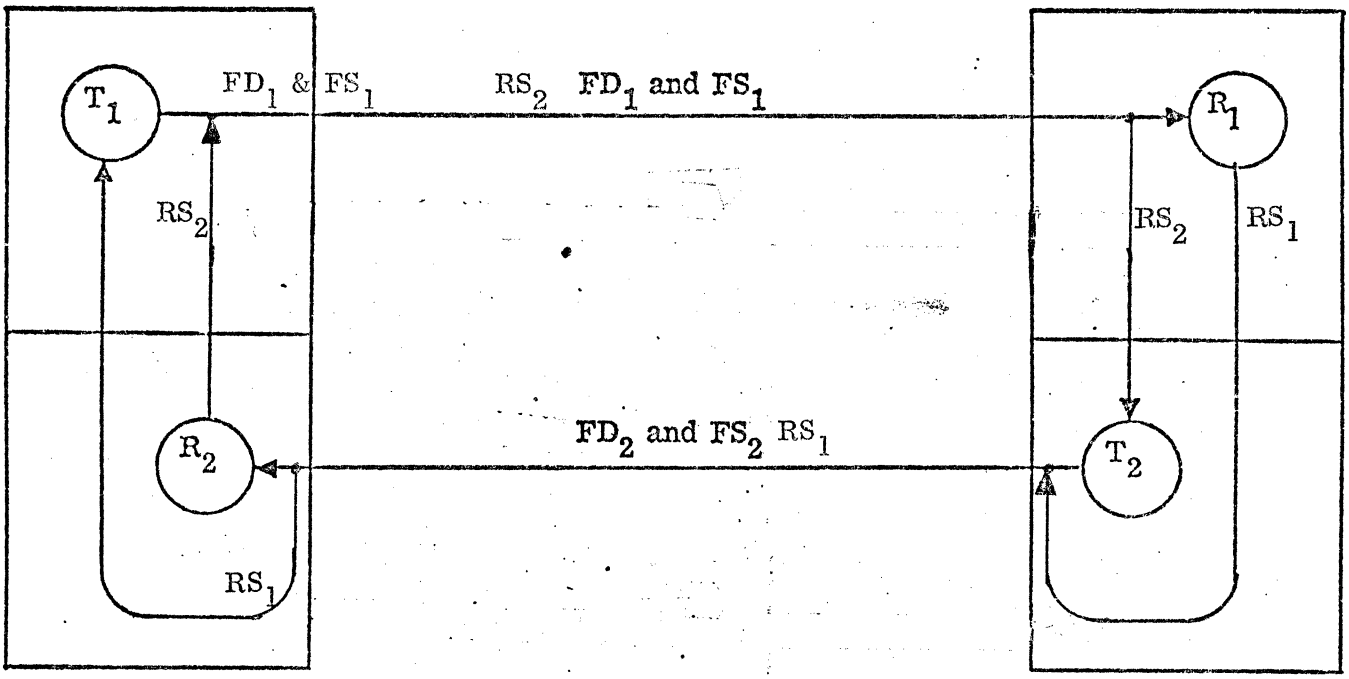


Figure 2-1 Communications Paths for Block Flow Control

2.1.3 Remote Batch Facilities (Con't)

Punch Files

Punch files are supported to the 2780, 3780 and HASP terminals if they are configured with punch devices. Punch files are specified by setting the Forms Code. Intercom will recognize the Forms Code and identify the file to the 2550 by the connection number.

Output files to the punch will be proceeded by the "banner message" (generated by Intercom) which results in a "lace file separator card" punched as a "nul" in columns 1 thru 60 and the Job Name in columns 71-80. Each card record must not contain the carriage control character and may contain from 1 to 80 characters. Short cards may "optionally" be punched with the BSC record separator in the last column for 2780 and 3780 terminals by selection within the Intercom DEFINE command.

Output files to the punch may also be in transparent. In this case the user is responsible to ensure all 80 columns are present.

Compression/Expansion

Data received from the terminal in compressed form will be expanded to the "Standard" SCOPE file format before the record is written to disc. Conversely, data read from an output file is compressed before transmission to the terminal. (Both compression and expansion are performed within the 255X). The methods that may be used for compression and expansion are different for each of the terminal types supported and not all of the available methods may be supported by the 255X.

Mode 4

For input card data, trailing blanks will be suppressed and the end of card indicated according to the standard SCOPE file format. Output data to the printer will have non-trailing zeros and blanks compressed. Trailing blanks will not be transmitted.

2780

Trailing blanks for input non-transparent card data will be suppressed and the end of card indicated according to the standard SCOPE file format. Trailing blanks will not be transmitted to the printer or punch when in the non-transparent mode.

3780

Non-trailing blanks will be expanded for input non-transparent card data. Trailing blanks will be suppressed and the end of card indicated according to the standard SCOPE file format. Output non-transparent data will have all blanks compressed whenever possible.

HASP

All compressed characters will be expanded except trailing blanks for input non-transparent card data. The end of card will be indicated according to the standard SCOPE file format. All like character strings will be compressed for non-transparent output to the printer or punch.

2.1.4 Terminal Features

Each of the terminals supported by this communication subsystem have characteristics and features which are not essential for normal operation or are optional selections for the terminal itself. This section will list these features for each terminal and specify if they are supported.

Note that line type (including speed) support and terminal support are independent to the communication subsystem. Therefore, it may be assumed that any combination of line type and terminal, that is supported by both the terminal and the communication subsystem, may be used.

2.1.4.1 TTY (Mode 3)

The features supported for TTY terminals will be the same as implemented for the standard Network Products Release 1. In addition, all asynchronous line types listed in Table 2-2 are supported and "block mode" input is supported.

2.1.4.2 Mode 4

The features supported for Mode 4 terminals will be the same as implemented for the standard Network Products Release 1. In addition, all synchronous line types listed in Table 2-2 are supported.

2.1.4.3 2780/3780

The following features of the 2780/3780 devices are supported:

Feature	2780	3780
Character Set	EBCDIC	EBCDIC
Horizontal Format Control	No	No
EBCDIC Transparency	Yes	Yes
Multiple Record Feature	Yes	N/A
Space Compression/Expansion	N/A	Yes
Print Width	80-150	80-150
Punch/Component Selection	Yes	Yes
Line Speeds	2000-9600 bps	2000-9600 bps
Printer Character Set	EBCDIC 63	EBCDIC 63
Multi-point	No	No
Terminal ID	Accept. (not checked)	Accept. (not checked)

2.1.4.3 2780/3780 (cont.)

Feature	2780	3780
Conversational Mode	Not used by 2550	Not used by 2550
Processor Interrupt	Not used by 2550	Not used by 2550
Multiple cards in Transparent	OK	OK

2.1.4.4 HASP Workstation

The following features of the HASP workstations are supported.

Feature	Supported
Multi-card	Yes
Character Set	EBCDIC - 64
EBCDIC Transparency (256 char)	Yes
Character Compression/ Expansion	All character strings
Console	Mandatory
No-Console	Not Supported
Printer Character Set	EBCDIC - 64
Punch	Yes
Print Width	80-150
Binary Cards	No
Plotter, Magnetic Tape, Paper Tape	As Card/print emulation or Transparent data
Line Speeds	2.0 - 19.2Kb

Transmission Facility	CLA Type	Modem Type	Answer Mode	Carrier Type	Circuit Type	Turn-Around Required	Turn-Around Delayed	Transmission Mode
HDX	2560-1	RS232-201A/208B Compatible	Switched	Controlled	2 Wire	YES	NO	Synchronous
FDX*	2560-1	RS232-201B/208A Compatible	Dedicated	Controlled	4 Wire	YES	NO	Synchronous
FDX	2560-1	RS232-201B/208A Compatible	Dedicated	Constant	4 Wire	NO	NO	Synchronous
FDX	2561-1	RS232-103E/113 Compatible	Switched	Constant	2 Wire	NO	NO	Asynchronous
FDX	2561-1	RS232-103E Compatible	Dedicated	Constant	2 Wire	NO	NO	Asynchronous

* Operating with HDX Protocol

Table 2-2 Line Types

Table 2-3

BATCH CARRIAGE CONTROL

Intercom Control Character	Mode 4		3780		HASP		2780	
	Before	After	Before	After	Before	After	Before	After
1	New Page		New Page		New Page		New Page	Space 1
+	No space		No space		No space			Space 1
0	Space 2		Space 2		Space 2		Space 1	Space 1
-	Space 3		Space 3		Space 3		Space 2	Space 1
b	Space 1		Space 1		Space 1			Space 1
All others	Space 1		Space 1		Space 1			Space 1

Table 2-4

Interactive Carriage Control

Intercom Control Characters	TTY		Mode 4		2780/3780		HASP	
	Before	After	Before	After	Before	After	Before	After
1	CR,3LF		Clear Write	New Line	Skip to Top of Page	Space 1	Not Applicable	Not Applicable
*	CR,3LF		Reset Write	New Line	Skip to Top of Page	Space 1	-	-
+	CR		-	New Line	*	2780-space 1 3780-No space		
0	CR, LF, CR, LF		New Line	New Line	Space 1			
-	CR, LF, CR, LF CR, LF		New Line New Line	New Line	Space 2	Space 1		
Blank	CR, LF		-	New Line		Space 1		

Note: Carriage control to the console on the HASP terminal is controlled by the terminal itself.

3.0 DATA TRANSPORT

3.1 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 function of the type of source transmitting the data.

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 but not necessarily between the separate paths.

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.

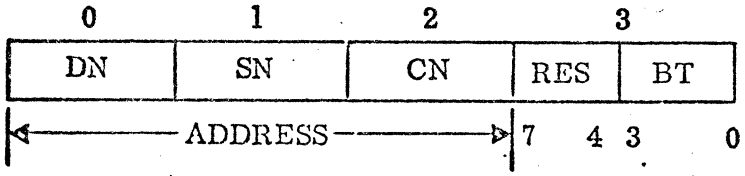
3.1 Block Flow Control (BFC) Procedure (continued)

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 abnormals at the receiver.

The block formats described in Section 3.1 are those defined for the Standard Product Release 4 CCP. CCI uses only a subset of the total BFC capabilities as noted within the brackets.

3.1.1 Block Formats

Every block will have a header consisting of four bytes. The first three bytes provide a standard network address. The last four bits of the last byte indicate the Block Type (BT), and the other four bits in this byte are reserved for Network use. The content of the remainder of the block, if any, varies with the BT. Figure 2-2 depicts the general block header.

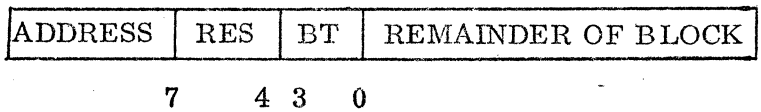


DN Destination Node
SN Source Node
CN Connection Number
BT Block Type
RES Reserved for Network use -

CCI 3.0 - Bit 7 not used.
Bits 4 - 6 = Modulo 8 Block Serial Number (BSN) for each logical connection

Figure 2-2 Block Header Format

Since the address portion of all blocks is the same, all block formats subsequently presented in this specification will be shown in the general form:



3.1.2 Addressing

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

3.1.2.1 Node

A system of host and NPU front ends consists of two or more nodes. Each NPU has one to four unique node IDs. Each interface between host and an NPU has one to four unique node IDs. The set of all assigned node IDs is a dense set whose lowest value is one. Node ID = 0 is reserved for the host.

Host interfaces are assigned lower values than those assigned to terminal nodes. 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. More than one node ID is assigned to a terminal node when the number of logical connections which must be made between a host node-NPU pair exceeds 255. (CCI 3A only uses one NPU NODE ID)

3.1.2.2 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, are unique. A separate connection number will be assigned for each device on a terminal, e.g., Card Reader(s), Line Printer(s), Punch(es). Also each interactive data stream will have a separate connection number and TCB even if the same device is used for batch input and output.

3.1.2.3 Service Channel

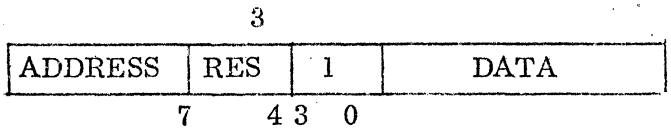
A block having a connection number equal to zero is called a Service Message, and the logical connection via which it is communicated is called the Service Channel. Unlike logical connections which can be dynamically created and destroyed, the Service Channel always exists. Service Messages are always 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.

3.1.3 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.

3.1.3.1 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 0 to 2043 bytes of data immediately following the four byte header. The content of the data field is determined arbitrarily by the communicating process. The format of a BLK block is shown below.



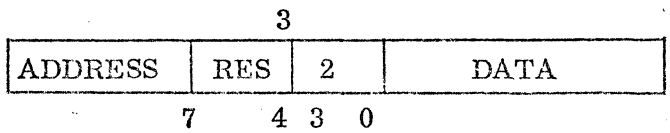
[CCI 3.0 - A block that does not contain EOR or EOI]

3.1.3.2 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 for some other reason it is desired to segment the message, all segments but the last are transmitted via BLK blocks, and the last segment is transmitted via a MSG block.

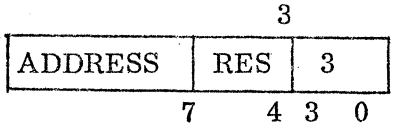
The format for a MSG block is shown below.



[CCI 3.0 - Each block containing EOR or EOI will be a "msg" block.]

3.1.3.3 BACK (RS)

A BACK, shown below, is sent as BLK, MSG and CMD blocks are sent from 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 an Allowable Block Limit (ABL) for each connection. The BACK, which acknowledges a previously transmitted block, allows the transmitter to maintain an outstanding block count to ensure that the ABL is not exceeded.



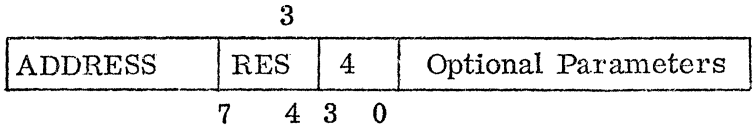
The Allowable Block Limit is established at each end of the connection as part of the configuration process.

[CCI 3.0 - only ABL = 1 will be implemented, i.e., each block must have associated BACK and only one may be transmitted before the BACK is received.]

3.1.3.4 CMD (FD)

CCI 3.0 - Some CMDS will be asynchronous with the data stream; therefore a CMD could be either FS or FD depending on the command

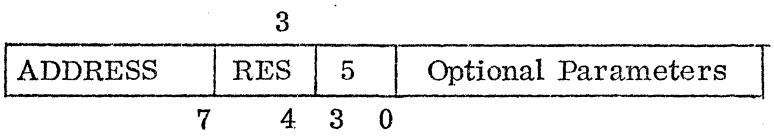
A command is provided to allow connected processes to communicate outside of the data stream but synchronous with that stream. The command will be received by the destination process in the same ordering sequence to the data stream or other commands as existed at source.



The parameters of a CMD are bilaterally defined by the communicating process. Note that a CMD with a CN of 0 has special system significance as a Service Message.

3.1.3.5 BRK (RS)

The BRK indicates a discontinuity in the data stream travelling 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 START must not be sent prior to receipt of the RST.

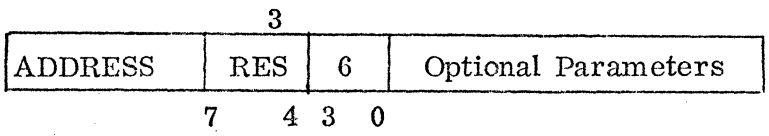


Any optional parameters included with the break will be passed to the connecting process. Such parameters are outside the block protocol.

CCI 3.0 - Not Used

3.1.3.6 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.

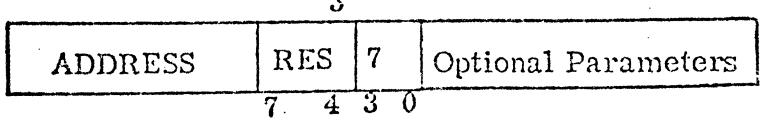


The use of a STP is indicated when a process is unable to deliver data to final destination. Examples of this are terminal inoperative, not ready, line inoperative, etc. Any optional parameters included with the STP will be passed to the connected process. The sender of the STP will discard all blocks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

CCI 3.0 - Not Used

3.1.3.7 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.

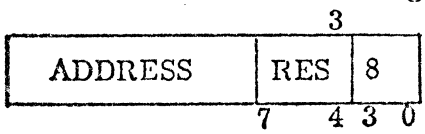


Any optional parameters are passed to the connected process.

CCI 3.0 - Not used

3.1.3.8 RST (FS)

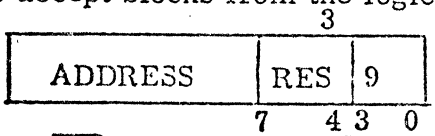
The RST is sent in response to either a BRK or STRT. It serves to delimit the data stream and indicate the point in the data stream at which the BRK or STRT was actioned. From the time the BRK or STRT was sent until the receipt of the RST all unacknowledged blocks and all new blocks are discarded.



CCI 3.0 - Not used

3.1.3.9 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.



CCI 3.0 - Not Used

3.1.3.10 Bad Blocks

When NPU software detects a bad block, i.e., any block with fields that contain unexpected or undefined information, the NPU will discard the block. If the block is a BLK or MSG, no BACK is sent to the host. For any 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.

Table 3-1 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.

Codes 0, 12, 13, 14, 15 are reserved for Network Assurance Protocol

Codes 10 and 11 are spare

3.1.4 Block Formats

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 its type and source. All blocks are of three general formats within the system.

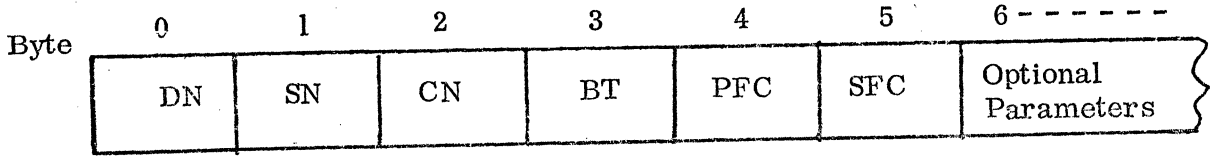
1) Non-data blocks are blocks that travel interactive, batch, or the service message channels and are of type other than BLK or MSG (e.g., CMD, BACK).

2) Batch data blocks are blocks that travel on a batch connection that are of type BLK or MSG.

3) Interactive data blocks are blocks that travel on an interactive connection that are of type BLK or MSG.

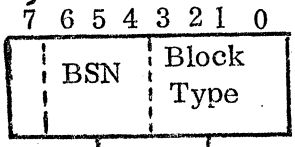
3.1.4.1 Non-Data Blocks

For the CCI 3.0 implementation Non-Data Blocks consist only of BACKs and CMDs, and have the following general format:



DN - Destination Node
SN - Source Node
CN - Connection Number
BT - Block Type

Address present on all blocks. Reference Section 3.1.1



1 - BLK
2 - MSG
3 - BACK
4 - CMD

Block Serial Number *

PFC - Primary Function Code
SFC - Secondary Function Code

See Section 5 and Appendix A for Service Message codes.
See Table 3-1 for code used on non zero connections

* The BSN field is not used on connection number zero. BSNs on non zero connections are assigned sequentially, modulo 8, by the block originator. Numbering starts with zero when a connection is established and is independent for upline and downline blocks - i.e., no correlation to upline and downline serial numbers on a connection may be assumed. BSNs are used for diagnostic and debugging aids only.

3.1.4.1.1 Commands

CMDs on the service channel (connection number zero) are generally called Service Messages (SMs). The BSN field is always zero for CMDs on the service channel. Section 5 and Appendix A describe all Service Messages used within CCI 3.0.

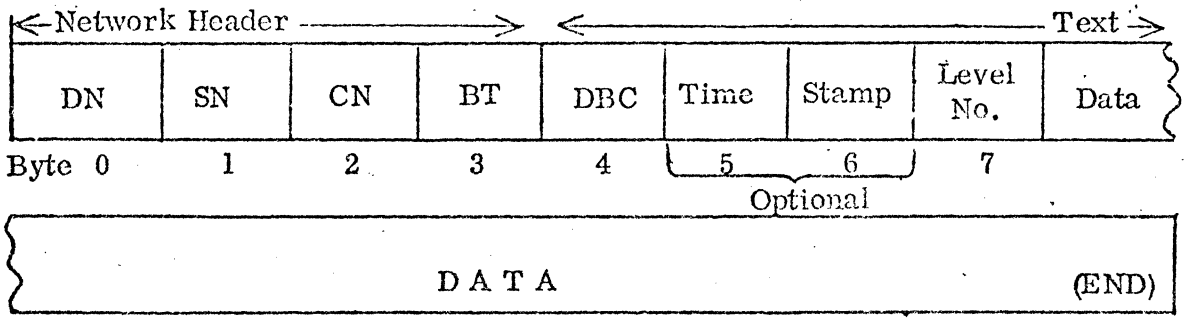
CMDs are also used on non-zero connections to control data streams, control terminal nodes, and to report status between the host and the NPU. CMDs may use either the Reverse Supervision (RS) or Forward Data (FD) channels of a connection, depending on the type of command. In either case the CMD flows asynchronous to any data block on the channel and, therefore, cannot be used to mark position within the data stream. Commands received on a connection are not acknowledged by a BACK in the reverse direction. Table 3-1 summarizes the general format and PFC and CMD used on non-zero connections within CCI 3.0.

BACKs are used to acknowledge receipt of data blocks and to solicit the next data block on a connection. For CCI 3.0, only one data block may be transmitted on a connection before the BACK is received for that block. That is, the number of outstanding data blocks (allowable block limit) is always one. BACKS in this system do not provide for block assurance, but are used for flow control only.

3.1.4.2 Data Block Formats

Upline and downline data (block types BLK and MSG have the following general format for both interactive and batch data blocks.

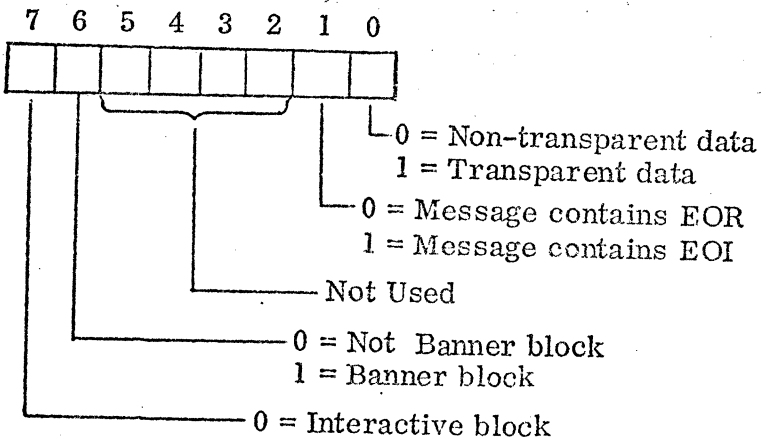
DATA BLOCK FORMAT



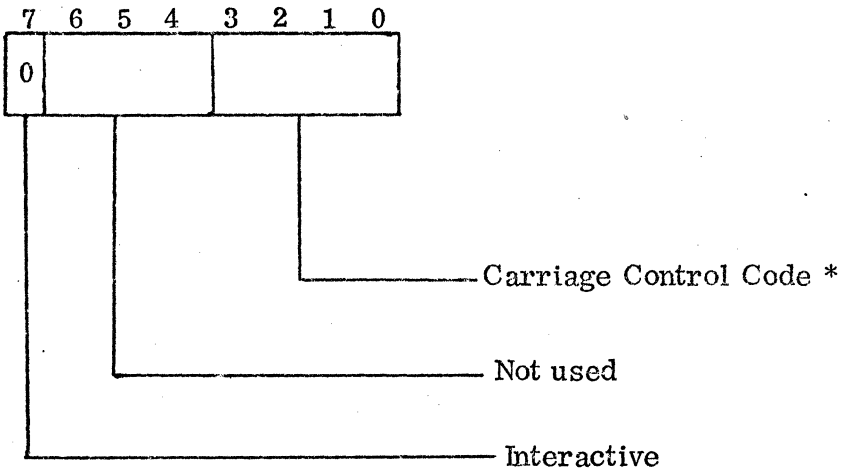
DN - Destination Node
SN - Source Node
CN - Connection Number
} Address present on all blocks - Reference Section 3.1.1

BT - Same as non-data blocks' - Reference 3.1.4.1

DBC for Batch Data Block



DBC for an Interactive Block



* Carriage Control Code	General Function (specific function definition in TIP Sections 9 - 12)
0	New Position
1	New Page
2	New Physical Line
3	New Logical Line
4	No space
5	No Operation
6,7,14,15	Invalid
8 - 13	Same as 0-5 respectively with time fill

Time Stamp

The time stamp field contains the time the last character was placed in the PRUB for batch blocks if the time stamp system option was selected. Interactive blocks do not use the time stamp field, and its contents are undefined. The time stamp is used for performance analysis only.

Level Number

The level number field is used to contain the file level number received on the EOR card of batch data input. Interactive blocks do not use the level number field and its contents are undefined.

3.1.4.2.1 Interactive Data

Upline and downline interactive data will be transferred between the host and the NPU over the coupler channel in 7 bit Internal ASCII code set. The NPU translates characters between the code set of the terminal and Internal 7 bit ASCII. All active terminals will have at least one interactive connection configured.

Interactive streams are treated independently without concern for interference with batch streams and the TIP will resolve contention between interactive and batch streams. Interactive streams are always open and do not require commands to stop, start or notify the host when a connection is stopped under normal conditions. Only abnormal interactive stoppages and resumes will be reported to the host for status updating.

Interactive input will start after the first output to the device and will continue until the terminal or device fails, or the line/terminal is deleted. Interactive input will be suspended when pre-empted by batch input or output to the same terminal, if contention exists, but will automatically resume whenever possible.

Upline interactive data blocks generally contain one line of input from the terminal and are normally followed by one line of output from the host. The TTY support does, however, provide modes of operation (i.e., block mode and paper tape) that allow multiple inputs from the terminal before output.

Interactive input data is terminated by special characters, time outs and number of input characters, depending on the TIP and mode of operation. The input data may be either a BLK or a MSG, depending on the terminator. Reference TIP Sections 4 thru 12 for this information.

Downline interactive data blocks may be terminated by any character. Once the NPU has started output to a device in the interactive mode BLK blocks are delivered without allowing input until a MSG block has been delivered.

The host defines the desired interactive Carriage Control by specifying a logical Carriage Control function in the Data Block Clarifier DBC field of each interactive output block.

The TIP must translate the DBC codes to an equivalent function for output to the interactive device. In cases where there is not an equivalent function for the output device, the DBC code is either ignored or treated as a new line, depending on the terminal characteristics.

The DBC codes and the equivalent logical function are shown in Table 3-2. Refer to the appropriate TIP section to determine the specific carriage control action taken by each TIP.

3.1.4.2.2 Batch Data

Upline and downline batch data (block types BLK and MSG) will be transferred between the host and the 2550 over the coupler channel in the Physical Record Unit Block (PRUB) format. The PRUB is formatted to be directly compatable with a CYBER Physical (disc) Record Unit (PRU) with the network block header appended to the front. Each data character is 8 bits as stored in the NPU memory or transferred across the CYBER coupler interface. Data characters will be in either 6-bit display code (stored right justified in each 8-bit character) for non-transparent modes and the code of the terminal device for transparent data modes. Reference Sections 9-12 for more information on transparent and non-transparent modes.

The PRUB may contain up to one or two PRUs (either 640 character or 1280 characters maximum), depending on a system option set for both Intercom and CCP.

Each PRUB block will be terminated and forwarded to its destination when it contains the maximum number of characters or an End of Record (EOR), or End of Information (EOI) condition is detected. Only significant data within the PRUB is transferred across the connection.

PRUBs containing EOR or EOI must be less than the maximum PRUB size. Therefore, a record or file ending on exactly a 640 or 1280 character boundry will cause an additional PRUB to be generated which contains no data characters and is marked with either EOR or EOI in the header. This applies to both upline and downline blocks.

The Block Type BT field within the network header will specify MSG (Code 02) if the PRUB contains EOR, EOI or BANNER. The BT field will specify BLK (Code 01) for all other types of PRUB blocks.

Only block types BLK and MSG that travel on a batch connection will be marked as PRUB blocks in the DBC field (bit 7 = 1). All other blocks will be marked interactive (bit 7 = 0).

3.1.4.2.2.1 Non-Transparent Data

Non-transparent data within the PRUB is assumed to be card data for upline traffic and either print or punch data for downline traffic.

Each card input will have trailing blanks suppressed and the "end of card" signified by at least two binary zero characters on a modulo 10 character boundry. That is, the NPU will insert from two to eleven zeros following the last non-blank character on a card to make the total number of characters and zeros an even multiple of 10.

Downline punch or print data will signify the end of each card or print line with an 8-bit character of all ones (HEX FF), and the FF may optionally be preceeded by one binary zero character. Zero pad characters normally used to specify end of card or line within the PRU will not be transferred downline to the NPU. The first character of each print line is treated as a carriage control character using standard Intercom conventions.

The EOR or EOI conditions detected are listed below. The EOR or EOI card itself is discarded and the appropriate bits set in DBC field of the PRUB header.

EOR - 7/8/9 punch in column 1

EOI - 6/7/8/9 punch in column 1 for Mode 4

- /*EOI in columns 1-5 for BSC and HASP

Bisync ETX received from the terminal. ETX will be received when an ETX is punched in the last column of the last card or by entering the last card with the EOF switch depressed. This will be the only method of determining EOI when in the transparent mode.

A one or two digit level number may optionally be specified in columns 2 and 3 of the EOR card. This level number, if present, is converted to an 8-bit binary value and transferred upline to the host in the level number field of the header of any PRUB containing EOR. The level number field will be zero if not present in the EOR card.

Downline PRUBs containing EOR or EOI that are directed to a punch device will cause an EOR card (7/8/9 punch) or EOI card (/*EOI) to be punched. The level number contained in the header of an EOR block will also be punched in columns 2 and 3 of the EOR card.

3.1.4.2.2.2 Transparent Data

Transparent data within the PRUB provides a method of transferring batch data between the terminal and the host files without modification by the CCP software. The block header is identical to that defined for non-transparent data.

Input data received in the transparent mode is stored in the PRUB without code translation, data expansion or blank suppression. Transparent PRUB blocks are terminated and forwarded to the host when one half the number of characters specified for the PRUB size is reached (e.g., 320, 640) or the end of information is reached. EOR and EOI cards are not recognized in the transparent data mode; therefore, end of information is detected by ETX received from the terminal (Reference TIP Sections 9 and 11).

Transparent input data may be specified by three methods -
1) Optional parameter on the Intercom command Read File Name,
2) "TR" in columns 79 and 80 of the Job Cand and,
3) "TR" in columns 79 and 80 of the EOR Cand.

The full 8 bits of each data character is written to disc PRUs for transparent upline PRUB blocks.

Output files may also be specified as transparent. The host marks the header of each PRUB as "transparent" for
files. Transparent PRUBs are output to the terminal without modification of the data characters (i.e., no code translation, data compression, carriage control line folding, etc). Data characters are, however, blocked into the maximum size transmission blocks specified for the terminal/device receiving the data. Transmission blocks will also be terminated at the last data character of a PRUB block that is marked as an EOR or EOI block.

Table 3-1
Command Blocks Used For CCI 3.0
on Non-Zero Connections

Type	$Param_1$ Primary Function	PFC Code	$Param_2$ Secondary Functions	SFC Code	Comments
RS	Start Input	1	Initial, non-xpar Initial, xparent Resume	0 1 2	Start Input from batch device
RS	STOP Input	2	Terminate Suspend	0 1	Discard data Stop polling Stop polling and wait
FD	Input Stopped	3	End Break 1 2 ⋮	0 1 2 ⋮	Normal End Reason for Break defined by TIP No response
FD	Input Started	4	Restart after Stop	0	Interactive Resume after Break (status only)
RS	Output Stopped	5	Break 1 2 ⋮	0 1 2 3 4 5	Reason for Break defined by TIP
RS	Output Started	6	Restart after stop	0	Status only
FD	Restart Output	7	-	0	Resume output stream
FD	Stop Output	8	-	0	Discard data and terminate
RS	BSN Error	9	NPU has detected BSN out of order	0	Diagnostic purpose only

RS = Reverse Supervision FD = Forward Data

3.1.4.3 Common Host/TIP Interface

This section describes the common interface between the host and the NPU Terminal Interface Program (TIP) used to control terminals and the data flow. The block types and formats described previously are used for this interface and further clarification of CMD blocks and the block protocol sequences are described.

The host/TIP block protocol flow is designed to be common for all terminal types thus eliminating the host dependent code for each type supported. The host software will, however, have special logic for configuration of the different terminals and will be aware of the general data flow capabilities of each type, i.e., interleaved input/output, multiple input/output streams, HDX input/output streams. In addition, the interactive host/TIP interface is different for TTY terminals and all other types in order to utilize existing code for the initial implementation. The TIP sections further describe the unique handling of each terminal's operation, flow control, contention resolution, etc.

Some of the general rules used for the common interface in this implementation are listed below:

- Each block (CMD, BACK, BLK, or MSG) will contain a Block Serial Number (BSN) in bits 4, 5 and 6 of the DBC field. The BSN field will always be zero on the Service Message channel (connection number zero).

- BSN's will be assigned sequentially by the transmitting process in bits 4-6 of the Block Type field of the block header. This will be a module 8 count beginning with zero when the connection is established and continuing sequentially until the connection is dissolved. No correlation between upline and downline BSN's is assumed.

- The receiver will check BSN's on each block received against the "next expected" serial number and if correct the next expected BSN count will be updated. Detection of an out-of-sequence number by the NPU will result in an upline "stop command" to the host; PFC=0, SFC=0.

  No specific recovery logic will be included to restart a connection when the host receives an out-of-sequence upline block or the upline command from the NPU signifying receipt of an out-of-sequence block.

- The BSN's are included for program debugging purposes only at this time although recovery logic could be added at a later time.

- Stream control is generally different depending on the characteristics of the terminal and therefore is described in the TIP sections 9-12. However, all TIPs start batch input streams by a command from the host and start output streams by the first output block. Interactive streams are generally always open, therefore, do not require commands to start or stop.

- All intercom commands input at the terminal are passed to the host and may result in a downline CMD to control the TIP.

. The TIP must notify the host of any terminal condition requiring operator intervention.

. On batch or interactive connections Upline CMD's from the NPU to the host must be suspended until the host is expecting a block on that connection i.e. has BACK'ed the previous upline block.

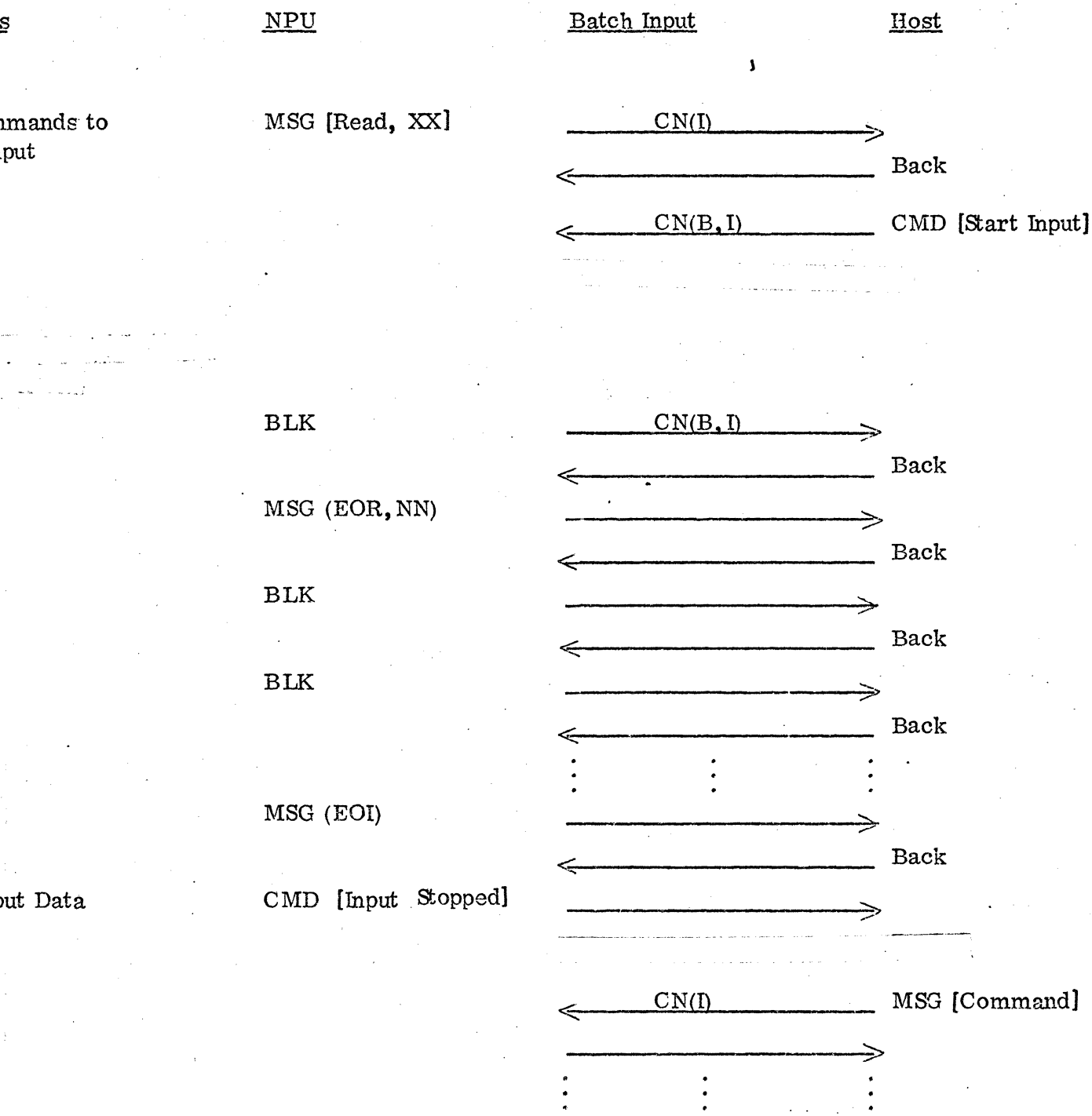
. On batch output connections Upline CMD's from the NPU to the host may not be sent unless the host is expecting a "BACK". If the upline CMD is "STREAM STOPPED" the "BACK" that is due is then suspended until a Restart Stream CMD is received from the host, the BACK is discarded.

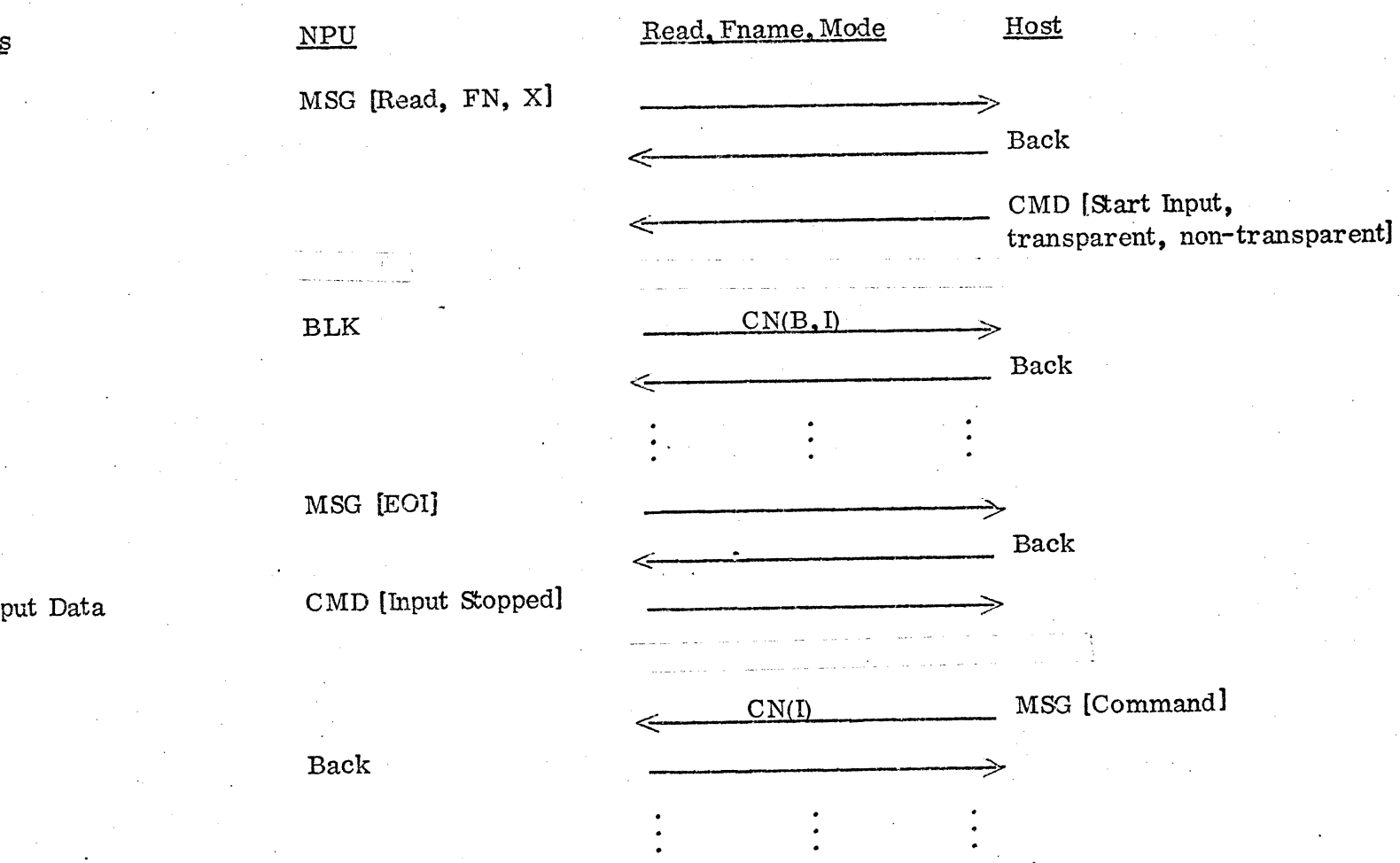
3.1.4.3.1 Block Protocol Flow

The following diagrams give examples of the block protocol sequences between the host and NPU for some typical normal and abnormal cases.

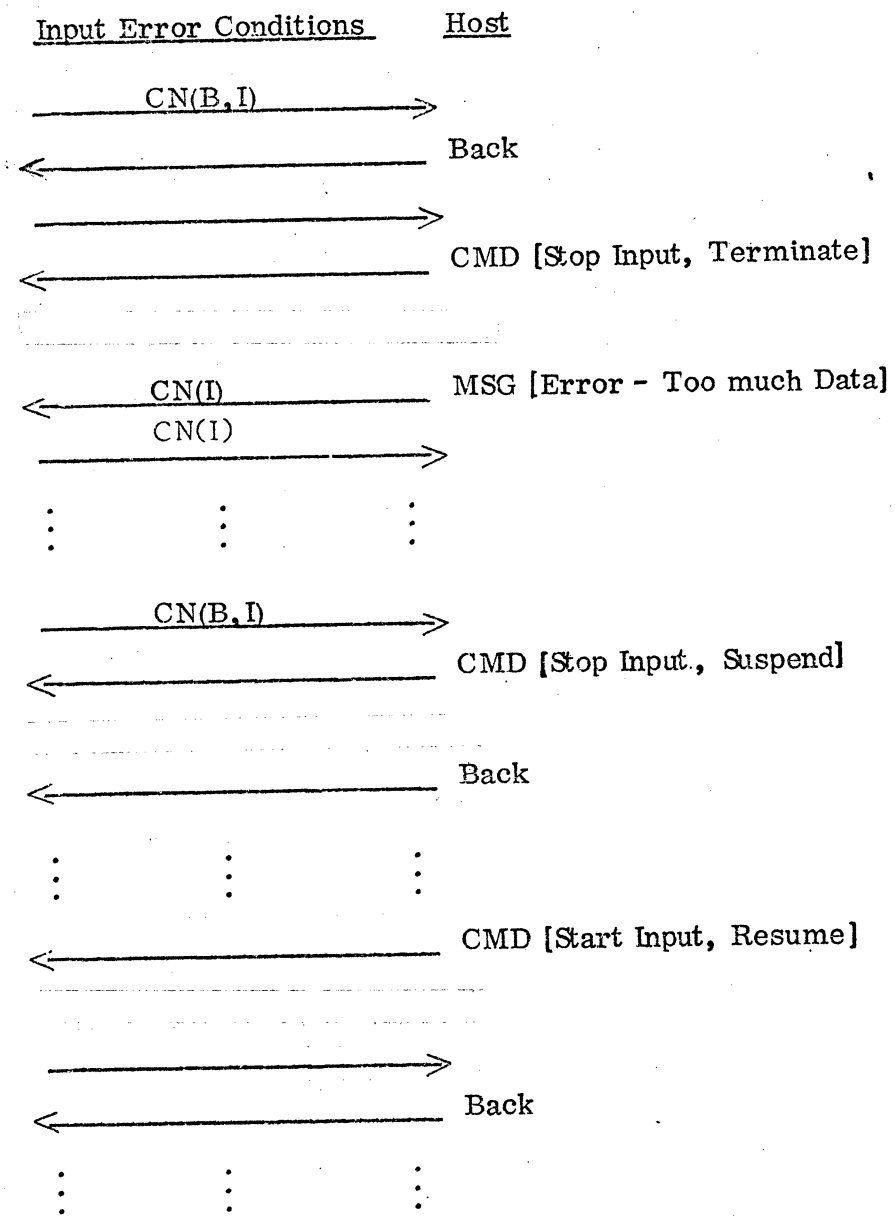
The notation used in the diagrams is as follows:

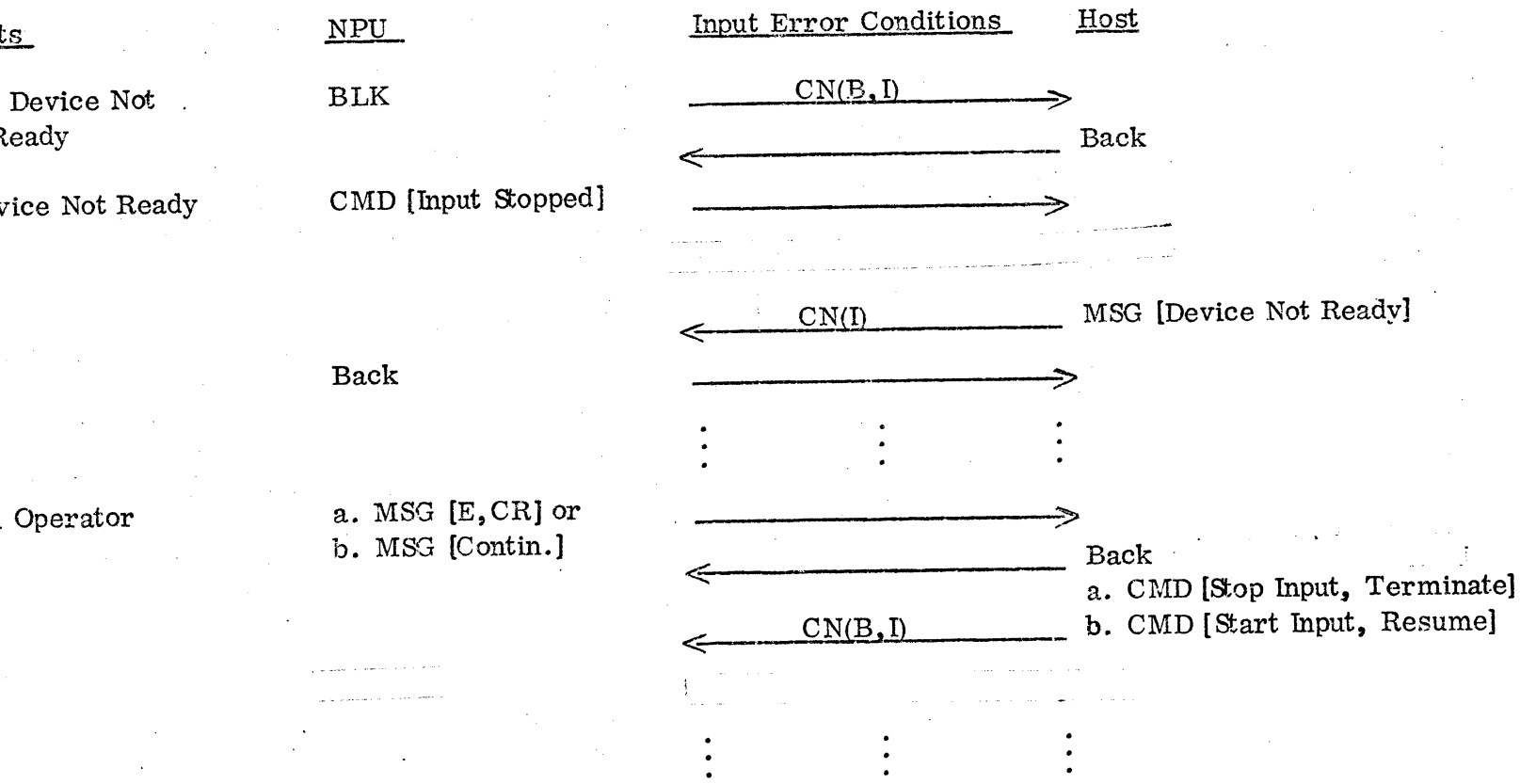
CN	-	Connection Number
SM	-	Service Message (See Appendix A)
I	-	Interactive
B,I	-	Batch Input
B,O	-	Batch Output
CMD	-	Block Protocol Command (See Table 8-1)
Command	-	INTERCOM-user command

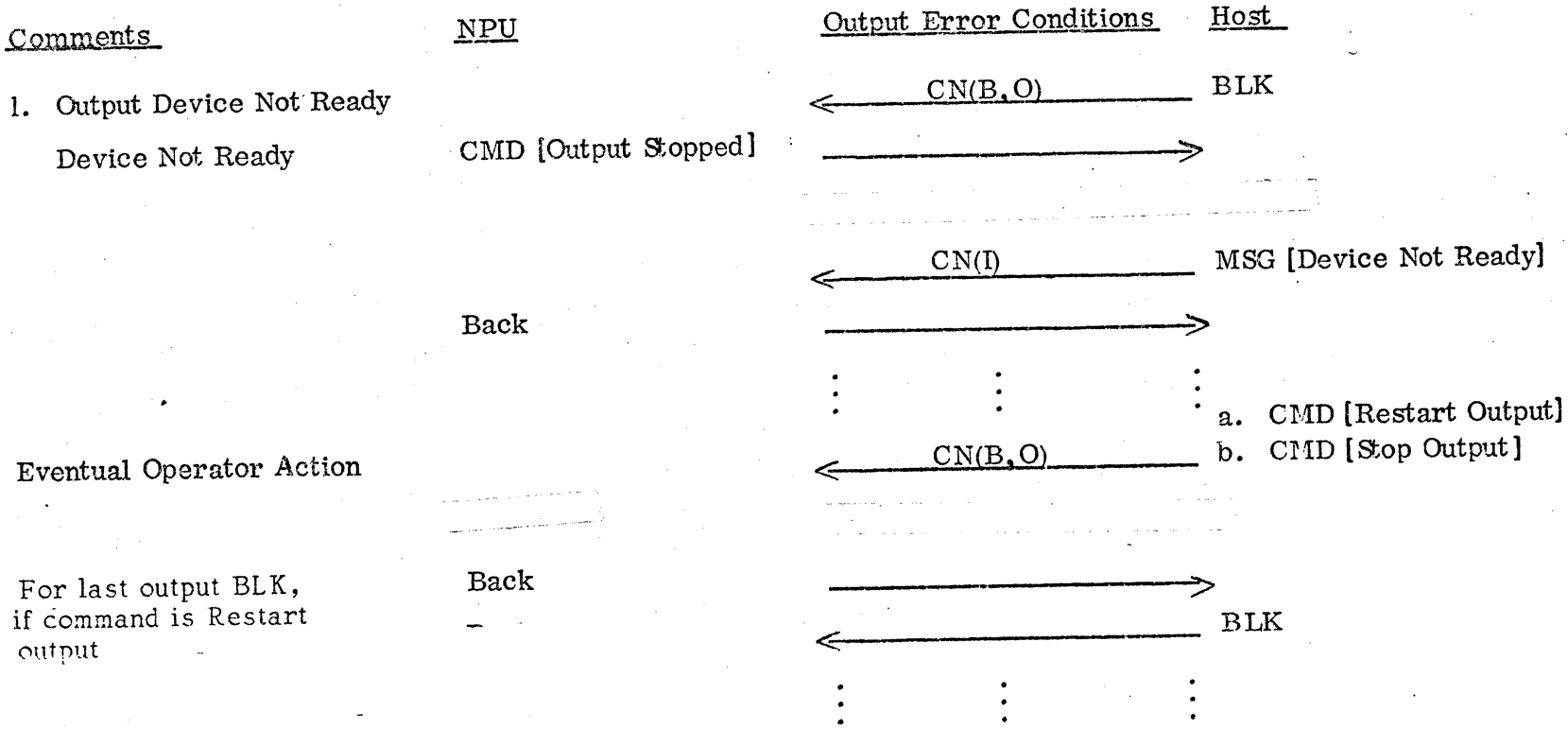

Comments
NPU
Batch Input
Host
User Commands to Initiate Input
MSG [Read, XX]
CN(I)
Back
CN(B,I)
CMD [Start Input]
BLK
CN(B,I)
Back
MSG (EOR,NN)
Back
BLK
Back
BLK
Back
MSG (EOI)
Back
End of Input Data
CMD [Input Stopped]
CN(I)
MSG [Command]

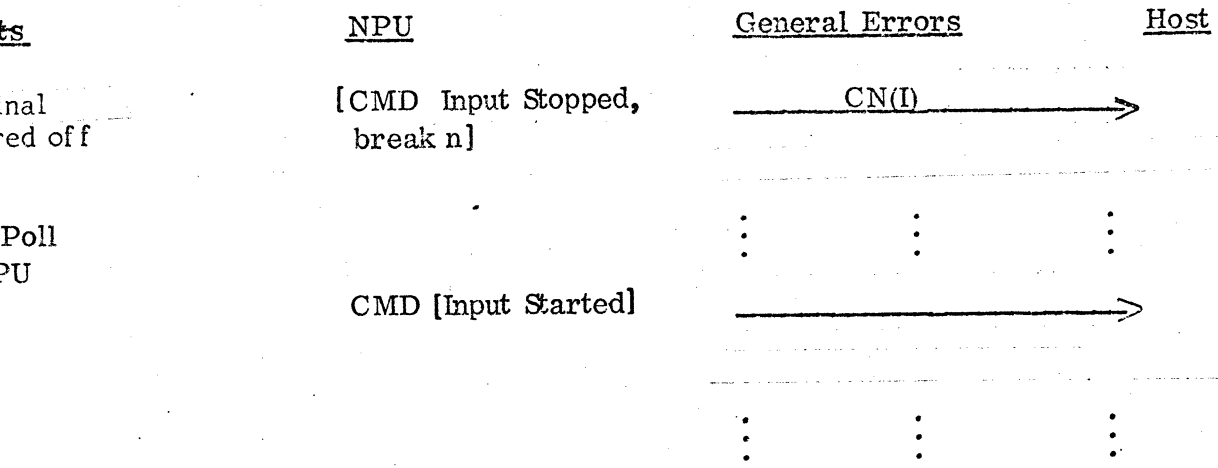

Comments
NPU
Read, Fname, Mode
Host
MSG [Read, FN, X]
Back
CMD [Start Input, transparent, non-transparent]
BLK
CN(B,I)
Back
MSG [EOI]
Back
End of Input Data
CMD [Input Stopped]
CN(I)
MSG [Command]
Back

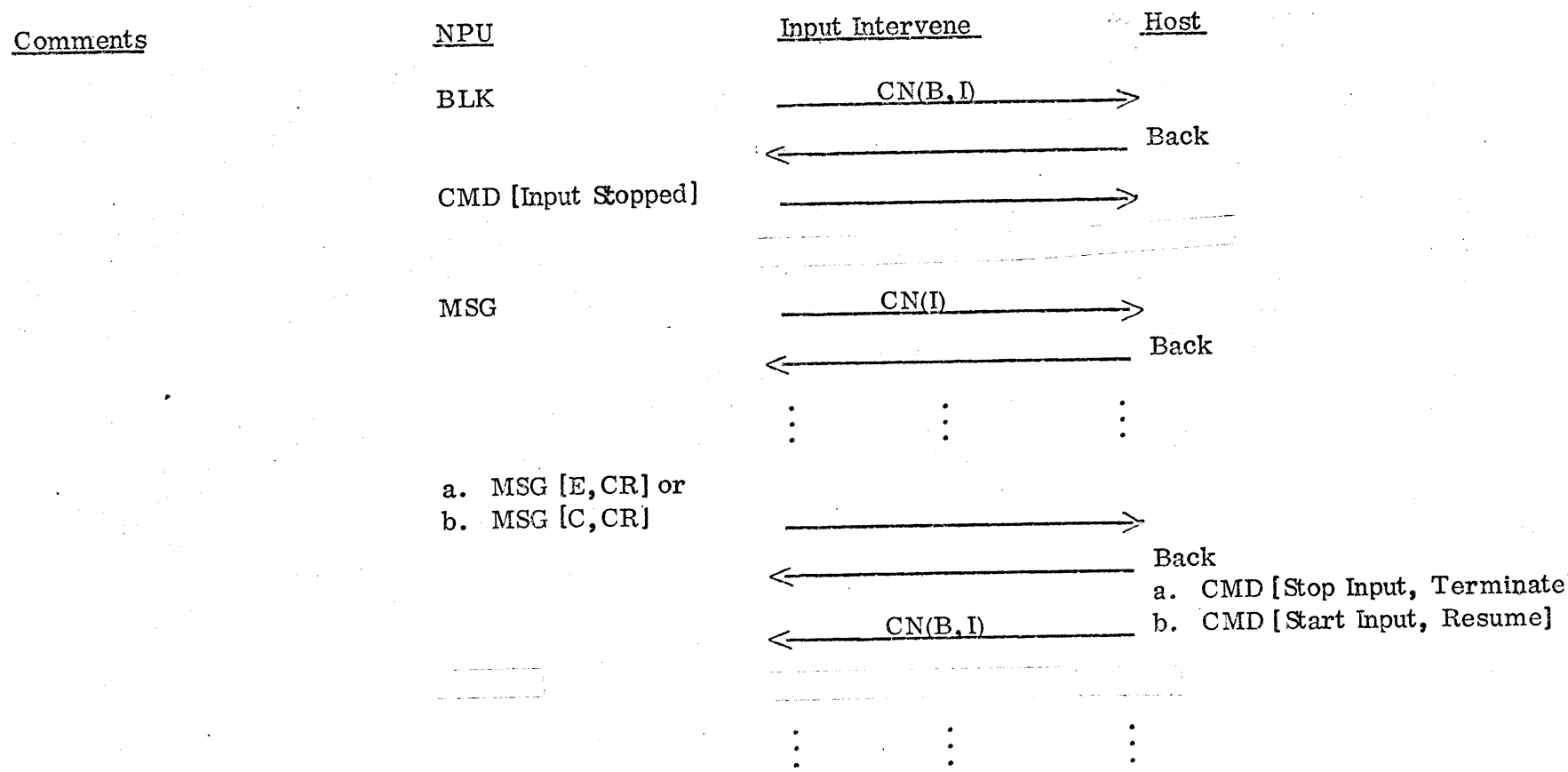
Input Error Conditions

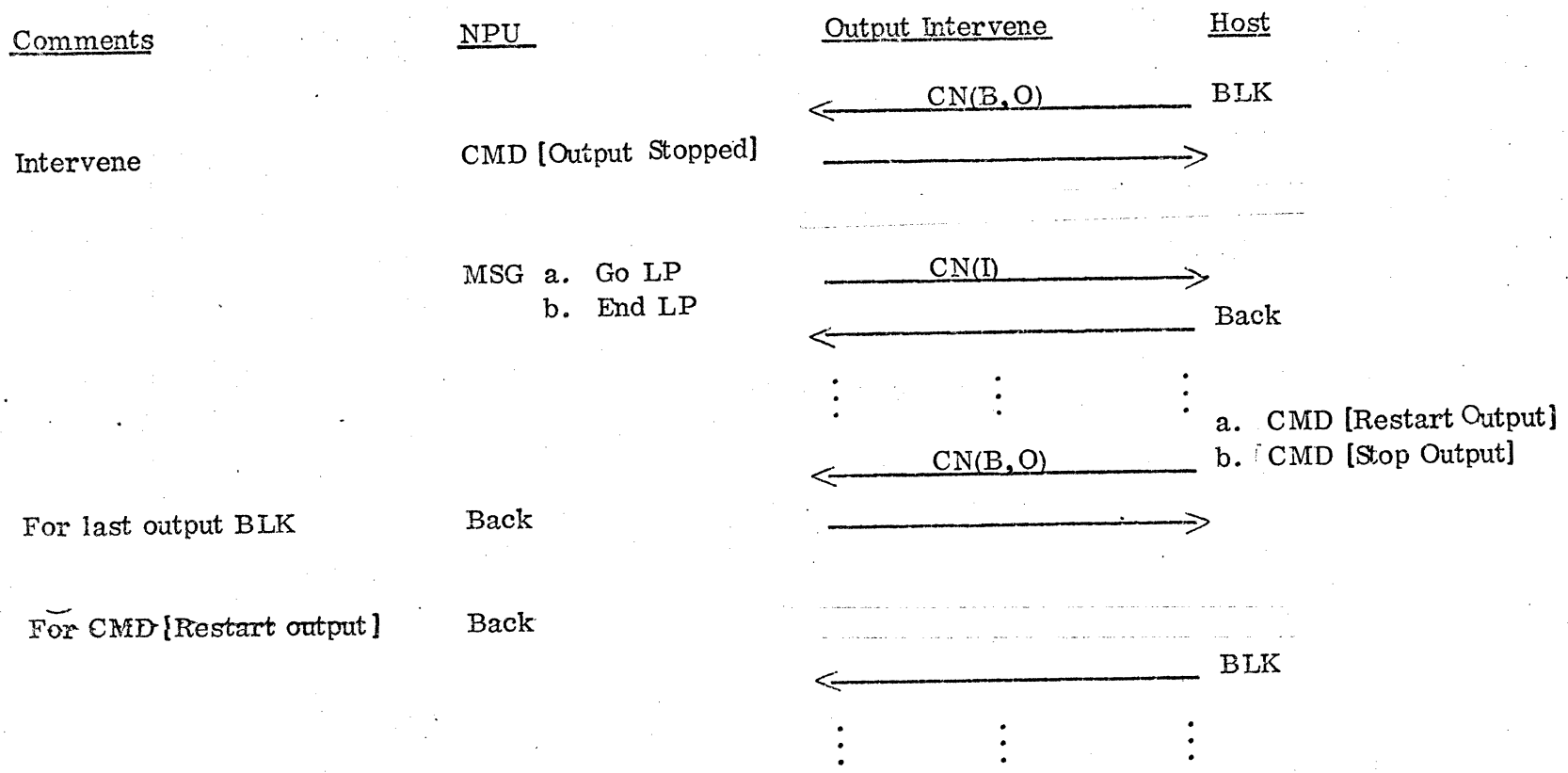


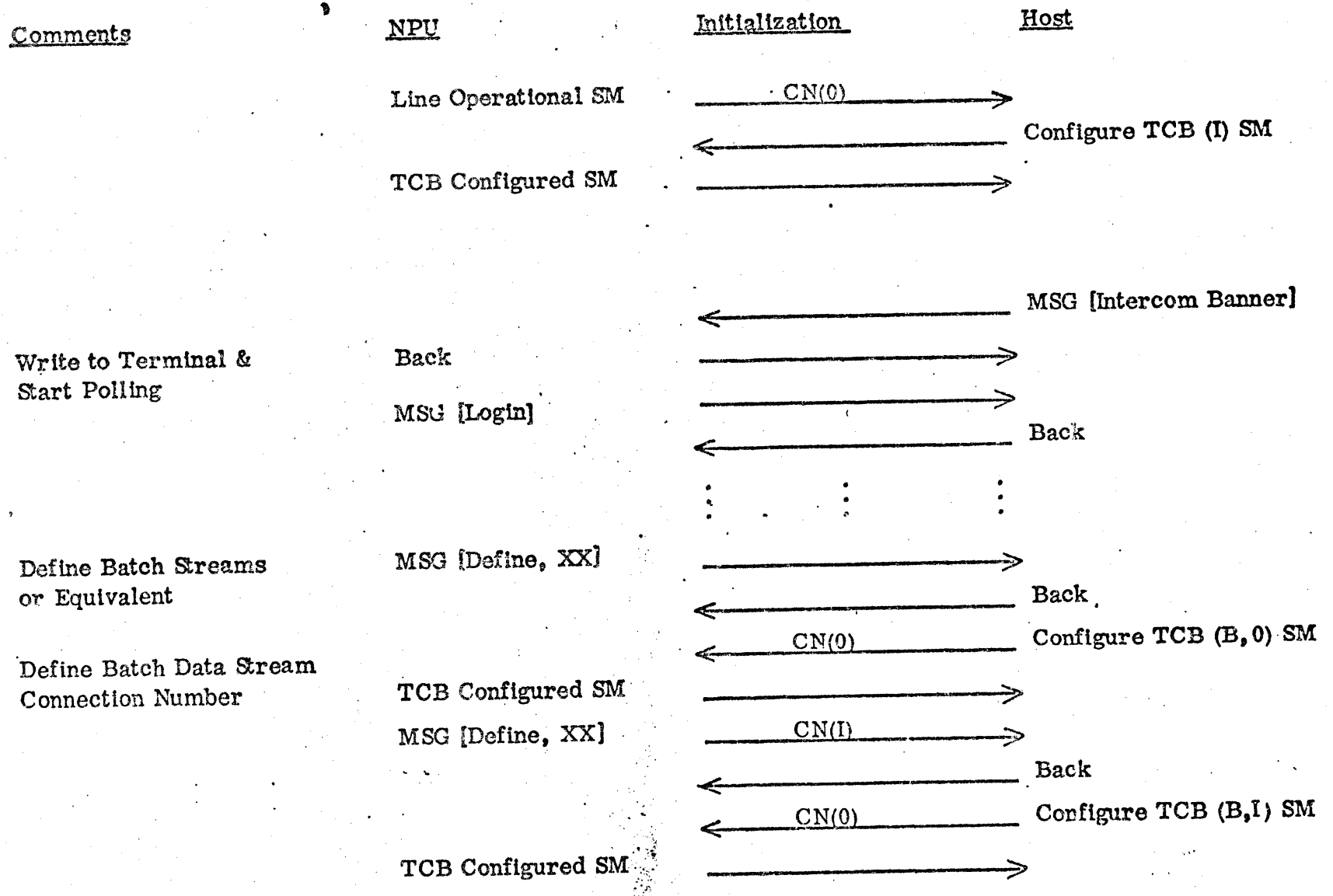

Comments
NPU
Input Error Conditions
Host
3. Input Device Not not Ready
BLK
CN(B,I)
Back
Input Device Not Ready
CMD [Input Stopped]
CN(I)
MSG [Device Not Ready]
Back
Eventual Operator Action
a. MSG [E,CR] or
b. MSG [Contin.]
Back
a. CMD [Stop Input, Terminate]
b. CMD [Start Input, Resume]
CN(B,I)

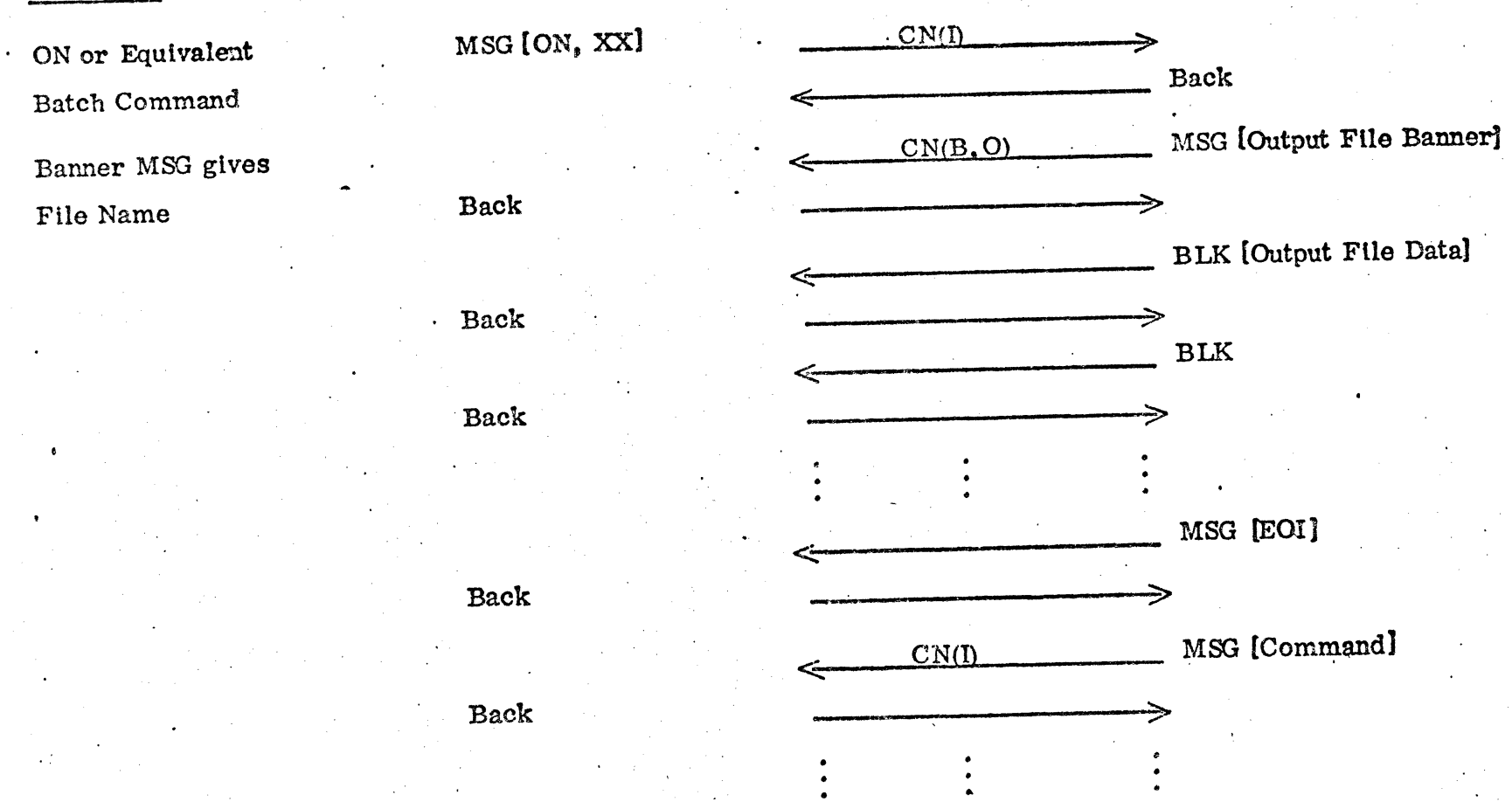

Comments
NPU
Output Error Conditions
Host
1. Output Device Not Ready
CN(B,O)
BLK
Device Not Ready
CMD [Output Stopped]
CN(I)
MSG [Device Not Ready]
Back
a. CMD [Restart Output]
Eventual Operator Action
CN(B,O)
b. CMD [Stop Output]
For last output BLK, if command is Restart output
Back
BLK


Comments
NPU
General Errors
Host
1. Terminal powered off
[CMD Input Stopped, break n]
CN(I)
Delayed Poll From NPU
CMD [Input Started]


Comments
NPU
Input Intervene
Host
BLK
CN(B,I)
Back
CMD [Input Stopped]
MSG
CN(I)
Back
a. MSG [E,CR] or
b. MSG [C,CR]
Back
a. CMD [Stop Input, Terminate]
b. CMD [Start Input, Resume]
CN(B,I)


Comments
NPU
Output Intervene
Host
CN(B,O)
BLK
Intervene
CMD [Output Stopped]
MSG a. Go LP
b. End LP
CN(I)
Back
a. CMD [Restart Output]
b. CMD [Stop Output]
CN(B,O)
For last output BLK
Back
For CMD [Restart output]
Back
BLK


Comments
NPU
Initialization
Host
Line Operational SM
CN(0)
Configure TCB (I) SM
TCB Configured SM
MSG [Intercom Banner]
Write to Terminal &
Start Polling
Back
MSG [Login]
Back
Define Batch Streams
or Equivalent
MSG [Define, XX]
Back
CN(0)
Configure TCB (B,0) SM
Define Batch Data Stream
Connection Number
TCB Configured SM
MSG [Define, XX]
CN(I)
Back
CN(0)
Configure TCB (B,I) SM
TCB Configured SM


Comments
NPU
Batch Output
Host
ON or Equivalent Batch Command
MSG [ON, XX]
CN(I)
Back
Banner MSG gives File Name
CN(B,O)
MSG [Output File Banner]
Back
BLK [Output File Data]
Back
BLK
Back
MSG [EOI]
Back
CN(I)
MSG [Command]
Back

4.0 NPU INITIALIZATION

This section describes the features and operator actions necessary to make the NPU fully operational. This includes loading, dumping and initialization.

The NPU system is loaded and initialized by the host computer system. NPU operating programs and tables are formatted into a load file that is resident on the mass storage of the host. To start NPU operation, the load files, one containing writable micromemory resident and a second containing the main memory-resident programs must successfully be transferred (loaded) into the NPU. If the downline load is unsuccessful, the host initiates and receives a dump of the NPU memory, micromemory and file registers. The initiation of another downline load attempt is under control of the host.

The NPU is loaded and dumped via the CYBER Coupler under the control of the Cyber's PPU. NPU software is not involved in this load/dump process.

Failure of an NPU is always detected by the host PPU channel coupler driver. Upon detecting a failure condition, the NPU will stop servicing the channel coupler. The PPU is then able to detect the NPU failure by a timeout of the protocol over the channel coupler.

4.1 LOAD/DUMP Phases

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

The 2550 baud 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 2550 dump is three-phase.
First the contents of mainmemory is dumped. Secondly, a small program is loaded into the 2550 and executed to transfer the file registers and a checksum of RAM to an area of main memory. Thirdly, this area of main memory 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 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 format of NPU dump output is described in the NPU dump analyzer specification.

4.1 LOAD/DUMP Phases (cont.)

* Although the loading/dumping of the 2552 is described here, its testing will not be performed as part of the Mobil/Suntech project.

4.2 NPU Loading

The NPU is connected to the host via a hardware channel and is loaded directly over a channel coupler by a PPU.

During the downline load, micromemory must be loaded prior to main memory load.

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

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.

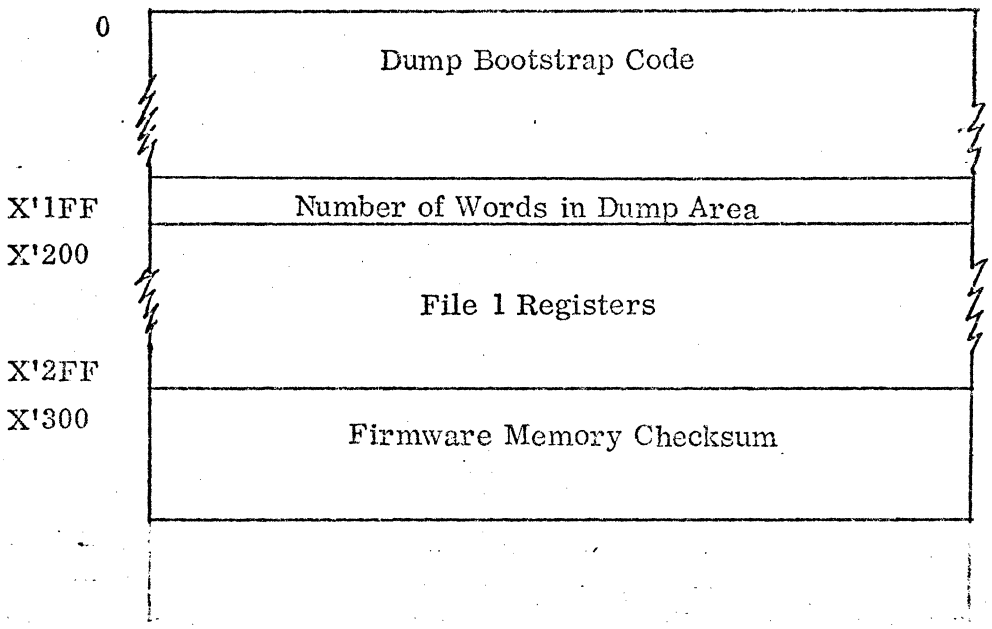
See Section 7.1.1.3 for additional information on the NPU load/dump sequence.

4.3 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 the 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 4.1 for a 2550 and as shown in Figure 4.2 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 4.1 Dump Bootstrap NPU Memory Format (2550)



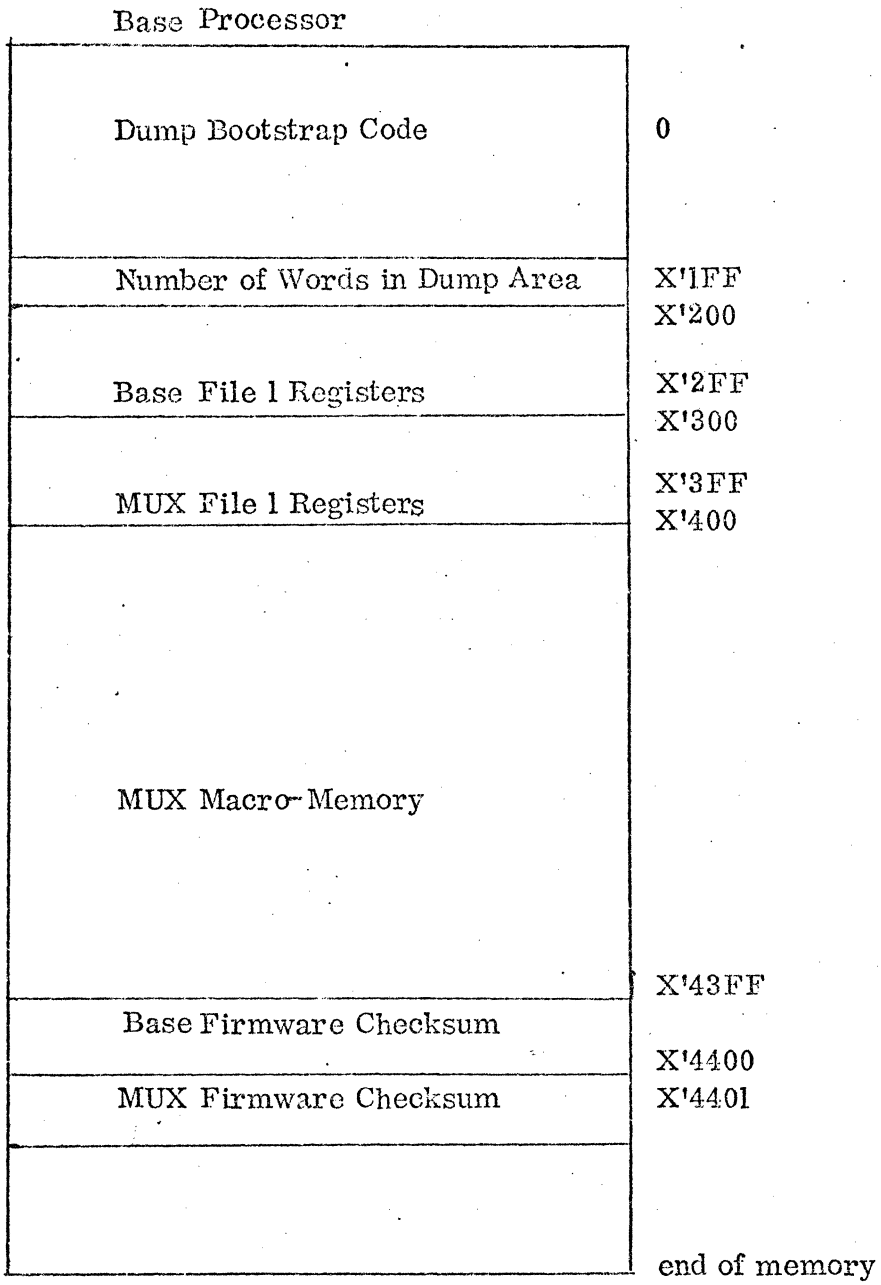


Figure 4.2 Upline Dump File Format (2552)

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, Host Unavailable"
Recovery from Host Unavailable terminal still connected				NPU sends "Input Resumed" Resume polling
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		Stop polling controlled terminals
All Ports Active	X	Disconnect existing dial-up lines.		Provided by Telephone Co. at end of rotary.

TABLE 4-6

Modem Class

The modem class denotes the maximum speed at which the modem is capable of operating and it must be specified according to the following criteria.

Test Mode	CLA Type	Maximum Modem Speed	Modem Class (ZZ)	Modems *
Internal and External Loopback	ALL	Not Applicable	0	NONE
Modem Loopback	2560-1 2560-2 2560-3 2563-1	Not Applicable	1	201B, 201A, 201C, 201D, 208A, 208B 358-2
	2561-1	100	2	
		110	3	
		120	4	
		134.5	5	
		150	6	
		300	7	103 series, 113A, 113B, VA3405-A-G
		600	8	VA 3405 A thru G
		800	9	
		1050	A	
		1200	B	VA 3405A thru G
		1600	D	
		2400	F	
		4800	10	
		9600	12	358-1

* The modems listed constitute only a small fraction of all possible modems offered by a variety of manufacturers.

5.0 NETWORK OPERATION

5.1 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.

5.1.1 Initiating Logical Connections

After downline load of an NPU, the NPU will inform the host that it is active by sending an NPU initialized SM. The host will then configure the lines. Whenever a line is reported to the host as operational, the host will configure and connect each terminal on the line.

A configure or reconfigure terminal SM is issued for the station to be connected 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. The host is informed of the successful completion of the configuration by a normal response.

5.1.2 Changing Logical Connections

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

5.1.3 Deleting Logical Connections

A logical connection will be deleted when a line or process supporting the connection becomes inoperative at either end of the connection or the logical link is disabled or fails. A logical connection may also be explicitly deleted by means of a Reconfigure TCB SM setting the connection number to zero.

5.2 Lines

The dynamic configuration of a line and terminal is performed by the exchange of SMs between INTERCOM in the CYBER host and the Service Module in the NPU(s). An overview of the process and the SMs involved is given by Figure 5.2-1. 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.

5.2.1 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 TTY, Mode 4, 2780/3780, and HASP.

5.2.1.1 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 19.2K bps determined by modem - provisions for external clock source
- Full RS232C/CCITT V24 interface
- Data transfer overrun/underrun detection

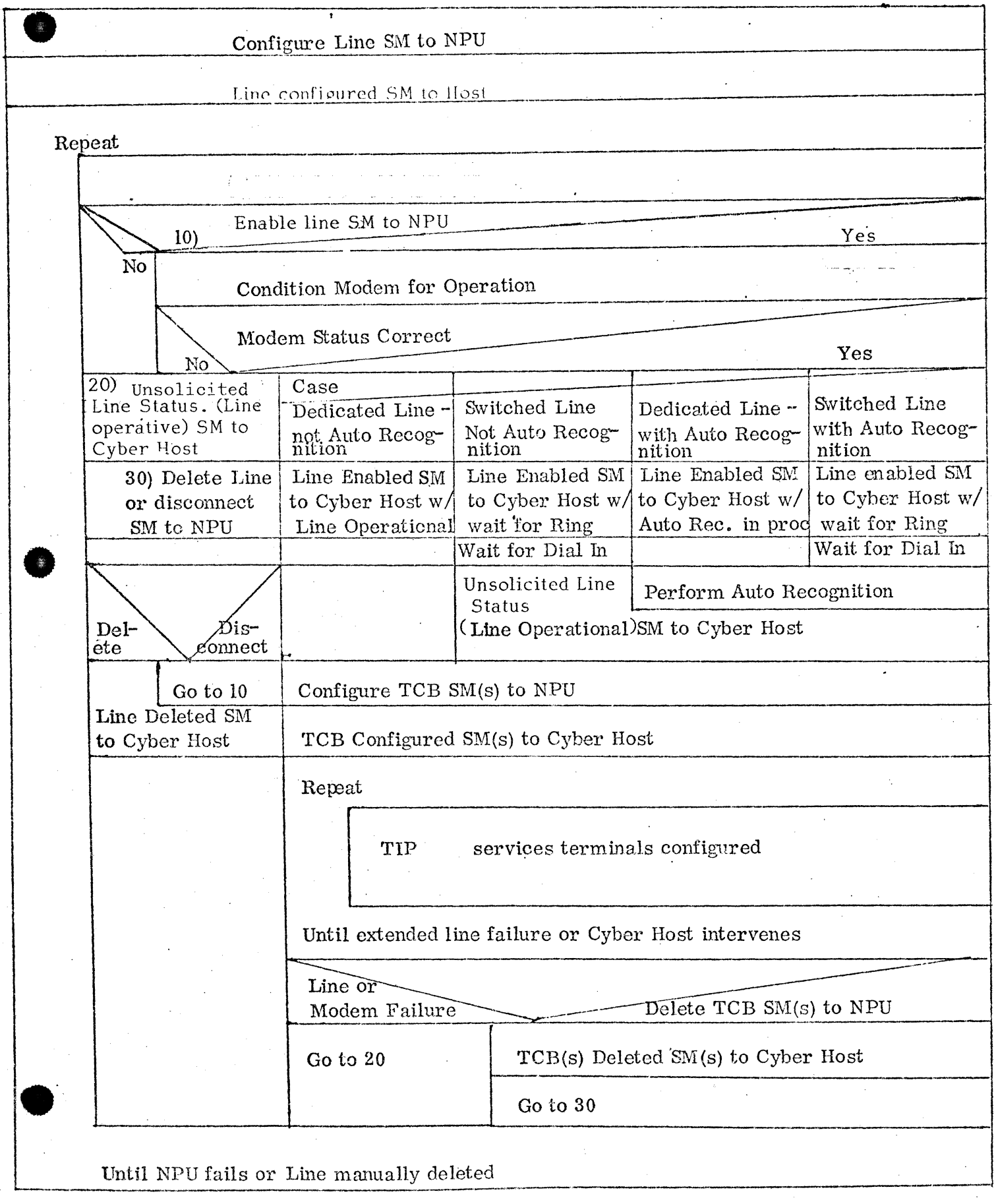


Figure 5.2-1. Configuration Process

5.2.1.2 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.

5.2.1.3 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.

5.2.1.4 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.

5.2.1.5 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.

5.2.2 Configure Line SM

Parameters

Port

Subport

Line Type

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

Field Name $_n$
Field Value $_n$
} One or more field name/field value pairs

Purpose

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

Stimulus

The Host required to configure lines after the NPU is loaded.

5.2.2 Configure Line SM (Con't)

Action

All Configure SMs 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.

Response to enable 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 as a normal response; otherwise, line inoperative is reported.

Response to enable 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 Status (with RC=Operational) SM when a dial-in connection occurs. At this time the NPU returns Data TerminalReady (DTR) to answer the call. If the host is not available when the Ring indicator occurs, 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 reenable at the earliest opportunity. If a response is obtained, the Current Terminal Type (CTT) is set accordingly, and Line Operational is reported. CTT contains the results of auto-recognition.

5.2.3 Delete Line SM

Parameters

Port

Subport

Purpose

To change line status to LCB Not Configured.

Stimulus

Host desires to disable the line.

Action

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

5.3 Terminals

5.3.1 Terminal Identification

Terminals are identified in service messages by specifying the line, the hardware address, Device Type (DT), Terminal Class (TC). The line identification is given in Section 5.2.1.

The table below shows the valid CA and TA values for each terminal.

	CA	TA
Mode 4A	X'70-X'7F	X-60
Mode 4C	X'70-X'7F	X'61-X'6F
TTY	0	0
HASP	0	0-7(1)
BSC	0	0-1(2)

(1) Equal to the stream ID of the device. The Interactive console must be 1, Card Reader(s) 1...7, Printer(s) 1...7, Punch(es) 1...7.
(2) Punch only, all other devices are zero.

5.3.2 Configure Terminal SM

Parameters

Port

Subport

Cluster Address

Terminal Address

Terminal Class and Device Type

Field Name$_n$
Field Value$_n$
} One or more field name/field value pairs

Purpose

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

Stimulus

Line Operational SM is received by the host and the processing of the system tables reveals terminals for lines which need to be configured.

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.

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 the host 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 the host. Exception responses are described in Appendix A.

5.3.2 Configure Terminal SM (cont.)

Action (cont.)

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

5.3.3 Reconfigure Terminal SM

Parameters

Port

Subport

Cluster address

Terminal address

Terminal Class and Device Type (current)

Field Name$_n$
Field Value$_n$ } One or more field name/field value pairs

Purpose

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

Stimulus

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

Action

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

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

5.3.4 Delete Terminal SM

Parameters

Port

Subport

Cluster Address

Terminal Address

Terminal Class and Device Type

Purpose

To delete a TCB which is no longer required.

Stimulus

The operator requests that a terminal be deleted. Terminal failure.

Action

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

5.4 Line States

Figure 5.2-1 shows the sequencing of upline and downline SM's.

5.4.1 Intentionally blank.

TABLE 5.4-1

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 But no response to input until Host Up	
Host Unavailable terminal connected				NPU sends "Input Stopped"
Recovery from Host Unavailable, terminal still connected				NPU sends "Repeat Last Input" Resume polling
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 Disconnect existing dial-up lines.		
All Ports Active	X			Provided by Telephone Co. at end of rotary.

5.4.2 Disconnect Line SM

Parameters

Port

Subport

Purpose

To combine the action of a Disable/Enable pair in one SM. Specifically designed for the switched line case. A Disconnect Line SM accomplishes exactly the same functions as a Disable Line SM immediately followed by an Enable 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 Disable followed by an Enable. The response to this SM is identical to the Line Status Response SM (see Appendix A) except that the function codes differ.

5.4.3 Line Status Request SM

Parameters

Port, Subport } Optional

Purpose

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

Stimulus

Send when the host 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 or more responses in this case for each configured line.

5.4.4 Line Status Response SM

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

5.4.4.1 Line Operational

Parameters

Port

Subport

Response code = 0

Line type

Configuration state

Number of terminals configured

Terminal type (if unsolicited status SM)

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 , an unsolicited Line Status SM sent to the host after Ring indicator has been detected and auto recognition is complete (if configured for auto terminal).

Action

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

5.4.4.2 Line Inoperative

Parameters

Port

Subport

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

Data Carrier Detect (FDX Constant Carrier): Once a line in 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.

5.4.4.2 Line Inoperative (cont.)

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.

5.4.5 Terminal Status Request SM

Parameters

Port

Subport

Purpose

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

Stimulus

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

Action

The NPU sends a Terminal Status Response SM back to the sender for all terminals on the specified line.

5.4.6 Terminal Status Response SM

Parameters

Port

Subport

Cluster address

Terminal address

Response code

Terminal Class and Device Type

Destination node

Source node

Connection number

Total statuses reported

5.4.6 Terminal Status Response SM (cont.)

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.

Action

When terminal failure is detected, the correspondent is informed via the logical connection 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. The host may elect to attempt recovery from terminal failure, or it may terminate the associated logical connection.

5.5 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 as 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 3.

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

- NPU Input Regulation
- Station Input Discard

5.5.1 NPU Input Regulation

Regulation in the NPU consists of referencing 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.

5.5.2 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.

5.6 User Communication

5.6.1 Host Broadcast One SM

Parameters

P
SP line number

CA
TA terminal ID

Text (1-50 characters) in Interactive format.

Purpose

To allow the CYBER Host to send a message to an 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.

5.6.2 Host Broadcast All SM

Parameters

None.

Text (1 - 50 characters) in interactive compatible format.

Purpose

To allow the CYBER Host to send a message to all interactive terminal users on a logical link.

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.

5.7 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 the host. 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 Disable or Line Disconnect SM is received. The affected line and terminal statistics blocks are dumped prior to sending the Line Disable Response SM.
- a Delete Terminal SM is received. The affected terminal statistics block is dumped prior to sending the Delete Terminal Response SM.

5.7.1 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 3

Number of times at regulation Level 2

Number of times at regulation Level 1

Number of times at regulation Level 0

5.7.1 NPU Statistics SM (cont.)

Purpose

To send statistical information accumulated by the NPU to the host.

Stimulus

Sent periodically by 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.

5.7.2 Line Statistics SM

Parameters

Port number

Subport number

Blocks transmitted

Blocks received

Characters transmitted } Good blocks only

Characters received }

Purpose/Stimulus/Action

(As for NPU Statistics SM)

5.7.3 Terminal Statistics SM

Parameters

Port number

Subport number

Cluster address

Terminal address

Terminal Class and Device Type

5.7.3 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)

5.7.4 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.

5.7.4 CE Error SM (continued)

Purpose

To allow the NPU program to report detected errors to the host.

Stimulus

Occurrence of a program-detectable error.

Action

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

5.8 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.

6.0 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 inter-relationship.

6.1 CYBER Host Failure

The unavailability of the host will be detected at the NPU(s). The unavailability will be communicated to the connected terminals and further input will be inhibited.

6.2 CYBER Host Recovery

After recovery, the host will reload the NPU and reconfigure all lines. New connections must be established as terminals become "operational" (connected).

6.3 NPU Failure

The CYBER will be aware of the NPU failure.

6.4 NPU Recovery

The CYBER host driver PPU will stop servicing the NPU if any of the following halt conditions occur.

- Software failure causes deadman timer to expire.
- Hardware failure causes deadman timer to expire.
- Host operator-initiated software halt.
- Manually initiated halt as a result of operator depressing NPU Master Clear button.

In all of the above cases the host processor initiates an upline dump of NPU main memory, micromemory and file registers. The CYBER host operator is notified that the PPU driver is no longer servicing the NPU. It is the responsibility of the host operator to reinitiate servicing of the NPU by performing another downline load or (in the case of hardware malfunction) informing the local customer engineers.

6.5 Line Failure

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

6.6 Line Recovery

A line cannot recover from a failure spontaneously. It is necessary for host 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/HIP will commence to monitor for a change in status. When the line status changes to operational, this is reported to host. Host, receiving a line status change to operational, will attempt to configure the supported terminals as previously described.

6.7 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 Host for network management purposes. The correspondent to which the terminal is logically connected will be informed of the failure by the Block Protocol.

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

6.8 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 the host and the logically connected correspondent will be informed by the use of the Block Protocol.

7.0 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.

7.1 CYBER Host Interface Program

7.1.1 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 only one coupler on a channel. An NPU interfaces one or two couplers.

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.

7.1.1 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.

7.1.1.1 Registers (See Figure 7-1)

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 18-bit register, all bits of which can be written by the PPU, for the purpose of loading or dumping the NPU. The high order 10 bits (Address Register bits 17-8) 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 bits of the Address Register (bits 17-16) are actually implemented as bits 9 & 8 of the NPU Status word, which can therefore not be used for other purposes.

7.1.1.1.1 Coupler Status Register (See Table 7-1)

Table 7-1. 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.

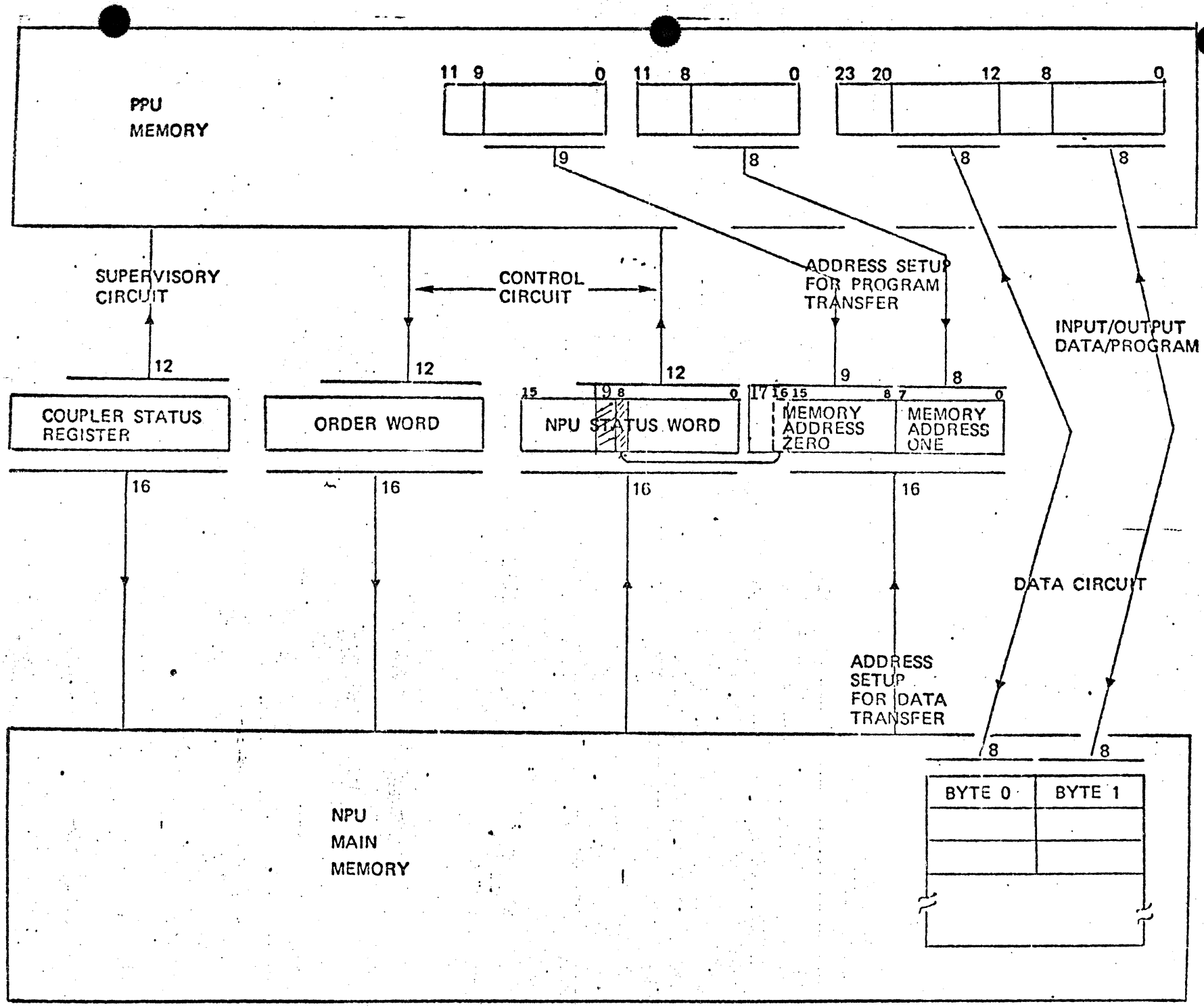
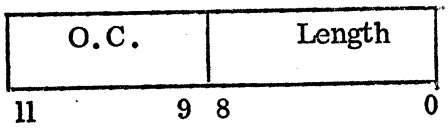


Figure 7-1. CYBER Coupler Registers

7.1.1.1.2 Order Codes

Four order codes are defined for the data transfer protocol. See Section 7.1.2 for description of the interpretation of these codes.



OC - Order Code (See Table below)

Order Code Value	Name	Regulation Level
1	Output Level 1 (Reverse Supervision & Service Messages)	1
2	Output Level 2 (High Priority Data)	2
3	Output Level 3 (Low Priority Data)	3
5	Not ready for input	

Length - In 8 byte increments, of the output block to be transferred. (May be larger than the actual length of the BLK.)

7.1.1.1.3 Status Codes

Seven status codes are defined for the data transfer protocol. See Section 7.1.2 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
12	Input available Batch (PRUB) block
13	Input available $\leq$ 256 bytes (Non PRUB)
14	Input available $\geq$ 256 bytes (Non PRUB)

Another status code is used for the dump protocol. See Section 7.1.2.

Code Value	Name
8	Ready for Dump

7.1.1.2 Data Formats

In two of the four coupler operational modes, 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.

7.1.1.2.1 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.

7.1.1.2.2 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.

7.1.1.3 Modes of Operation

7.1.1.3.1 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.

7.1.1.3.2 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 is terminated when the PPU sets the channel inactive after loading the first of two load modules into NPU main memory. The PPU then issues a Start NPU function code, causing the NPU to load μmemory from main memory and issue an Idle status code to the PPU. The PPU clears the NPU, specifies Memory Address Zero and One and repeats the load procedure with the second load module containing main memory contents. The second load is terminated when the PPU sets the channel inactive and again issues a Start NPU function code. The NPU responds with an "NPU Initialized" SM, after NPU initialization has completed.

7.1.1.3.2 Load/Dump (cont.)

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, a "halt" message is issued to the CYBER's dayfile, a dump of main memory, and file registers is performed, and the operator is notified of NPU load failure. It is the responsibility of the CYBER host operator to reinitiate the downline load or notify customer engineering (in the event a hardware error prevented loading).

7.1.1.3.3 Single Word Transfer (Control)

The NPU can write the NPU Status Word. 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. 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.

7.1.1.3.4 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.

7.1.1.3.4 Multiple Character Data Transfer (Block Transfer) (cont.)

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.

7.1.1.4 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 7-2.

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.

7.1.1.5 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.

7.1.2 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 7-2) and the other showing operation of the NPU (Figure 7-3).

7.1.2 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.

7.1.2.1 Timers

Five timers are used by this protocol: three in the PPU and two in the NPU.

7.1.2.1.1 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 down. The NPU Dead Timer is not explicitly shown on Figure 7-3, but is implicit in the notation, as explained in Section 7.1.2.2.

7.1.2.1.2 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.

7.1.2.1.3 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
3. PRUB (Batch) data

Downline (Separate queues in Host)

1. Service Messages and Reverse Supervision (BACK's, BRK's, STP's, STRT's)
2. Data Blocks and related Forward Supervision of the highest priority (host may queue Interactive traffic here)
3. Data Blocks and related Forward Supervision at the next lower priority (host may queue Batch traffic here)
4. Data Blocks and related Forward Supervision of the lowest priority (host may queue Host to Host traffic here in future applications).

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 4/3, then both Levels 4/3 and 3/2, then Levels 4, 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.

7.1.2.2 Description of the State Diagrams

A larger arrow is used in some places on Figure 7-3. 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 dead timer 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 dead timer timeout occurs before the interrupt, the HIP calls a routine to note Host Unavailable - see Section 7.1.3 and restart the protocol at "A".

7.1.2.2 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 the NPU by sending "Not Ready for Input" or request OUTPUT.
- 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.

7.1.3 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.

Table 7-2. 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.

Table 7-3. 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.

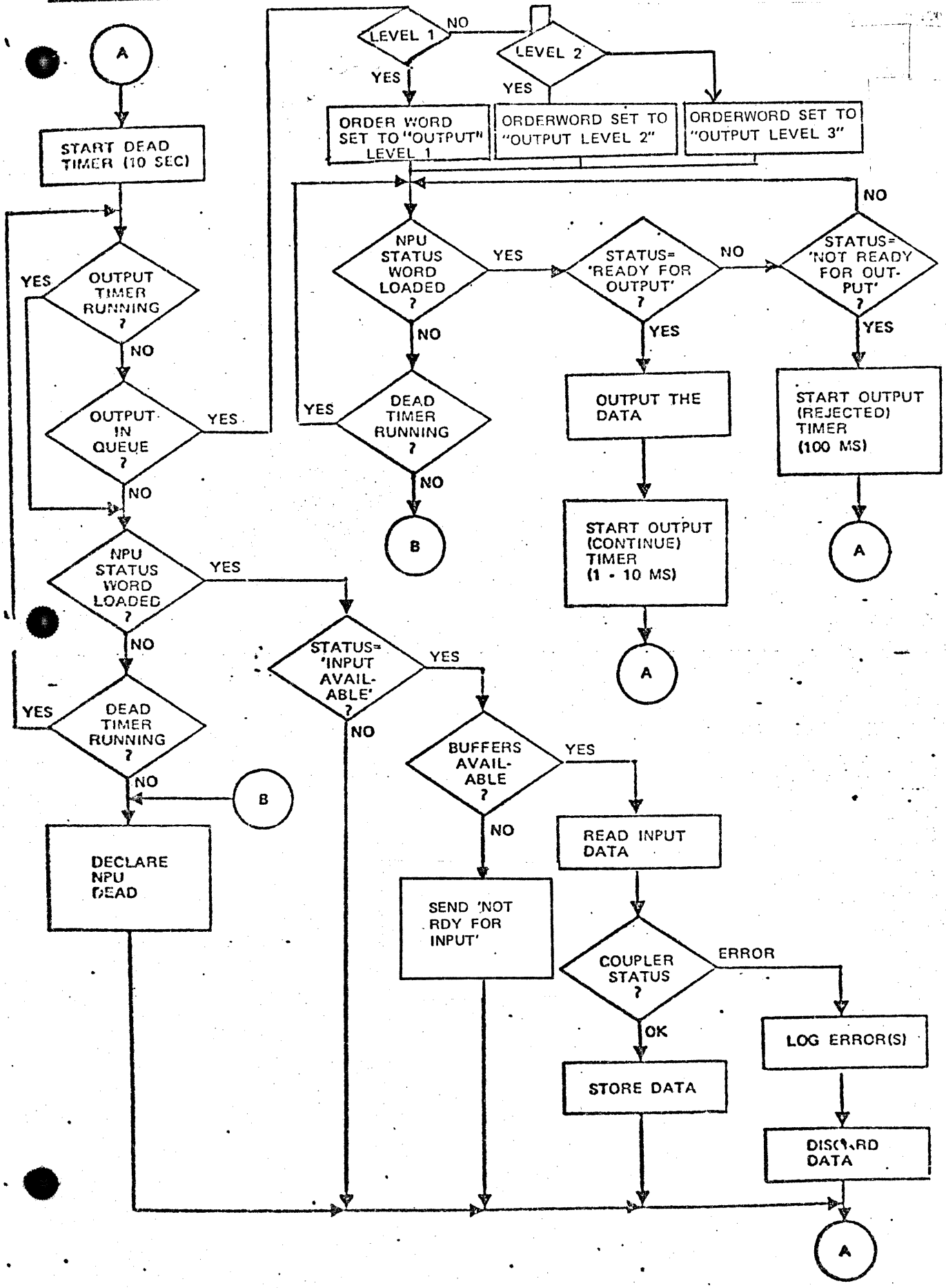
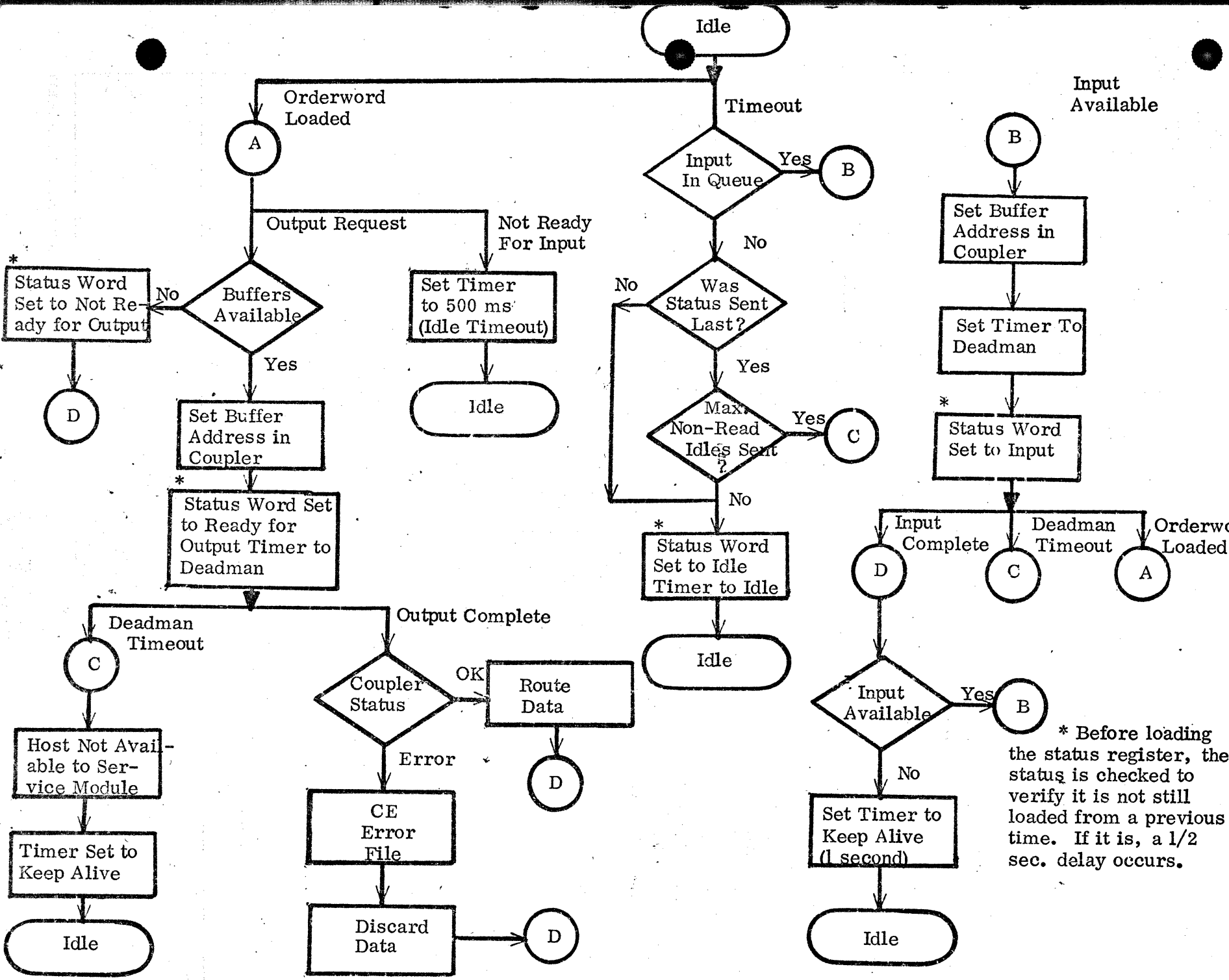


Figure 7-2. Data Transfer Protocol - PPU Sequence



* Before loading the status register, the status is checked to verify it is not still loaded from a previous time. If it is, a 1/2 sec. delay occurs.

Figure 6-3. Data Transfer Protocol - NPU Sequence

8.0 This section has been deleted.

9.0 BSC Terminal Interface Package

The Binary Synchronous Communications (BSC) TIP provides for the interchange of data between an application's process in the host computer and a remote 2780, 3780 or compatible batch terminal. The line protocol used is BSC operating point to point on a dedicated or dial in line.

9.1 Overview

Each BSC terminal consists of a card reader and a line printer and may optionally have a card punch. Remote batch operation with the terminal provides for support of the standard Intercom remote batch features and commands with minor extensions necessary to support 2780 and 3780 terminals as generally described in Section 2.1.

The operational procedures for submittal of remote batch jobs and return of generated print or punch files for the 2780 and 3780 terminal will be patterned after that described for the 200 UT terminal. Certain differences exist in the operation due to constraints imposed by the terminal characteristics. The Intercom Remote Batch command set is available to a user at the terminal. The commands are entered from pre-prepared cards at the card reader. One command is allowed per card. Messages to the terminal operator from the host and other unsolicited diagnostics and error messages are likewise directed to the printer. Interactive message output will be followed by a form feed to position the message for reading. Interactive message output will occur at input and output file boundaries only.

For 2780 and 3780 terminal input and output data, transfers are one direction at a time for the entire file transfer. That is intermediate blocks of input and output files may not be interleaved to or from the terminal as with Mode 4, 200 UT operations.

The TIP must resolve which stream is active at any one time. There are four or optionally five (card punch) different streams that must be managed to or from the two or optionally three devices on a terminal. The three possible devices are card reader, line printer, and card punch and the five possible streams are interactive card input, interactive printer output, batch card input, batch card output and batch punch output. The general rules the TIP uses to determine which stream is active are described below.

After "dial in", the host will output an "Intercom Banner" message on the interactive stream which will be delivered to the line printer. Input from the terminal will then be accepted from the card reader and will be treated as "interactive input". For dedicated lines, input will be accepted immediately after the terminal is configured (after the Banner is output).

9.1 Overview (cont.)

All input is treated as "interactive input" until a "Start Input" CMD on the batch connection is received from Intercom. (Intercom will generate the "Start Batch Input" CMD to the TIP immediately upon receipt of a Read or Read File Name command from the terminal.) After receipt of the "Start Input" CMD, all subsequent input is assumed to be "batch input" until EOT is received from the terminal at which time the TIP will revert to processing input as "interactive input" until the next "Start Input" CMD is received.

Interactive output is generated by the host and sent to the TIP. All interactive output messages in queue will be delivered to the line printer immediately following the completion of any active batch input or output file (EOT received or transmitted) or immediately following EOT received on the "interactive input" stream. After output of EOT to the terminal, a 3 sec delay will be initiated by the NPU to allow input before starting the next output file.

When contention exists between batch input and batch output streams, input will take precedence. When contention exists between two batch output streams, printer output will take precedence.

9.1.1 Operational Characteristics

Each terminal may be operated in the non-transparent mode and optionally, if the feature is supported by the terminal, may operate in the transparent mode. Also the operational capabilities of the 2780 and the 3780 terminals differ somewhat. The operational characteristics of each terminal for transparent and non-transparent modes are described in the following sections.

9.1.1.1 2780 Input Non-Transparent Terminal Mode

Commands are entered one per card and may be stacked in the card reader only if an ETX is punched as the last column of each command. Commands may be input without an ETX punched in the last column if only one command is placed in the reader at a time and the EOF toggle switch on the terminal is "on".

The last command entered before an input file or job deck must be either a READ or a READ, FILE NAME. The job deck or input file may be stacked directly behind the READ or READ, FILE NAME command if the ETX is punched in the last column. If the ETX is not included, the command must be entered separately from the input file or job.

The first card of batch input is assumed to be a JOB card and has special meaning to the TIP. Batch input may be terminated in one of three ways: 1) a /* EOI in columns 1 thru 7; 2) an ETX in the last column of the last input card (may be first column of a separate card); or 3) input of the last card with the EOF toggle switch on. Note that if ETX is used to terminate a job, another job cannot be stacked directly as the TIP will treat the next input after an ETX/EOT as interactive input.

9.1.1.1 2780 Input Non-Transparent Terminal Mode (cont.)

The TIP does not distinguish between batch input initiated by a Read or Batch Input initiated by a Read, File Name, however the following rules apply for Intercom even though no special checks are made in the TIP. If the data transfer was initiated by a Read command, multiple jobs may be stacked in the reader with each JOB terminated by /*EOI. Multiple /*EOI cards between jobs will be discarded by the TIP. The first non /*EOI card is assumed to be the job card for the subsequent job. If the data transfer was initiated by a Read, File Name command, only one file may be stacked in the reader. Subsequent input must be initiated by a new Read or Read, File Name command.

9.1.1.2 2780 Input Transparent Terminal Mode

If the 2780 terminal has the transparent option, data may be input with the transparent switch "on" but some operational difference from the non-transparent mode will apply.

Each command must be entered separately with the EOF toggle switch "on" as ETX is not recognized in this mode.

Each card input causes a full 80 characters to be transferred across the line and each card is transferred as a separate transmission block. Operation in this mode is less efficient than in the non-transparent mode.

All other characteristics are the same as in the non-transparent terminal mode.

Note: This mode should not be confused with the transparent data feature initiated by a "TR" optional parameter in the Read File Name command or by placement of a TR in columns 79 and 80 of the EOR card of a job deck. (See Section 9.1.1.5.)

9.1.1.3 3780 Input Non-Transparent Terminal Mode

The operational characteristics of the 3780 terminal with respect to card input are the same as those described for the 2780 except as described below:

ETX may not be punched in a command card or as the last card of an input job or file to terminate input.

Each command card (interactive input) must be input separately with the EOF toggle switch "on".

Multiple jobs may be stacked in the reader separated by a /*EOI card(s) but the last one must be input with the EOF toggle switch "on".

9.1.1.4 3780 Input Transparent Terminal Mode

If the 3780 terminal has the transparent mode option, data may be input with the transparent switch "on".

The operational characteristics in this mode are identical to the non-transparent terminal mode. However, there are some differences with respect to line efficiency and number of blocks transmitted. Each card input causes all 80 characters to be transmitted across the communication line; i.e., trailing blanks at the end of card are not suppressed and embedded strings of blanks and zeros are not compressed as in the non-transparent mode.

9.1.1.5 Input Transparent Data Mode

The TIP provides another mode of inputting the batch stream for both 2780 and 3780 terminals where the data is sent to the host without translation by the TIP. This mode of input may be specified only when the terminal is operating in the transparent mode. Three methods are provided to enter the transparent data mode. For local input files, an optional parameter "TR" is included as part of the Read File Name command or a "TR" may be included in columns 79 and 80 of either the Intercom Job card or EOR card for normal input jobs.

When "TR" is specified, all data following is not translated and is stored as 8 bit characters. No EOR (7-8-9 punch) are recognized when in this mode. Therefore, a file must be terminated using the EOF toggle switch.

Transparent 8 bit characters are stored in the PRUB without marking any record boundaries such as 80 column card boundaries. A BSC transparent transmission block may contain single records which are terminated by DLE ETB or multiple records with each record separated by an DLE ITB (DLE ITB and DLE ETB are not stored in the PRUB). The 3780 also has an optional feature to input four (4) fixed length 80 character records in each transmission block. This feature is not specifically supported by the TIP and, if used, will result in 320 characters stored in the PRUB for each transmission block received.

In any case, characters will be stored as received in the PRUB blocks without regard to record or transmission block boundaries. Transmission blocks will be split across PRUB boundaries and input will be stored until ETX is received.

9.1.1.6 2780 Output Non-Transparent Transmission Mode

Output streams may be directed to either the line printer or the card punch (provided the terminal has the punch option). The host software determines which device is to receive output and the output to each device is controlled by separate connections.

9.1.1.6 2780 Output Non-Transparent Transmission Mode (cont.)

The TIP accepts PRUB blocks from the host and converts the data from display code to external EBCDIC code and formats print lines into BSC transmission blocks for output. Each transmission block is made up of multiple print lines (records) where the number of lines is either limited to two or may be up to seven, depending on the terminal option defined to the system. In any case the transmission blocks may not exceed 400 characters and a print line is never split across transmission block boundaries.

The first character of each print line in the PRUB block is an Intercom carriage control character and is translated to BSC printer carriage control as defined in Table 2-3. The algorithm used for the conversion is shown in Table 9-1.

The carriage control operations performed by the TIP may be suppressed by use of the Intercom command "SUP" (suppress). If this option is envoked, the host must set the "suppress" field in the TCB and the first character of each print line will be discarded (with no specific carriage control characters transmitted to the terminal; a space 1 after print will occur automatically).

A "P" carriage control character followed by a "M" character will envoke the standard Intercom feature for embedding an operator message within the print file. For the 2780, the print line following the "PM" will be printed on the line printer preceded and followed by a "skip to top of page" carriage control. Output to the printer will then stop and the host notified of the condition by an upline break CMD. The TIP will then allow "interactive input" from the card reader and will not resume printing until commanded by the host.

Table 9-1

Interactive DBC Code	Intercom Batch Carriage Control Character	TIP Action For 2780
0 or 1, 8, 9	1 New Page	Transmit null line with skip to channel 1, then transmit the output line with the automatic space after print
2, 10	⊬ Space 1	No special action – a space 1 after print will occur automa tically
3, 11	+ No space	No special action – a space 1 after print will occur automatically (cannot be suppressed)
4, 12	0 Space 2	Transmit null line which will cause the automatic space, then transmit the output line with automatic space 1 after print
5, 13	– Space 3	Transmit null line with space 2 carriage control character then transmit the output line which automatically spaces 1 after print.
All Others	All Others	No special action – a space 1 after print will occur automat ically

9.1.1.6 2780 Output Non-Transparent Transmission Mode (cont.)

The printer line width may be defined for each terminal as any value from 50 to 150 characters by use of the Intercom Define command after Login. The host changes the TIPs terminal line width by sending a Reconfigure TCB SM with the appropriate FN/FV pair. The default line width will be set to 144 characters. Characters in a PRUB block print line that are in excess of the PRUB block print line are printed on the next line. Trailing blanks for short print lines are not transmitted to the terminal.

Transfer of an output data file to either the printer or punch normally continues until completion or a failure condition occurs. A method of interrupting an output stream is also provided to allow input of commands to the interactive input that might be necessary to change the disposition of the output stream. The method used for interrupting output called "intervene" may differ operationally for 2780 and 3780 terminals and emulators of these terminals.

In general, making the printer not ready will cause a timeout for some length of time (approximately 30 seconds) then input from the card reader will be allowed. Making the printer ready before the timeout has expired will cause continuation of the output file. In some cases several print lines may be duplicated. The timeout period is a function performed by the terminal itself and is usually extendable by pressing the "printer stop" key again.

In any case, an EOT received from the terminal as a response to an output block will cause that "output stream" to be stopped and input from the card reader allowed.

Card punch output is processed by the TIP in a manner similar to print output except the first character of each card (BSC record) is treated as data instead of carriage control and each record must be 80 characters or less in length. Characters within the PRUB record in excess of 80 will be punched on the next card. An option is provided to punch an "EM" character as the last character in each card that contains less than 80 characters of data. This option may be specified using the Intercom Define command and will result in a Reconfigure TCB SM to the TIP. The default value will be set for no EM character.

9.1.1.7 2780 Output Transparent Transmission Mode

Output to a terminal is normally transmitted as non-transparent transmission blocks. If the terminal contains the transparent mode optional feature, data will be output in the transparent transmission mode only if the data file being output is specified as transparent. Transparent data files are designated to Intercom by parameters within Intercom Route command. Intercom in turn marks the data file as transparent in the DBC field of each PRUB block of the file.

9.1.1.7 2780 Output Transparent Transmission Mode (cont.)

The TIP processing of transparent output blocks is different than non-transparent blocks. PRUB block characters are output as transmission blocks without code conversion and carriage control transforms. Characters are taken from the PRUB to make up transmission blocks without regard to record markers (FF character) or the PRUB boundary. Characters will be transferred from the PRUB to the transmission block until the transmission block has reached its defined maximum or the last character of an EOR or EOI PRUB has been transferred. An EOI PRUB terminates the file. An EOR PRUB terminates only the transmission block. The next transmission to the terminal will continue with the next PRUB block from the host and a new transmission block.

9.1.1.8 3780 Output Non-Transparent Transmission Mode

Output to the 3780 terminal functionally provides the same capabilities as with 2780 terminals. The internal processing differences from those described for the 2780 are described below.

The number of print line records or card records is not limited to 2 or 7. The number of records included in a transmission block is limited only by the transmission block size which is normally 512 characters, but may be configured to any size. No record may be split across a transmission block boundary.

Output to the 3780 follows the rules for data compression for that terminal; i.e., multiple blanks are compressed.

Short records (print lines or cards) are terminated by the EBCDIC IRS character and trailing blanks are transmitted if contained in the PRUB record.

Carriage control for 3780 printer output is designed to be directly compatible with Mode 4 terminal carriage control as shown in Table 2-3. The algorithm used by the TIP for conversions from Intercom carriage control characters to 3780 carriage control is shown in Table 9-2.

Table 9-2

Interactive DBC Code	Intercom Batch Carriage Control Character	TIP Action for 3780
1, 9	1 New Page	Transmit null line with skip to channel 1, then transmit the line with suppress space after print carriage control
	b Space 1	Transmit a blank line with space 1 after print carriage control followed by the text line with suppress space carriage control
4, 12	+ No space	Transmit the line with suppress space after print carriage control
	0 Space 2	Transmit a blank line with space 2 after print carriage control followed by the text line with suppress space carriage control
	- Space 3	Transmit a blank line with space 3 after print carriage control followed by the text line with suppress space carriage control
All Others	All Others	Transmit null line with skip to channel 1, then transmit the line with suppress space after print carriage control

10.0 MODE 3 (TTY) TERMINAL INTERFACE PACKAGE

The Terminal Interface Package (TIP) for MODE 3 terminals provides a set of procedures for the interchange of interactive data between the host processor and MODE 3 terminals. The MODE 3 terminals can be teletypes or teletype compatible terminals.

10.1 Overview

The TTY TIP supports single terminal switched or dedicated asyncronous lines at speeds of 110, 150, 300, 600, 1200, 2400, 4800, and 9600 BAUD. The lines are considered to be half duplex, i.e., the TIP can be transmitting or receiving on a given line, but not both simultaneously. No code translation or parity check is performed on data characters input from the terminal. Data characters output to the terminal are output as received from the host with the exception of the character parity bit which is complemented when necessary to make all output characters have even parity.

10.2 Data Formats

Characters transmitted between the host and terminal during both input and output are passed in full 8-bit form, without code translation or parity generation and checking. Characters sent by the TTY TIP to the terminal always have even parity. An input message is sent in one or more blocks. The maximum size of an input BLK or MSG block generated by the TIP is controlled by a program build parameter in the range up to 2047. When the message length exceeds the maximum block size parameter, all but the last block will be type 1 (BLK) blocks and the last block will be a type 2 (MSG) block. Where the message length is less then the maximum block size, the single block sent is of type 2 (MSG) type.

10.3 Operating Modes

The TIP operates in either the interactive mode or the tape mode.

10.3.1 Interactive Mode

In the interactive mode the TIP interfaces the network to a teletype (or TTY compatible) device for either input or output. The interactive mode will handle input from a TTY keyboard and also blocks (Block Mode) characters from a TTY compatible device where the characters are received at line speed.

In the interactive mode, the TIP operates in half-duplex fashion with three basic states (idle, input and output). The TIP is driven from the idle state to the input state by the arrival of a character which is not a carriage return, line feed or pad ($FF or $7F). If any of these three characters is received while in the idle state, the character is discarded and the TIP remains in the idle state. Once in the input state the TIP remains in that state until an end-of-input-message is received. The end-of-input-message for interactive mode is a carriage return. Line feed characters received from the terminal will cause the TIP to output a carriage return if another character is not received within 100 msec.

10.3.1 Interactive Mode (cont.)

Carriage return characters are echoed with line feeds when another character is not received from the terminal within 100 msec.

The TIP is driven from the idle state to the output state by the arrival of an output message from the host. After the output message queue has been emptied, the TIP returns to the idle state. If an input character is received from the terminal when the TIP is in the output state, the current message being output is placed back on the top of the output queue, the character received from the terminal is discarded, and the TIP goes to the input state.

10.3.2 Tape Mode

In the tape mode the TIP interfaces the network to a tape reader (paper tape or tape cassette). Tape mode only applies to input as output is identical for all terminals. The TIP is commanded to enter the tape mode by a downline command from the host. The start input command, with a primary function code of 1, will cause the TIP to send an X-ON character ($11) to the terminal which will start the tape reader. Message blocks are sent upline to the host when a carriage return character is received from the terminal. The X-OFF character is the last data character of the message block which is sent upline to the host. After the X-OFF is received, the TIP returns to the interactive mode.

10.4 Carriage Control

The carriage control for interactive TTY terminals is defined by the Data Block Clarifier (DBC) character received from the host. Only the least significant four (4) bits of the DBC are used by the TIP to determine the carriage control character sequence. The carriage control character sequence is sent to the terminal prior to the first character in the output message block.

DBC	Character Sequence
0	CR, LF
1	CR, 3LF
2	CR, LR
3	CR, LF
4	CR
5	(Nothing)
6*	CR, LF
7*	CR, LF
8	CR, LF, 4 nulls
9	CR, 3LF, 4 nulls
10	CR, LF, 4 nulls
11	CR, LF, 4 nulls
12	CR, 4 nulls
13	4 nulls
14*	CR, LF, 4 nulls
15*	CR, LF, 4 nulls

*C.E. Error message generated by TIP for invalid DBC received from Host.

10.4 Carriage Control (cont.)

The number of line feed characters in the above character sequences is decremented by one if: 1) The line feed count in the sequence is nonzero, 2) A line feed was the last character sent to the terminal in response to a carriage return received from the terminal, 3) The system parameter GOLFSTRIP is assembled as TRUE.

10.5 Auto Baud Recognition

The TIP will perform auto-baud recognition of lines which have been configured as switched auto-baud by the host. The three baud rates which can be recognized are 110, 150, and 300. After the terminal has been dialed in the operator must enter a carriage return character as the first character to enable the TIP to determine the baud rate.

10.6 Input Regulation

Before soliciting or accepting input messages, the TIP checks buffer thresholds to ensure a sufficient supply of the size required. A higher threshold value is used for new messages than for messages already in progress. This gives higher priority to messages in progress when contending for available buffers. When the regulation level is reached an "INPUT STOPPED" message is output to the terminal. When sufficient buffers become available a "RESUME INPUT" message is output to the terminal.

10.7 Commands

The general format for commands is found in section 8 of this specification. Listed below are the upline and downline commands for the TTY TIP.

10.7.1 Upline Commands

Primary Function	Secondary Function	Description
3	0	Input stopped, X-OFF received

10.7.2 Downline Commands

Primary Function	Secondary Function	Description
1	X	Start input. Used to initiate tape mode. TIP will send X-ON to the terminal to start the reader.

11.0 HASP Terminal Interface Program

The Terminal Interface Program (TIP) for the HASP Terminals provides a set of procedures for the interchange of data between a host processor and a HASP Terminal (usually referred to as a HASP multileaving workstation). The TIP supports one interactive device (console) and multiple (up to 7 of each type) card readers, printers and punches per workstation. HASP workstations without a console are not supported.

11.1 Overview

The interface provided for HASP terminals is compatible with IBM's version of the HASP multileaving protocol, but does not necessarily emulate this interface exactly. A list of the general terminal features supported is given below.

Feature	Supported
Multi-card	Yes
Character set	EBCDIC-64
EBCDIC Transparency (256 char)	Yes
Character Compression/Expansion	All character strings
Console	Mandatory
Printer Character Set	EBCDIC-64
Punch	Yes
Print Width	80-150
Binary Cards	No
Plotter	
Magnetic Tape	As card/print emulator
Paper tape	

The TIP interfaces with the host processors using the block protocol and data block formats for CCP 3.0 as defined in section 3 of this specification. Two types of data are supported to and from the host interface, transparent and non-transparent. Non-transparent data is treated as Six bit characters formatted for cards on print line images. Transparent data is treated as eight-bit characters and is blocked and de-blocked between the terminal without regard for card lengths , carriage control or print-line width.

The TIP design is insensitive to line speeds, but has been tested at standard Synchronous line speeds thru 9600 baud. Lines may be dedicated (with or without a modem/transceiver) or switched (dialup) with a modem. The transmission facilities are used by the TIP in a half duplex manner, i.e. the TIP is either transmitting to the line or receiving from the line, but not both simultaneously.

The basic line protocol is standard BSC point to point (one terminal per line) and either transparent or non-transparent modes of BSC transmission are automatically recognized by the TIP on each received data block from the terminal. The TIP then uses the detected transmission mode for subsequent communication with that terminal.

11.1 (Continued)

A line may be configured for auto recognition of either a HASP or 2780/3780 terminal. When communications is first established with a terminal (after dial in),the software will determine whether the terminal is a 2780/3780 terminal or a HASP workstation and automatically begin service of that terminal using the correct terminal protocol. (If 2780 or 3780 the terminal, the Host must declare which terminal type after communication is established). The 2780/3780 will only report auto recognition of the 2780 termonal type.

11.2 Host Interface

The host interacts with the TIP via the block protocol, the commands defined in table 3-1 and the block protocol flow outlined by the diagrams in section 3.1.4.3.1. The following sections further clarify the way in which the commands, data block clarifier, break codes, etc. are used within the HASP TIP.

11.2.1 Configuration and addressability

The HASP terminal may consist of up to 7 devices of each type in addition to the console which is mandatory. Each device has a separate stream ID and is specified in the configuration service messages from the host in the Terminal Address (TA) field. The cluster address (CA) field is always zero. Stream numbers are identified from 1 to 7 for each device type. Table 11-1 list possible CA,TA,DT values for HASP devices.

CA	TA	Device Type	Stream/Device Number
00	01	09	Console 1
00	01	29	Card Reader 1
.	.	.	.
.	.	.	.
.	.	.	.
00	07	29	Card Reader 7
00	01	49	Printer 1
.	.	.	.
.	.	.	.
.	.	.	.
00	07	49	Printer 7
00	01	69	Punch 1
.	.	.	.
.	.	.	.
.	.	.	.
00	07	69	Punch 7

Table 11-1

11.2.1 (Continued)

Intercom commands or status to or from the terminal operator use the stream/device number when referring to a particular device e.g. line printer number 1 would be "LP 1".

11.2.2 Interactive

Each HASP terminal must have a console device. The TIP accepts BLKS or MSG's from the HOST and delivers each to the terminal. There is no difference in the TIP logic for BLK's verus MSG's. Data is converted from ASCIZ to EBCDIC code prior to output to the terminal, but no other transformations take place. The DBC carriage control values are ignored by the TIP since the HASP workstation does internal carriage control and screen formatting to its interactive device.

Blocks received from the terminal are delivered to the host after conversion from EBCDIC to ASCII code.

The console stream is always open and does not require any commands to start or stop the streams. If the terminal stops the console stream (by resetting the console stream FCS bit) for more than 30 secs contiguously, the terminal is assumed to be inoperative and the host is notified by an INPUT STOPPED CMD , NO RESPONSE (PFC=3, SFC=4).

11.2.3 Card Reader

A HASP card reader is activated by the START INPUT command on the card reader connection. The card reader connection once activated is run in a so-called "hot card reader" mode. That is, the card reader stream is always on unless terminated by: (1) input of a /*EOS (end of stream), (2) the HASP WORKSTATION or the communications line fails and a recovery of the entire line/workstation takes place, or (3) a "STOP INPUT" (PFC=2,SFC=0/1) command is received from the host.

The card reader stream must be terminated to change from a Read of non-transparent data to a Read Filename or a Read Filename with transparent data, but a mode change request from the interactive console will not be acted on by the host software unless the card read stream is currently receiving data. Due to this restriction, the stream must normally be terminated by entering a /*EOS card when changing modes.

If the last input received from the card reader, terminated the previous input file (either /*EOI or ETX), any subsequent /*EOI or /*EOS card will be discarded by the TIP. A /*EOS card will always result in an "INPUT STOPPED" (PFC=3, SFC=0) to the host and subsequent data received from the card reader will be discarded until the stream is started again.

The normal termination of an input job with a /*EOI card or ETX from the terminal will result in the termination of the PRUB data block with "EOI" marked in the DBC field. No upline command will be sent to the host in this case.

11.2.3 (Continued)

Since card reader "not ready" is not reported by the HASP workstation, this condition is not reported to the host.

A STOP INPUT command received from the host will result in termination of the input stream and will, in addition, result in a INPUT STOPPED (PFC=3, SFC=0) being returned to the host signifying recognition of the STOP INPUT. Any subsequent data received from the card reader on that stream will be discarded by the TIP.

Data received from the card reader is transformed to PRUB format in either the transparent or non-transparent modes. The stream may be placed in the transparent mode by one of three methods. For local input files, an optional parameter "TR" is included as a parameter in the READ FILENAME command or a "TR" may be included in columns 79 and 80 of either the JOB card or EOR card. A change in the transparent mode by the READ FILENAME will result in a START INPUT command from the host to the TIP specifying the mode. When "TR" is specified, all data following is not translated and is stored as 8-bit characters. No EOR (789 punch) or /*EOI cards are recognized when in this mode. The file is read until ETX is received from the terminal.

The HASP TIP also examines columns 79 and 80 of all JOB cards and EOR cards to determine if the code translation should be the 026 or 029 character set. A 26 in column 79 and 80 specifies 026 and a 29 specifies 029. The code translation defaults back to the Intercom default code set after a /*EOI card is detected.

11.2.3 (Continued)

A summary of the commands used on the HASP card reader stream is given in the following table.

PFC	SFC	NAME	MEANING
1	0	START INPUT NON-TRANSPARENT	Downline command to start a card read input in the non-transparent data mode.
1	1	START INPUT TRANSPARENT	Downline command to start a card read input stream in the transparent data mode.
1	2	RESUME INPUT	Downline command to resume card read input after a suspend input. Note, this command resets suspension of all streams on a workstation when issued on any stream.
2	0	STOP INPUT TERMINATE	Downline command to STOP a card read stream and discard all data received.
2	1	STOP INPUT SUSPEND	Downline command to stop a card read stream and hold all data received. Note, this command suspends all streams on a workstation when issued on any stream.
3	0	INPUT STOPPED	Upline command signifying /*EOS received on a card read stream. Also sent upline when a STOP, INPUT, TERMINATE is received to signify all data has been terminated on the connection.

Table 11-2

11.2.3.1 Non-Transparent Data Mode

When in the non-transparent data mode, characters received are expanded from the HASP compressed format, translated to display code and stored in the standard PRUB format. Trailing blanks on each card are not stored and the end of each card is marked within PRUB with from two to eleven zero characters (Modulo ten characters for the entire PRUB, but must have at least two).

11.2.3.2 Transparent Data Mode

Transparent 8-bit characters are expanded from the HASP compressed format and stored in the PRUB without translation or marking of card boundaries. Records or transmission blocks are stored contiguously within the PRUB and, therefore, may be split accross PRUB boundaries. Data is stored until ETX is received. When ETX is detected, the last PRUB is marked as EOI in the DBC and the card read stream is returned to the non-transparent data mode.

11.2.4 Printer

Output to the printer is activated by the host sending the first downline data block on a printer connection. The first block received is normally a banner MSG block. If the terminal is configured for the "banner off" condition, the banner block is discarded; otherwise, the MSG is converted to two copies of the file identification banner page.

Each subsequent PRUB is converted to output transmission blocks according to the BSC point to point and the HASP multi-leaving protocols described in section 11.3.

The DBC of each PRUB block received is examined for transparent or non-transparent data and the data converted to a transmission block accordingly.

11.2.4.1 Non-Transparent Data Mode

Data within the PRUB is considered print lines and the end of each line is detected according to the standard PRUB format (section 3).

If the print line taken from the PRUB is greater than the printer line width configured, the excess characters are automatically printed on the next line. Print lines are never split accross transmission blocks. The first character of each line is normally interpreted as a carriage control character and is converted to the corresponding HASP WORKSTATION Sub-Record Control BYTE (SRCB) according to table 11-3. Optionally, if the stream is configured for "Suppress carriage control", the first character of each line is ignored and replaced by a single space before print.

All characters are converted from Display code to EBCDIC code prior to output.

11.2.4.1 (Continued)

Intercom Code	Function Before Print	HASP SRCB Value
1	New Page	$B1
+	No Space	$80
0	Space 2	$A2
-	Space 3	$A3
⊬	Space 1	$A1
Others	Space 1	$A1

Table 11-3

Printer Carriage Control

11.2.4.2 Transparent Data Mode

For PRU's marked as transparent data, no print lines are detected within the PRUB. Characters are placed in the transmission block without code conversion, carriage control or end of line processing.

If an EOR or EOI block is received, the transmission block will be terminated with the last character of that PRUB. Otherwise, transmission blocks are filled to the maximum size configured.

11.2.4.3 Command Interface

Output of print files to the printer continues with no intervention required between files. The host is not notified when the last block of a file is output. The host is notified by an "output stopped" printer not ready (PFC=5,SFC=1") if the stream is suspended by the terminal for more than 30 secs. The TIP also notifies the host and stops the printer stream when a "PM" message is detected as the first two characters of a print line. The command sent for this condition is Output Stopped, PM(PFC=5,SFC=3). The PM print line is then sent to the console device.

The print stream may be restarted after a stop condition by an Intercom Command to the host which results in a down line RESTART OUTPUT CMD (PFC=7,SFC=0). The print stream may also be aborted by the down line command STOP OUTPUT(PFC=8,SFC=0) which causes any PRUB queued for the stream to be discarded.

A summary of the commands applicable to a HASP print stream is given in the table below.

PFC	SFC	NAME	MEANING
5	1	OUTPUT STOPPED, PRINTER NOT READY	Upline command to notify the host the print stream has been suspended by the terminal for more than 30 secs.
5	3	OUTPUT STOPPED, PM MESSAGE	Upline command to notify the host the print stream has been stopped due to receipt of a PM operator message.
7	0	RESUME OUTPUT	Downline command to cause the TIP to restart output after a stop condition.
8	0	STOP OUTPUT, ABORT	Downline command to cause the TIP to discard buffers and stop output to the printer.

Table 11-4

11.2.5 Punch

Card Punch output is processed by the TIP in a manner simular to print output except there is no carriage control and output HASP records are fixed at 80 characters or less in length. A lace card consisting of 70 characters of *, the 7 character Job name and 3 blanks, is punched as the first card of each job file as a separator.

Punch output also has an option to punch an "EM" ($19) character immediately following the last data character on each card that is less than 80 characters in length. The "EM" character allows card reading to be more efficient for some terminals.

Files output to the punch may also be specified as transparent data in the same manner as print files. In this case, they would be handled exactly as described for transparent data for the line printer except the lace card would be punched in place of the printer banner page.

A summary of the commands applicable to the punch stream are given in the table below.

PFC	SFC	NAME	MEANING
5	1	OUTPUT STOPPED, DEVICE NOT READY	Upline command to notify the host the print stream has been suspended by the terminal for more than 30 secs.
7	0	RESUME OUTPUT	Downline command to cause the TIP to restart output after a stop condition.
8	0	STOP OUTPUT, ABORT	Downline command to cause the TIP to discard buffers and stop output to the punch.

Table 11-5

11.3 HASP 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.

11.3.1 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

11.3.1 (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.

11.3.2 Multi-Leaving Block Descriptions

11.3.2.1 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.

11.3.2.1.1 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	70	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.

11.3.2.1.2 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.

11.3.2.1.3 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.

11.3.2.1.4 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.

11.3.2.2 Data Block Control Bytes

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

11.3.2.2.1 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

11.3.2.2.2 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

11.3.2.2.2 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

11.3.2.2.3 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.

11.3.2.2.4 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

11.3.2.2.5 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.

11.3.3 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 11.1.

11.3.4 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 11-1

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 11.9 and the BCB error blocks are described in the Error Conditions Section 11.8.

11.3.4.1 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.

11.3.4.2 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.

11.3.4.3 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)

11.3.5 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

11.3.5.1 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.

11.3.5.2 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.

11.3.5.3 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.

11.3.5.4 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

11.3.5.4 (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.

11.3.6 Terminal Start-Up and Termination

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

- Terminal initialization
- Communication line initialization
- Sign-On

11.3.6.1 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.

11.3.6.2 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.

11.3.6.3 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: normal1 /* SIGNON although data on this card is not checked or used. Any valid data block will initiate communications between the host and the terminal.

The Sign-On Block Format is:

(BSC Header)

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

11.3.6.3 Sign-On Block (cont.)

SRCB = H000001 - Initial Sign-On
Sign-On Record
RCB = 0000000 - Transmission block terminator

(BSC Trailer)

The Sign-On record is sent to the Cyber Host and causes the Control Data System banner to be output to the terminal requesting Login.

12.0 MODE 4 TERMINAL INTERFACE PROGRAM

The Terminal Interface Program (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.

12.1 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

12.2 Overview

The Mode 4 TIP interfaces with the host process using the block protocol and data block formats for CCP 3A as defined in Section 3 of this specification.

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 transceiver) 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 may be dedicated or switched. Where multiple terminals are on a line, the TIP services each terminal in sequential order withour priority.

The TIP will perform auto-recognition of terminal types when requested by the host. This procedure will determine the code set of the terminal (ASCH or External BCD) and mode (Mode 4A or Mode 4C).

12.2 Overview (cont.)

In addition the cluster address and terminal address flags 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. A separate connection is established for each Mode 4 device on a line; e.g., display card reader and printer.

The TIP supports one display, card reader and line printer for each Mode 4A terminal and multiple displays for Mode 4C terminals.

All terminal polling is performed by the TIP. A command from the host starts input from a card reader and the first output block to a printer begins output. Display devices are always "on" and will be polled whenever there is no contention with a batch stream or an output to the display.

The TIP resolves all contention between devices and controllers on a line and reports to the host any condition that has stopped an active connection except those caused by host command.

The TIP performs recovery for line and terminal errors, reports permanent terminal failure conditions and monitors for long term recovery of the terminal.

12.3 Host Interface

The host interacts with the TIP via the block protocol, the commands defined in Table 3-1 and the block protocol flow outlined by diagrams in Section 3.1.4.3.1 The following sections further clarify the way in which the Commands, Data Block Clarifier, break codes, etc. are used within the Mode 4 TIP.

Both ASCII and BCD Mode 4 terminals are supported. All data on an interactive connection between the host and the TIP must be ASCII code.
The TIP must translate all interactive characters to or from a BCD terminal to ASCII. All data on a batch connection between the host and the TIP is in display code. The TIP must translate data between the terminal code External BCD or ASCII and Display Code (Internal BCD).

12.3.1 Interactive Interface

The interactive 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 this interface with additional logic to resolve contention for the common display buffer used also for batch devices.

The display is activated by delivery of the first output to the device. Polling begins following delivery of data to the terminal and continues until the terminal is deleted or fails. Polling is suspended to deliver output to the display, and during Mode 4A batch I/O operations. Polling resumes after completion of these conditions.

Output has priority over input. If there is data in the output queue, the TIP builds a transmission block from multiple BLK blocks received from the host. The transmission block is transmitted to the terminal when either a MSG block is received or the transmission buffer is filled with BLK blocks from the Host. If the last blocks delivered is a MSG block polling is resumed. If the last block delivered is a BLK block, the TIP unconditionally waits for more output and does not resume polling. The TIP inserts a new line character after each BLK block within the transmission block and inserts the appropriate number of blanks at the end of the last BLK or MSG to position the curser at the beginning of the next line.

Input received while in the interactive mode is sent to the host as a MSG block.

During receipt of input, the TIP calculates the horizontal position of the cursor on the CRT screen. This calculation must consider that: 1) escape carriage-return from any Mode 4 CRT and 2) new line code from any Mode 4C CRT force the cursor to the left most position on the next line. After input the TIP generates a WRITE El block containing sufficient blanks to force the cursor to wrap around to the first position of the next line. To do this, the TIP needs to know the screen width. When the TCB is built, the screen width is initialized to Intercom default value. The screen width parameter can be set to any value by the downline Configure/Reconfigure Terminal Service Message (see Appendix A, Table A-8).

Interactive carriage control for each output line is based on the Data Block Clarifier (DBC) in each data block. Valid codes for the Mode 4 terminals are shown below.

For the 200UT, the use of the display will cause the card reader and printer connections to send a STOP CMD. These interchannel interactions are intended to signal the use of the 200UT transmission buffer which is shared by the display, card reader and printer.

12.3.2 Card Reader Interface

The Mode 4 card reader is activated by sending a START INPUT command on the card reader connection. The TIP transforms card reader data into PRUB record format (see Section 3.1.4.2). Trailing blanks on each card will be suppressed. Each block of data is reported to the host as a BLK block until an EOR (7/8/9 punch in column 1) or an EOI (6/7/8/9 punch in column 1) card is detected. A block containing EOR or EOI is reported to the host as a MSG block with appropriate flags set in the DBC header field. The EOR or EOI card is not included in the MSG block and multiple EOI or blank cards received after EOI are discarded until the first non-EOI/non-blank card is sensed.

The data following the last EOI is considered part of the next message thus allowing multiple jobs to be stacked in the card reader.

A file "level number" is taken from columns 2 and 3 of the EOR card, converted to a binary value and placed in the level number field of the upline PRUB header.

Columns 79 and 80 of the EOR and job card are not tested for either 26/29 option or TR (transparent) as neither feature applies to the Mode 4 TIP. Even though the 26/29 option is supported by some Mode 4 terminals, the special character conversions are performed by the terminal and are transparent to the CCP and Intercom support of the terminal.

When "card reader empty" is detected by the TIP, an "INPUT STOPPED" CMD is forwarded to the host following the last PRUB block. The "INPUT STOPPED" CMD will indicate "normal end" if the last card read was an EOI or a "break condition" if the last card was not EOI. Any partial PRUB will be saved in CCP buffers and the host must inform the TIP of the desired disposition of the partial PRUB. A START INPUT, RESUME CMD will cause card read input to continue into the same PRUB and a STOP INPUT, TERMINATE CMD will cause the partial PRUB to be discarded and card reader connection to be returned to the idle state.

12.3.2 Card Reader Interface (cont.)

An El response to a poll for cards causes an Input Stopped with a secondary function code of 5 for Batch Interrupt.

The possible secondary function codes for an Input Stopped CMD on the card reader connection are shown below.

(PFC=3 Input Stopped) Secondary Function Codes	Meaning
0	CR Empty EOI Received
1	CR Empty No EOI Received
2	Batch Interrupt
3	Slipped Card
4	No Response
5	Bad Response
6	Error Response

Section 12.4.7 has further clarifications of error codes 4, 5, 6.

Flow control for upline PRUB is regulated by the downline BACK block from the host. The TIP will temporarily suspend polling for card read data if the previous PRUB block has not been acknowledged and one additional transmission block has been received from the terminal. Polling will then continue when the BACK is received for the outstanding PRUB.

Table 12-1

DBC	Mode 4
0	Clear Write
16 (X'10')	Clear Write
All Other	Write

Note: The TIP will automatically insert a Clear Write on the first output to a Mode 4A display after a batch input or output operation.

The reason for break field appears in the upline command "Input Stopped" when input is terminated by the TIP. Interactive input is terminated by the TIP for abnormal conditions only. The secondary function break codes that may be generated by the interactive interface are shown below.

Table 12-2

(PFC=3 Input Stopped) Secondary Function Codes	Meaning
4	No response from Terminal
5	Bad response from Terminal; Unable to Select
6	Error response from Terminal; Unable to Deliver

Section 12.4.7 has further clarification on these break conditions.

Whenever an "Input Stopped", break command is generated on an interactive connection, any output queued for that connection is discarded.

12.3.3 Printer Interface

Output to the printer is activated by the host sending the first downline data block on a printer connection. The first block must be a MSG block in PRUB format and must be marked as a Banner MSG in the DBC field. If the terminal is configured for the "banner off" condition, the MSG will be discarded; otherwise, the MSG is converted to the file identification banner page.

Each subsequent PRUB is converted to output transmission blocks depending on the terminal code set and line width. If the print line taken from the PRUB is greater than the defined printer line width, the excess characters are automatically printed on the next line.

The first character of each line is normally interpreted as a carriage control character according to Table 12-3. Optionally, the carriage control character is ignored and replaced by single space when the "suppress carriage control" is in effect. This option is controlled by configuring/reconfiguring the "Suppress Carriage Control" flag in the Terminal Control Block.

Table 12-3

Function	Display Code	BCD Terminal	ASCII Terminal
New page	X'1C' (1)	X'41' (1)	X'41' (A)
New line	X'2D' (⅛)	X'50' (⅛)	X'20' (B)
Space 2	X'1B' (0)	X'4A' (0)	X'4A' (J)
No Space	X'25' (+)	X'B0' (+)	X'B0' (0)

The end of line character sequence is inserted at the end of each line of an output transmission block except the last. (The E code sequence is appended to the end of each transmission block and takes the place of the end of line sequence.)

The end of line control code sequences inserted by the TIP are:

BCD Terminal	ASCII Terminal
X'3E50' (ESC ⅛)	X'9B40' (ESC a)

12.3.3 Printer Interface (cont.)

The TIP will send a "BACK" for each PRUB received. The BACK is used for flow control only and is sent upline to solicit the next downline PRUB block. All lines of a PRUB block may not have been delivered to the terminal when the "BACK" is transmitted to the host for that block.

Output to the printer is always sent as a Clear Write MTI. A response code of E1 or E2, received when polling for the printer E code, will cause an upline OUTPUT STOPPED CMD. The secondary function code of the CMD will indicate 2 if E1 was received (batch interrupt) or 1 if E2 was received (printer not ready). Any undelivered PRUB data will remain in CCP buffers until the host issues a RESTART OUTPUT CMD causing the file output to be resumed or a STOP OUTPUT CMD causing all data to be discarded and the printer returned to the idle state.

The last PRUB block of an output file must be marked as a MSG block but does not cause an OUTPUT STOPPED CMD to be sent upline. It is used only to condition the TIP to expect a "Banner" as the next block.

The possible secondary function codes for an OUTPUT STOPPED CMD for the printer connection are shown below.

(PFC=5 Output Stopped) Secondary Function Codes	Meaning
0	
1	Printer Not Ready
2	Batch Interrupt
3	PM Message
4	No Response
5	Bad Response
6	Error Response

12.4 Mode 4 Terminal Interface

The Mode 4 TIP performs transformations between the host interface data control formats and the terminal data and control formats depending on the particular type of Mode 4 terminal that is being interfaced. The terminal type being serviced is determined by the line and terminal service messages as described in Section 5 and the appendix. The terminal type may also be automatically determined by the TIP and reported to the host in an upline Line Status Service Message (auto-recognition).

12.4.1 Auto-Recognition

The host may request auto-recognition for Mode 4 lines. This will invoke a procedure for determining the address and terminal that exists on the line. When the host enables 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 addresses 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 and the remote operator must press the send key on at least one of the displays. 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
CA	Cluster Address
TA	Terminal Address
DT	Device Type

For all terminals, the appropriate current 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 as appropriate for a 200UT. The terminal address for all three terminals will be X'60.

The configuration request terminal feature will be used for the Mode 4C terminals to determine the actual terminal addresses (TA) and actual device types (DT). Either a console or a printer may be reported and addresses of X'61'-6F are valid.

12.4.1 Auto-Recognition (cont.)

The printer device code (see Appendix A, Table A-2) for a mode 4C Impact Printer (IP) will be 2 and the code for a mode 4C "Non-Impact Printer" (NIP) will be 4.

The line status (operational) SM sent upline after Mode 4 auto recognition is shown below.

	PFC	FSC	P	SP	RC	LT	CFS	NT	TT	CA
Header	6	2	i	o	o	j	6	o	k	l

TA_1	DT_1	...	TA_n	DT_n
m	p	...	m	p

i = configured port number
j = configured line type (1,2,3)
k = 90 - Mode 4A BCD
91 - Mode 4A ASCII
92 - Mode 4C
l = Cluster Address of first terminal responding to poll of addresses "70" thru "7F"

Note that only one CA is reported as multi-cluster auto recognition is not supported. Multi clusters may, however, be configured on an auto recognition type line after the auto recognition is complete and the line is reported operational.

m = For Mode 4A, m will always be X'60'. For Mode 4C, m will be sequential values from X'60' thru X'6F' received from "configuration" poll of the l cluster address

p = For Mode 4A, three TA,DT pairs will be reported: TA='60', DT='0A' (console), TA='60', DT='2A' (card reader), TA='60', DT=4A (line printer).

For Mode 4C, up to 16 TA,DT pairs will be reported where the TA value may range from X'60' thru X'6F' and DT may be any of the following depending on the cluster configuration. '0A' = console
'4A' = impact printer
8A = non impact printer

12.4.2 Terminal Addressing

CA and TA are the Cluster and Terminal Addresses, respectively. The range of permissible values (in hexadecimal notation exclusive of parity) for the cluster and terminal addresses is a function of terminal type, as indicated below:

12.4.2 Terminal Addressing (cont.)

	200UT	711	714
CA Cluster Address	70-7F	20-7F	20-7F
TA Terminal Address *	60		61-6F
Cluster Controller		60	60
CRT/Keyboard		61	
Printer		64	

*Bit 4 of the Terminal Address is the toggle bit. It is shown here as zero, but may be a one. When the NPU transmits to the cluster, it changes this bit with each succeeding output. The input response to the output carries, in the same bit position, the same value if the output was correctly received by the cluster controller, or the opposite value if the output was not correctly received.

12.4.3 Transmission Block and Internal Block Formats

12.4.3.1 Code Conversion

All interactive data blocks to or from the host are in ASCII. The TIP does character conversion for BCD terminals. The ASCII to External BCD translation includes folding the lower case letters into uppercase and substituting blanks for any control code. Those BCD terminals having switch selection for internal or external BCD must have the switch set to external. No translation is performed by the TIP on interactive data to or from ASCII terminals.

All batch data blocks to or from the host are always in Display code (Internal BCD). The TIP does character for character conversion between the terminal code (external BCD or ASCII) and display code. For upline traffic from an ASCII terminal that has a 96 code character set, all lower case characters will be folded into upper case display code characters.

12.4.3.2 Transmission Block

Each byte, including the LPC, has odd parity in bit 7. LPC (Longitudinal Parity Check) is odd parity on bits 0-6 of all characters except SYN. The TIP inserts 7-14 SYN characters between the MTI and TEXT on all output where the MTI is Clear Write or Reset Write.

12.4.4 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 Data Block Clarifier in interactive mode only.

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	

POLL, ALERT, REJECT, ACK and ERROR transmission blocks are non-data blocks, and have the following format:

SYNC SOH CA TA MTI ETX LPC

12.4.5 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.

12.4.5 E-Codes (cont.)

E-CODE	X' CODE	WRITE (Output)	READ (Input)
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 the START INDEX	Not possible.

12.4.6 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

12.4.7 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 • Text in block which should not have text
3	ERROR response (indicates transmit error)

12.4.7 Error Correction and Load Regulation (cont.)

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 calls break routine, specifying the reason for break (RB) as one of the error counter numbers defined in the table above.

The send break routine will then generate an upline CMD "INPUT STOPPED" or "OUTPUT STOPPED" depending on the stream direction at the time the error occurred. The secondary CMD function code will indicate one of the following reason codes:

Reason for Break (RB)	Description
4	No Response
5	Bad Response
6	Error Response

An error condition caused by terminal malfunction will normally be reported separately on each connection that is active.

If the TIP is unable to aquire sufficient buffers for an input block, any partial block is discarded and the terminal is polled again later. If the host is down the terminal is not polled.

Long Term Error Recovery

After the TIP detects the terminal operation is abnormal and the reason for break code has been sent to the host on all active (delivering data or polling for data) connections, the TIP begins a "failure mode polling" cycle which polls the interactive device at a reduced rate (rate varies depending on number of terminals on a line and system activity). No output will be delivered while in the "failure mode polling" cycle. If the TIP received a good response to a poll, normal operation is restored on the terminal and a CMD "Input Started" PFC=4, SFC=0 will be sent upline on the interactive connection. Any output in queue on the interactive connection would then be delivered.

12.4.8 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.

12.4.9 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 in 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 Introduction to the Mode 4 TIP Flow Charts

The flow charts presented in this section show the general flow of the Mode 4 TIP operation. These diagrams show the block interactions with host and the block interactions with the terminal. Contention resolution between devices on a terminal is also shown.

Table 12-4 represents the flags used to control the general operation of the TIP.

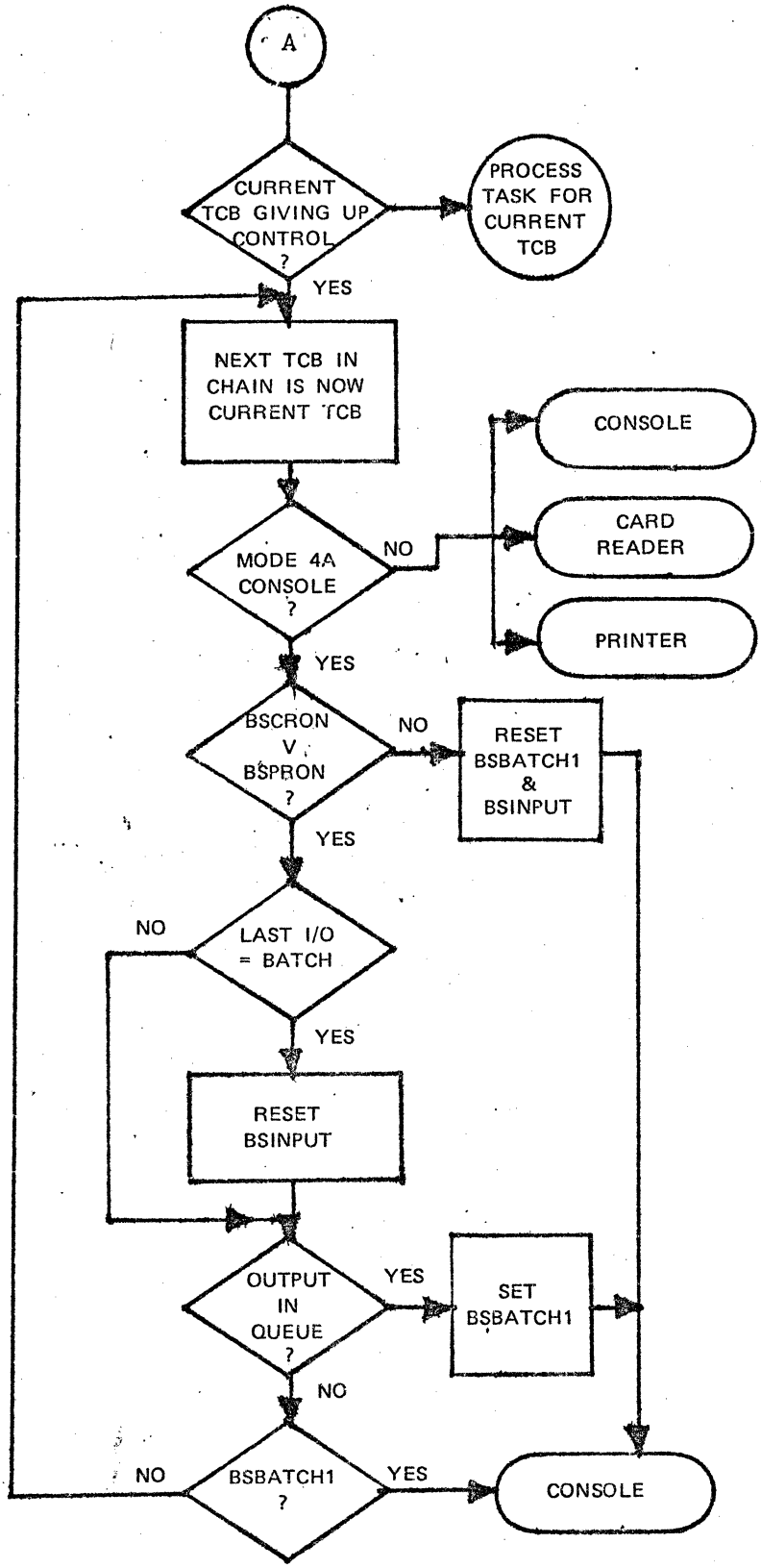
The flowcharts represent the service of a single Mode 4 line and each possible device on the line. Once a transaction has been started with a particular device on a terminal, control will be returned to that terminal and device flow until that transaction sequence is complete.

Table 12-4

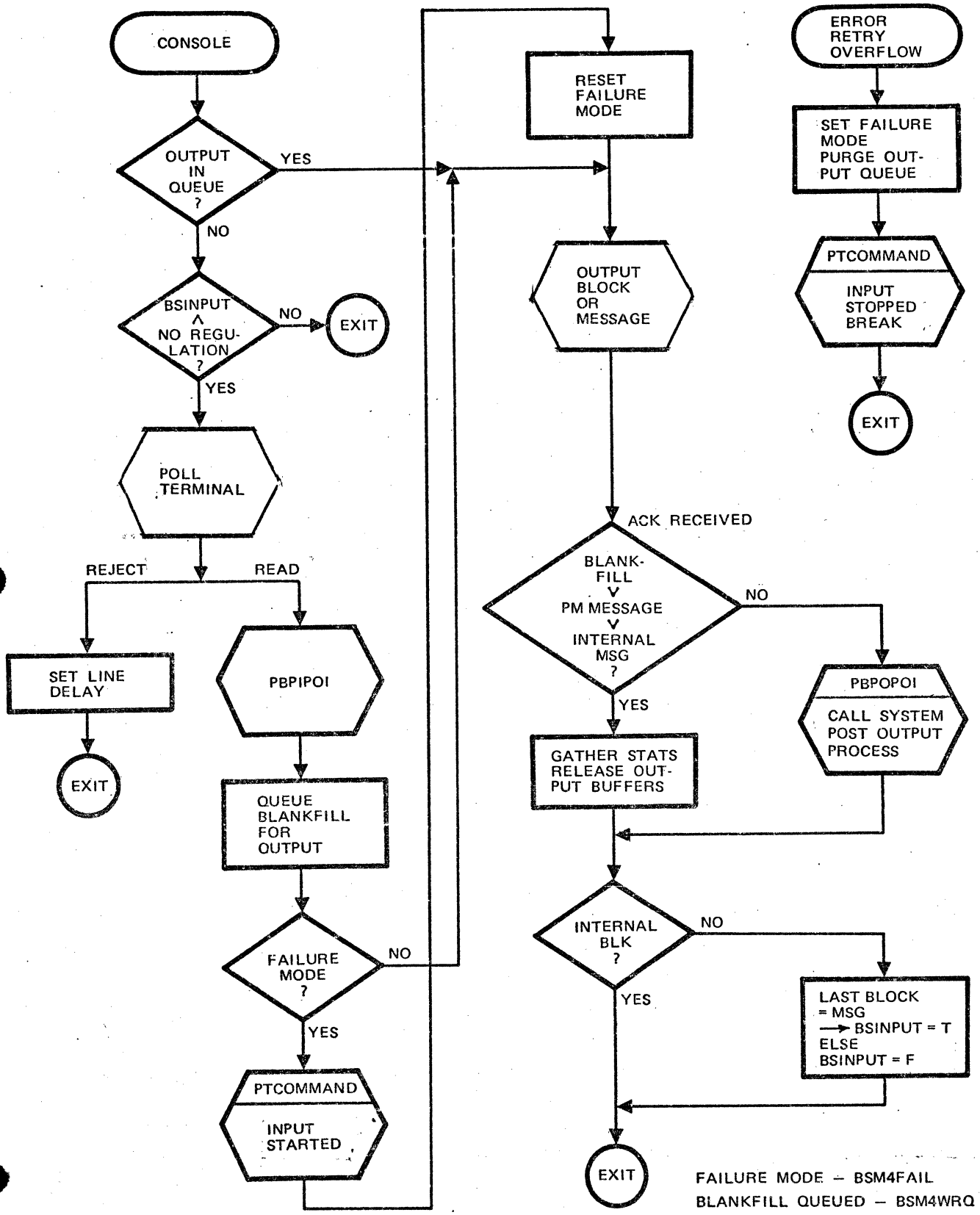
FLAG NAME	DESCRIPTION
Batch Interrupt (BSBATCHI)	USAGE - used to stop batch operations for display input or output. SET 1. Batch Interrupt from operator during Card Read or Print. 2. Output in Q for display device. RESET 1. CMD from host for START Input, Resume or RESTART Output. 2. Initial condition.
Input On (BSINPUT)	USAGE - Display input may be accepted - poll the display. SET 1. After each MSG block output to the display. 2. Initial condition. RESET 1. MSG received from the display. 2. BLK block output to display.
Printer on (BSPRON)	USAGE - Used to detect when output stream to the printer is active. Contention resolution for Mode 4A. SET 1. Output BLK or MSG received from the host. 2. Restart Output CMD received from the host. RESET 1. Stop Output CMD received from the host. 2. Output End of File.

Table 12-4 (cont.)

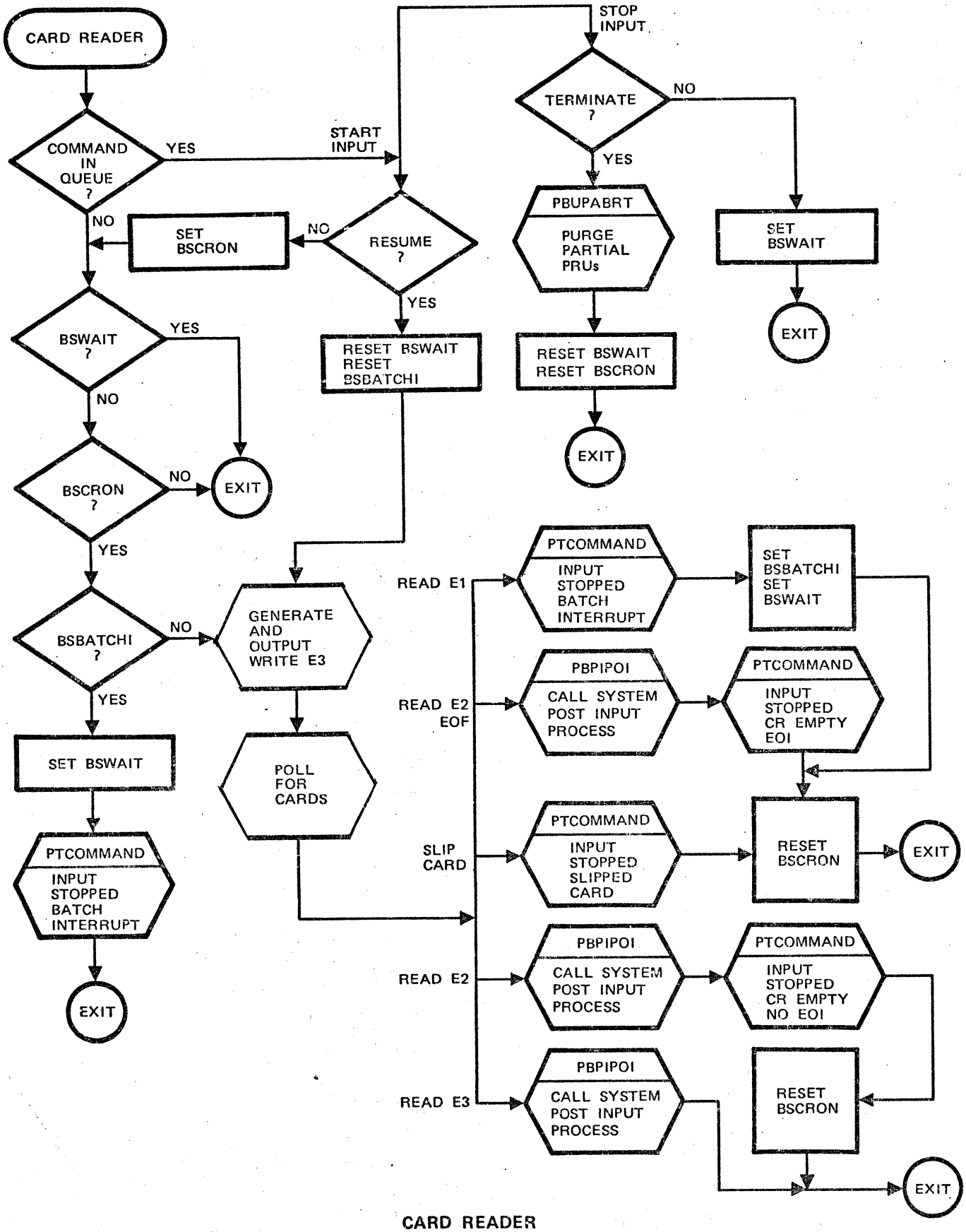
FLAG NAME	DESCRIPTION
Card Reader On (BSCRON)	USAGE - Used to detect when the input stream from the card reader is active. SET 1. Start input received from the host. RESET 1. Input stopped CMD sent to the host. 2. EOI received from the CR. 3. Batch interrupt received from the terminal.
Batch Wait (BSWAIT)	USAGE - Used to temporarily stop batch input or output. SET 1. Stop Input CMD. 2. When Input or Output Stopped, CMD is sent to the host. RESET 1. Stop Input or Output Terminate CMD received from the host. 2. Restart Input or Output CMD from the host.



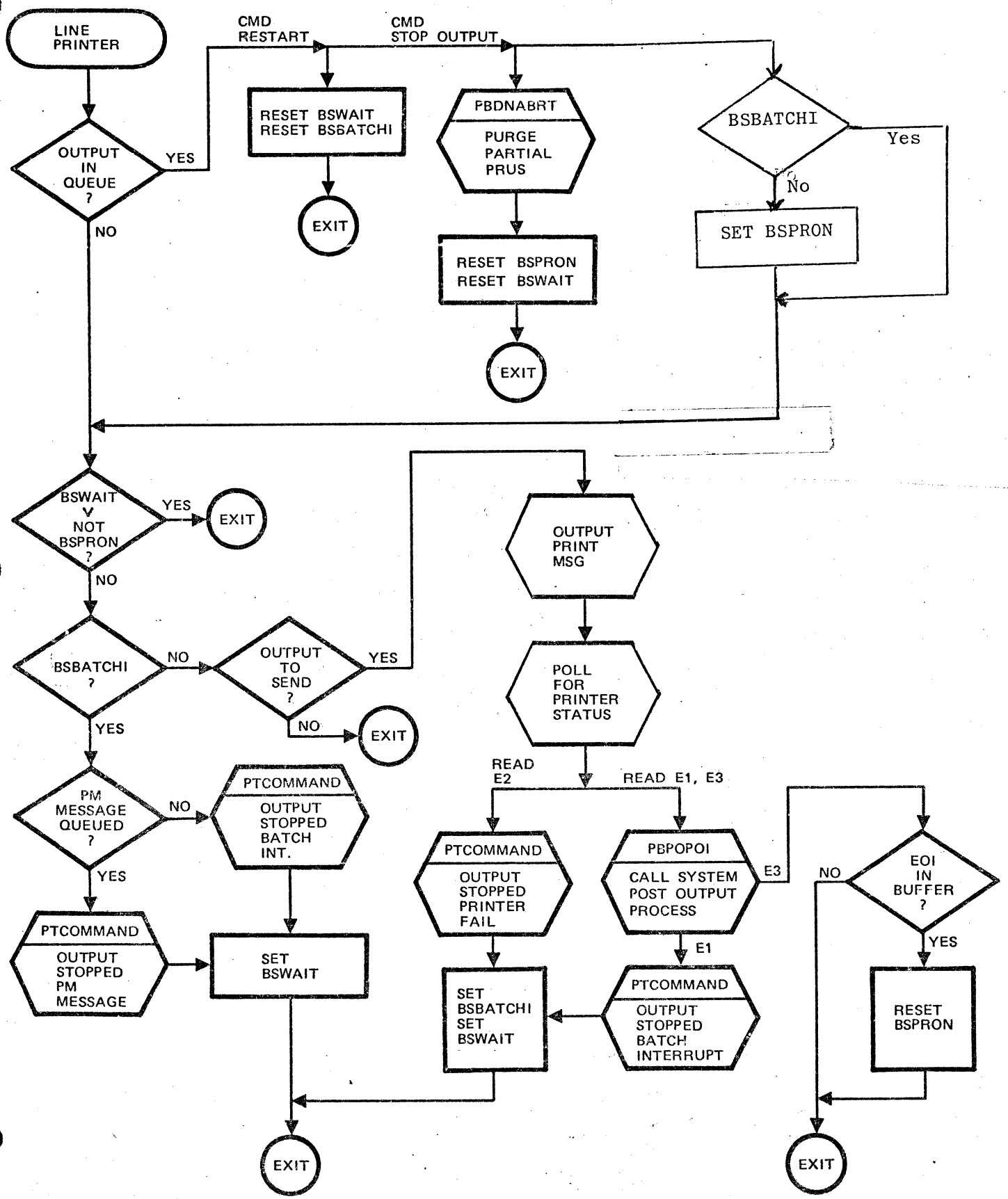
PT4GOTASK
(LINE WORK ALLOCATOR)


CONSOLE
OUTPUT IN QUEUE ?
YES
NO
BSINPUT ^ NO REGULATION ?
NO
EXIT
YES
POLL TERMINAL
REJECT
READ
SET LINE DELAY
EXIT
PBPIPOI
QUEUE BLANKFILL FOR OUTPUT
FAILURE MODE ?
NO
YES
PTCOMMAND
INPUT STARTED
RESET FAILURE MODE
OUTPUT BLOCK OR MESSAGE
ACK RECEIVED
BLANK-FILL v PM MESSAGE v INTERNAL MSG ?
NO
YES
GATHER STATS RELEASE OUT-PUT BUFFERS
INTERNAL BLK ?
NO
YES
LAST BLOCK = MSG → BSINPUT = T
ELSE BSINPUT = F
EXIT
ERROR RETRY OVERFLOW
SET FAILURE MODE PURGE OUT-PUT QUEUE
PTCOMMAND
INPUT STOPPED BREAK
EXIT
PBPOPOI
CALL SYSTEM POST OUTPUT PROCESS
FAILURE MODE – BSM4FAIL
BLANKFILL QUEUED – BSM4WRQ

CONSOLE


CARD READER
COMMAND IN QUEUE ?
YES
NO
START INPUT
SET BSCRON
NO
RESUME ?
YES
STOP INPUT
TERMINATE ?
NO
YES
PBUPABRT
PURGE PARTIAL PRUs
SET BSWAIT
EXIT
BSWAIT ?
YES
NO
RESET BSWAIT RESET BSBATCHI
RESET BSWAIT RESET BSCRON
EXIT
BSCRON ?
NO
EXIT
YES
BSBATCHI ?
NO
YES
GENERATE AND OUTPUT WRITE E3
POLL FOR CARDS
SET BSWAIT
PTCOMMAND
INPUT STOPPED BATCH INTERRUPT
EXIT
READ E1
PTCOMMAND
INPUT STOPPED BATCH INTERRUPT
SET BSBATCHI SET BSWAIT
READ E2 EOF
PBPIPOI
CALL SYSTEM POST INPUT PROCESS
PTCOMMAND
INPUT STOPPED CR EMPTY EOI
SLIP CARD
PTCOMMAND
INPUT STOPPED SLIPPED CARD
RESET BSCRON
EXIT
READ E2
PBPIPOI
CALL SYSTEM POST INPUT PROCESS
PTCOMMAND
INPUT STOPPED CR EMPTY NO EOI
READ E3
PBPIPOI
CALL SYSTEM POST INPUT PROCESS
RESET BSCRON
EXIT

CARD READER



LINE PRINTER

13.0 On-Line Diagnostics

13.1.0 Introduction

This specification describes the Communications Control Intercom 3.0 diagnostic features.

13.2.0 On-Line CLA/Modem Diagnostics

13.2.1 Abstract

Basic CLA and modem loop-back tests will be included in CCP on-line software. These tests do basic data and control turn-arounds and test to determine if all data and status is normal. The On-Line Diagnostics can be controlled from the local 255X console. One or more CLAs can be tested concurrently without impacting services to other lines in the system. Local and remote modem tests can also be run for modems possessing loop-back features.

On-line Diagnostic software is implemented as a CCP resident routine.

Typical operation of running diagnostics on a faulty line follows:

1) Disable the line
2) Send data messages to the NPU to start the diagnostic
3) Receive a diagnostic response
4) Terminate the diagnostic if no errors are found
5) Repeat 2, 3, 4 until the faulty component has been isolated and corrected
6) Enable the line

13.2.2 Description of Tests Performed

CLA Internal Loopback Test

The test consists of a CLA command test which verifies the operation of the CLA related to command function, and a data verification test.

Modem Loopback Test

This test provides for the isolation of a problem further out into the communication system on modems on which the modem loopback feature is available. The test consists of a data verification test with limited analysis of modem control signals.

External Loopback Test

Provides for the loopback of data external to the CLA. The test consists of a command and data verification test whose primary purpose is to verify the operation of the line drivers and receivers. This test can only be run in conjunction with the loopback jumper plug.

13.2.3 Commands Entered Only At the NPU Console

Disable Line (NPU Command)

Parameters

Element name

Purpose

To free the line for diagnostics or other maintenance activities

Stimulus

Sent before an operator starts diagnostics for a line or before beginning maintenance

Action

Causes the TIP to terminate all activity on the specified line

Enable/Configure Line (NPU Command)

Parameters

Element name

Purpose

To return an access line to service

Stimulus

Sent after a diagnostic test and/or maintenance activity is complete

Action

Restores the line to the network. The Enable Line SM will be rejected by the NPU if the line is running diagnostics (see CCP R4 ERS for SM response codes).

13.2.4 Commands Entered at the NPU Console

13.2.4.1 Start Diagnostic Test

Parameters

NPU name
Type of test
Port number
Type of CLA
Modem Class

13.2.4.1 Start Diagnostic Test (cont'd)

Command Format

Overlay Data SM Header	ASCII String . . . (MAX = 22 CHARS)

The ASCII string is:

[INT / I / EXT / E / MOD / M] , XX, Y, ZZ

Type of test: INT = I = CLA Internal Loopback
EXT = E = External Loopback
MOD = M = Modem Loopback

XX = Port Number - expressed as two hexadecimal digits 01 - X'FE
Y = CLA type

0 - Synchronous RS232 (2560-1)
1 - Asynchronous (2561-1)
2 - Synchronous non-RS232 (2560-2, 2560-3)
3 - Synchronous SDLC (2563-1)

ZZ = Modem Class – See Table A-6

Response Format

Overlay Data Normal Response SM Header	ASCII String

The ASCII string is:

PORT XX STARTED
PORT XX INV PORT
PORT XX INV CLA TYPE
PORT XX INV MODE
PORT XX NOT DISABLED
PORT XX TEST IN PROC.
PORT XX LOW BUFFERS
PORT XX INV MOD CLASS

13.2.4.1 Start Diagnostic Test (cont'd)

Purpose

Initiate on-line diagnostic tests

Stimulus

Sent as a result of command entry to start diagnostics

Action

A Start Diagnostic Test Response SM is sent and the operator is notified of the initiation of on-line diagnostics.

Any errors detected during the test result in:

- delivery of a response service message at the originator console with an appropriate error indication
- termination of the test. To reinitiate the test, the operator must re-input the Start Diagnostic Test command.

If no error is detected, the diagnostic will continue to loop until a TERMINATE TEST command is entered.

13.2.4.2 Terminate Test SM

Parameters

Port number

Command Format

Overlay Data SM Header	ASCII String (MAX = 22 CHARS)

The ASCII string is:

[TERM / T] , XX

XX = Port number expressed as two hexadecimal digits 01 - X'FE

Response Format

Overlay Data Normal Response SM Header	ASCII String

13.2.4.2 Terminate Test SM (cont'd)

The ASCII string is:

PORT XX TEST COMPL – NO ERRORS
PORT XX INV PORT
PORT XX NOT IN PROCESS

Purpose

Terminate on-line diagnostic tests

Stimulus

Sent as a result of command entry while a test is in progress

Action

The diagnostic currently in progress is terminated at the end of the normal test cycle. A Terminate Test Response SM is sent.

13.2.4.3 Diagnostic Test Report Response

Parameters

Port number
Error Code
CLA Status Word

Results Format

Overlay Data SM Header	ASCII String

The ASCII string is:

PORT XX ERROR YY, PP, QQQQ, RRRR

Where: XX = Port number expressed as two hexadecimal digits 01 - X'FE
YY = Type of error - see Appendix Table A-5
PP = Subsection number of test being performed by diagnostic program at the time of error detection

QQQQ and RRRR are additional data relative to the type of error received - see Appendix Table A-6. Each field represents a 16 bit word output as 4 hexadecimal digits.

13.2.4.3 Diagnostic Test Report Response (cont'd)

Purpose

Report results on an on-line diagnostic test to the operator

Stimulus

Detection of an abnormal condition during a test run

Action

Detection of an error condition causes a report to be output to the controlling device and the on-line diagnostic program to enter an idle mode which requires a Start Diagnostic Test SM to reactivate the diagnostic program.

13.2.5 Operation From the NPU Console

On-line diagnostics for the CCI 3.0 must be operated from the NPU local console in place of the NOP console. This can be done by inputting commands at the NPU console to direct diagnostic service messages and responses to the NPU console. This operation is limited and must be done in the following sequence:

1) Have the NPU operator disable the line(s) to be tested by changing the CLA address to a non used value or have the Host disable the line.
2) Type in the NPU console a command to accept overlay data messages and to display overlay responses at the NPU console.

Ⓖ /OVL Ⓓ

where Ⓖ means depression of the "CONTROL" key

3) Operate the on-line diagnostics by inputting overlay data messages at the NPU console

Ⓖ [DATA = / DA =] string Ⓓ Ⓖ

"string" information is defined by 13.2.4.1

Responses will appear at the NPU console in "string" format as defined in 13.2.4.1 and 13.2.4.3

4) Return the NPU console to its normal mode. Redirect overlay service messages and response functions to the NOP. This is a command input at the NPU console.

Ⓖ /SUP Ⓓ

5) Have the NPU operator enable any line(s) now deemed operational.

13.3.0 In-Line Error Reporting

13.3.1 Error Log Messages

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 buiid 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 Table A-5.

Purpose

To allow each NPU program to report detected errors to Host.

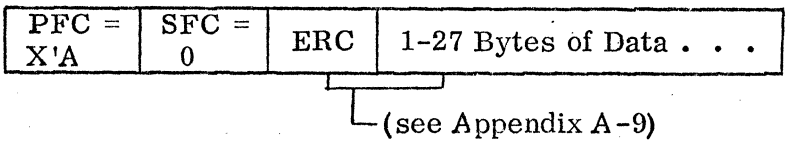
Stimulus

Occurrence of a program-detectable error.

Action

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

Format



Response

None

13.4.0 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 Diagnostic Handbook.

13.5.0 Decision Logic Tables

Diagnostic Decision Logic Tables (DDLT's) will be provided in a Diagnostic Handbook. They will cover failure and alarm conditions in the following areas:

- Load time failures
- In-line diagnostic alarms delivered to the NOP
- Response from On-line diagnostics
- Post Mortem dumps and halt codes
- Printouts of error log messages by the Hardware Performance Analyzer (HPA)

APPENDIX A

SERVICE MESSAGES

AND

TABLE FORMATS

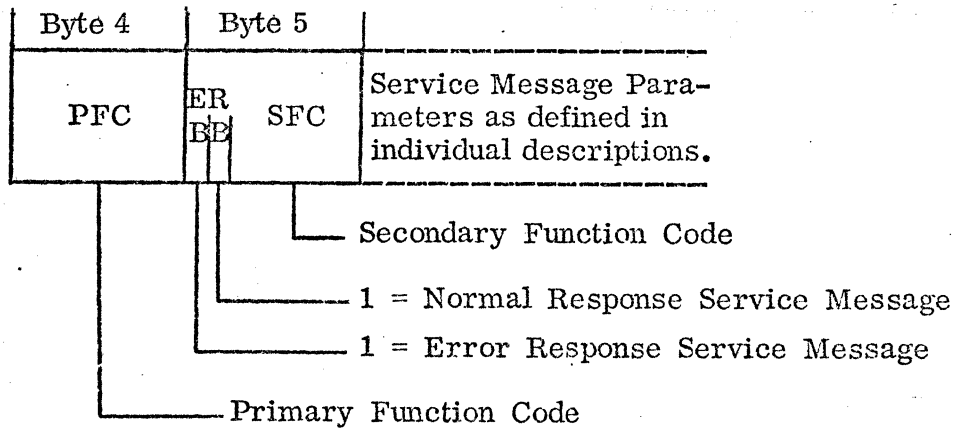
A.0 Service Message General Format

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.

Block Header

Destination Node	Source Node	Connection Number = 0	BSN = 0	BT = 4
Byte 0	Byte 1	Byte 2	Byte 3	

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. The BSN field will always be zero for service messages.



Service Message Parameters

The following table defines abbreviations used in the individual service message descriptions.

Abbreviation	Meaning
ABL	See Network Block Limit (NBL)
BSN	Block Serial Number - part of the block protocol. See Section 3.1.1.
BT	Block Type - SM's are always of type CMD. See also Section 3.1.3.
CA	Cluster Address - part of a terminal's physical identification (see Section 5.3.1).
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 3.1.2.1.)
DT	Device Type - defines the Terminal Class and type of device on a terminal (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).

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. (Previously Allowable Block Limit.)
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 5.3.1).
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 - used to define the TIP and Sub-TIP that services a terminal.

Table A-1

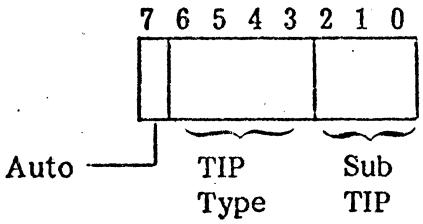
Service Message Summary

Service Message Name	PFC	NPU Mnemonic	SFC	NPU Mnemonic
NPU Initialized	1	D8LOAD	1 2	D9FRC D9INIT
	2			
Configure Line Delete Line Configure Terminal Reconfigure Terminal Delete Terminal	3	D8CONFIG	0 1 2 3 4	D9LNCNF D9LNNDLT D9TMLCNF D9TMLRCNF D9TMLDLT
	4			
	5			
Line Status Request Terminal Status Request Line Count Request	6	D8STATUS	2 3 5	D9LNSTAT D9TMLSTAT
NPU Statistics Line Statistics Terminal Statistics	7	D8COUNTS	0 1 2	D9NPUCNTS D9CNTLN D9CNTML
Enable Line Disable Line Disconnect Line	8	D8LINE	0 1 2	D9ENABLE D9DISABLE D9DISCONNECT
	9			
CE Error	X'A	D8EVENT	0	D9CE
Host Broadcast One Host Broadcast All Operator Message Terminal	X'C	D8USER	0 1 2 3	D9BRD1 D9BRDCST D9OPMSG D9TMCL

INTENTIONALLY BLANK

Table A-2. Terminal Type (TT)/Device Type (DT)

Terminal Type (TT)



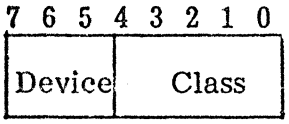
In the Configure Line SM the TT field will be defined as shown above (Terminal Type) using the following values:

Auto = 0 No auto recognition
= 1 Auto recognition performed when line becomes operational.

TIP Type =	0 N/A	1 TTY	2 Mode 4	3 Hasp	4 BSC (2780/3780)
Sub TIP = 0	N/A	ASCII - 110		N/A	N/A
= 1			M4A/BCD		
= 2		ASCII - 150	M4A/ASCII		
= 3		ASCII - 300	M4C		
= 4					

Note 1-Sub TIP is used for upline SM's only.
Note 2-Use TIP Type 4 for auto recognition on HASP, BSC type lines.

Device Type (DT)



Class	Terminals Supported (By Device)				
	0 Console	1 Card Reader	2 Line Printer	3 Card Punch	4 Non-Impact Printer
1	TTY Comp.				
2					
3					
4					
5					
6					
7	2780	2780	2780	2780	
8	3780	3780	3780	3780	
9	HASP	HASP	HASP	HASP	
10	Mode 4	Mode 4	Mode 4		Mode 4

When the DT byte is sent in a downline SM to identify a particular TCB, the TC field need not match field in the TCB as the latter can change at any time.

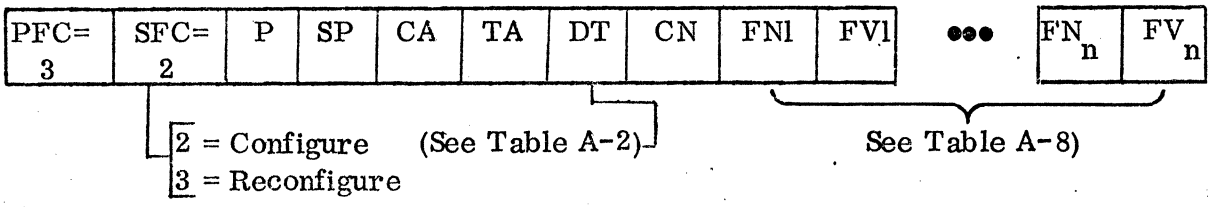
Table A-3

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)	RESERVED →								
(X'B)	RESERVED →								

*Operating with HDX Protocol.

A.3.2
A.3.3 Configure/Reconfigure Terminal



The table below shows the valid CA and TA values for each terminal.

	CA	TA
Mode 4A	X'70-X'7F	X-60
Mode 4C	X'70-X'7F	X'61-X'6F
TTY	0	0
HASP	0	0-7(1)
BSC	0	0-1(2)

(1) Equal to the stream ID of the device. The Interactive console must be 1, Card Reader(s) 1...7, Printer(s) 1...7, Punch(es) 1...7.
(2) Punch only, all other devices are zero.

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
9	Waiting for Ring

TABLE A-5

ON-LINE DIAGNOSTICS

ERROR CODES

ERROR CODE	SIGNIFICANCE	DATA TYPE
0	Test Complete (In Response to Terminate)	N
1	Unsolicited Input Detected	N
2	Unsolicited Output Data Demand Detected	N
3	Input Loop Error	S
4	Output Loop Error	S
5	Parity Error	S
6	Framing Error	S
7	Data Transfer Overrun	S
8	Next Character Not Available	S
9	No CLA Status After CLA Status Was Requested	S
A	Unsolicited CLA Status	S
B	CLA Status Not Cleared After ISON (Input Supervision ON) Was Sent	S
C	No Status After RTS (Request to Send) or ISR (Input Status Request) Was Sent	S
D	No CTS (Clear to Send) After RTS	S
E	No Status After DTR (Data Terminal Ready)	S
F	No DSR (Data Set Ready) After DTR	S
10	No SQD (Signal Quality Detect) After DTR	S
11	No RI (Ring) After DTR	S
12	No Status After SRTS (Secondary Request to Send)	S
13	No SRLSD (Secondary Received Line Signal Detector) After SRTS	S
14	No CLA Status After LM (Local Mode)	S
15	No DCD (Data Carrier Detect) After LM	S
16	Unsolicited Status After OM (Originate Mode)	S
17	No Status or Improper Operation of RI After TB (Terminal Busy)	S
18	No Status After NSYN (New Sync)	S
19	Improper Operation of DCD, RI, or QM (Quality Monitor) or Unsolicited Status After NSYN	S
1A	No RI After RTS	S
1B	Unsolicited Status After LM	S
1C	Input Data Timeout During Data Verification Test	N
1D	Unsolicited Status During Data Verification Test (DVT)	S
1E	Bad CRC's Received During DVT of SDLC CLA	N
	Test Conditions	
1F	DVT Failed (Synchronous CLA Even Parity)	D
20	DVT Failed (Synchronous CLA Odd Parity)	D
21	DVT Failed (Synchronous CLA No Parity)	D
22	DVT Failed (SDLC CLA)	D
23	DVT Failed (SDLC CLA)	D
24	DVT Failed (SDLC CLA)	D

TABLE A-5 (Con't)

ON-LINE DIAGNOSTICS

ERROR CODES

CONTINUED

ERROR CODE	SIGNIFICANCE	DATA TYPE
	Test Conditions	
25	DVT Failed (Asynch CLA, 40 Baud, Even Parity, 1 Stop Bit)	D
26	DVT Failed (Asynch CLA, 85.4 Baud, Odd Parity, 2 Stop Bit)	D
27	DVT Failed (Asynch CLA, 100 Baud, No Parity, 1 Stop Bit)	D
28	DVT Failed (Asynch CLA, 110 Baud, Even Parity, 2 Stop Bit)	D
29	DVT Failed (Asynch CLA, 120 Baud, Odd Parity, 1 Stop Bit)	D
2A	DVT Failed (Asynch CLA, 133.3 Baud, No Parity, 2 Stop Bit)	D
2B	DVT Failed (Asynch CLA, 150 Baud, Even Parity, 1 Stop Bit)	D
2C	DVT Failed (Asynch CLA, 300 Baud, Odd Parity, 2 Stop Bit)	D
2D	DVT Failed (Asynch CLA, 600 Baud, No Parity, 1 Stop Bit)	D
2E	DVT Failed (Asynch CLA, 800 Baud, Even Parity, 2 Stop Bit)	D
2F	DVT Failed (Asynch CLA, 1050 Baud, Odd Parity, 1 Stop Bit)	D
30	DVT Failed (Asynch CLA, 1200 Baud, No Parity, 2 Stop Bit)	D
31	DVT Failed (Asynch CLA, 1600 Baud, Even Parity, 1 Stop Bit)	D
32	DVT Failed (Asynch CLA, 1600 Baud, Odd Parity, 2 Stop Bit)	D
33	DVT Failed (Asynch CLA, 2400 Baud, No Parity, 1 Stop Bit)	D
34	DVT Failed (Asynch CLA, 2400 Baud, Even Parity, 2 Stop Bit)	D
35	DVT Failed (Asynch CLA, 4800 Baud, Odd Parity, 1 Stop Bit)	D
36	DVT Failed (Asynch CLA, 9600 Baud, No Parity, 2 Stop Bit)	D
37	DVT Failed (Asynch CLA, 9600 Baud, Even Parity, 1 Stop Bit)	D

Data types used to clarify on-line diagnostics mode.

DATA TYPE	DATA WORD 1	DATA WORD 2
S	CLA Status expected	CLA status received
D	Data expected	Data received
N	Zero - not applicable	Zero - not applicable

TABLE A-6

Modem Class

The modem class denotes the maximum speed at which the modem is capable of operating and it must be specified according to the following criteria.

Test Mode	CLA Type	Maximum Modem Speed	Modem Class (ZZ)	Modems *
Internal and External Loopback	ALL	Not Applicable	0	NONE
Modem Loopback	2560-1 2560-2 2560-3 2563-1	Not Applicable	1	201B, 201A, 201C, 201D, 208A, 208B 358-2
	2561-1	100	2	
		110	3	
		120	4	
		134.5	5	
		150	6	
		300	7	103 series, 113A, 113B, VA3405-A-G
		600	8	VA 3405 A thru G
		800	9	
		1050	A	
		1200	B	VA 3405A thru G
		1600	D	
		2400	F	
		4800	10	
		9600	12	358-1

* The modems listed constitute only a small fraction of all possible modems offered by a variety of manufacturers.

Table A-7
Line Control Block
Field Number/Field Value (FN/FB) Assignments

Field Number	NPU Mnemonic Name	Description	Mode 4 TIP	ASYNC	HASP	BSC
5	BZOWNER	Node ID of Owning Host	0*	0*	0*	0*
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	BSC
5	BSTCLASS	Terminal Class ***	10	1	9	7-8
12	BSOWNER	Node ID of Owning Host*	0	0	0	0
13	BSCN	Connection Number*	1-255	1-255	1-255	1-255
14	-	Destination Node (NPU)*	1-32	1-32	1-32	1-32
15	-	Source Node (Host)*	0	0	0	0
16	BSNBL	Network Block Limit	1	1	1	1
19	BSIPRI	Input Priority	1-3	1-3	1-3	1-3
28	BSPGWIDTH	Page Width	0-255		0-255	0-255
30	BSXBLKLL	Transmission Block Length Least Significant	0-255		0-255	0-255
31	BSBLKLM	Transmission Block Length Most Significant	0-7		0-7	0-7
32	BS2629	026/029 Code Option	0=29 1=26		0 = 29 1 = 26	0 = 29 1 = 26
33	BSNUMR	Number of Records per Block				2/7
34	BSSUPCC	Suppress Carriage Control	0 = N/S 1 = S		0=N/S 1=S	0=N/S 1=S
35	BSBAN	Banner On/Off	0 = ON 1=OFF		0=ON 1=OFF	0=ON 1=OFF
36	BSEM	"EM" at end of card for short records			0=No EM 1=EM	0=No EM 1=EM
37	BSCODE	TIP Code	1-3**			

NS =No Suppress; S = Suppress

* Required for configure TCB SM (These fields must be ordered 14, 15, 13, 12)

** Required only if not auto recognition on a configure TCB SM, 1= Mode 4A BCD; 2 = Mode 4A ASCII; 3 = Mode 4C

*** Required for "Reconfigure" SM; may not use for configure

Table A-9

CE Error Report Codes

Code	Significance	Text						Reported By
		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	
01	Disconnect of Switched Line ($\overline{DSR}$)	P	SP	S1*	S2*	—	—	PTCLAS
02	Abnormal Operation of DSR or CTS ($\overline{DSR}$ + $\overline{CTS}$)	P	SP	S1	S2	—	—	PTCLAS
03	Abnormal Operation of Data Carrier Detect ($\overline{DCD}$)	P	SP	S1	S2	—	—	PTCLAS
04	Unsolicited Output Data Demand	P	SP			—	—	PMWOLP
05	CLA Address Out of Range	P	SP			—	—	PMWOLP
06	Illegal Loop Cell Format	P	SP			—	—	PMWOLP
07	Unsolicited Input	P	SP			—	—	PMWOLP
08	Input Loop Error	P	SP	S1	S2	—	—	PTCLAS
09	Output Loop Error	P	SP	S1	S2	—	—	PTCLAS
X'0A	Output Data Demand Timeout	P	SP			—	—	PTTER
X'0B	Modem Timeout	P	SP			—	—	PTTER
X'0C	Abnormal Operation of SOCD	P	SP	S1	S2			
X'0D	CLA Status Overflow	P	SP	S1	S2	—	—	PTCLAS
X'0E	Framing Error	P	SP	S1	S2	—	—	PTCLAS
X'0F	Next Character Not Available	P	SP	S1	S2	—	—	PTCLAS
X'10	Data Transfer Overrun	P	SP	S1	S2	—	—	PTCLAS
X'11	MLIA Error Status	00 = Error Condition Restored				—	—	PBMLIA
		01 = Error Counts Given	Input Loop Error Count	Lost Data Count	Alarm Count	—	—	
		02 = MLIA Failure	—	—	—	—	—	

*CLA Status Bytes 1 & 2; see page A-18

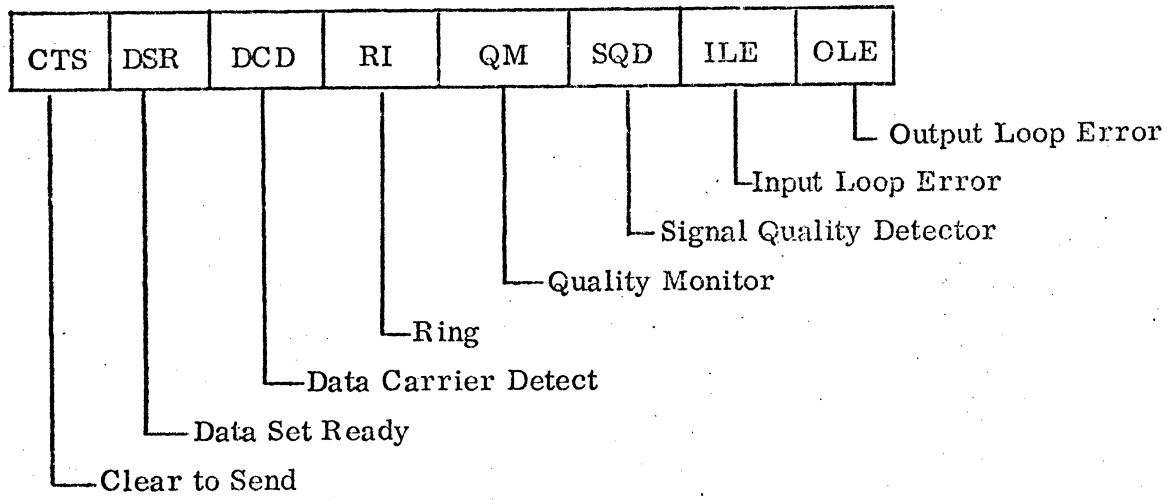
Table A-9 (continued)

X'12	Upline Break — Counter Overflow	P	SP	CA	TA	DT	02 = No Response Overflow 03 = Bad Response Overflow 04 = Error Response Overflow	MD4TIP
X'13	Invalid DBC	DBC Value						TTY
X'19 • • • X'1F	(Reserved)							
X'20	Deadman Timeout	Last State	Current State	—	—	—	—	HIP
X'21	Spurous Interrupt	Coupler-Status-Word*		—	—	—	—	HIP
X'22	Chain Address Zero	Coupler-Status-Word		—	—	—	—	HIP
X'23	Hardware Timeout on Input	Coupler-Status-Word		—	—	—	—	HIP
X'24	Input Data Transfer Terminated by PPU	Coupler-Status-Word		—	—	—	—	HIP
X'25 X'26	Not Used							
X'27	Output Data Transfer Terminated by PPU	Coupler-Status-Word		—	—	—	—	HIP
X'28	Hardware Timeout on Output	Coupler-Status-Word		—	—	—	—	HIP
X'29	EOP Missing	Coupler-Status-Word		—	—	—	—	HIP

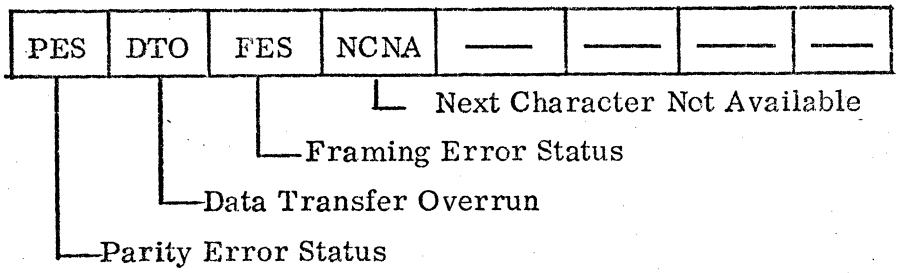
* See Table 7-1 for format

Table A-9 (Con't)

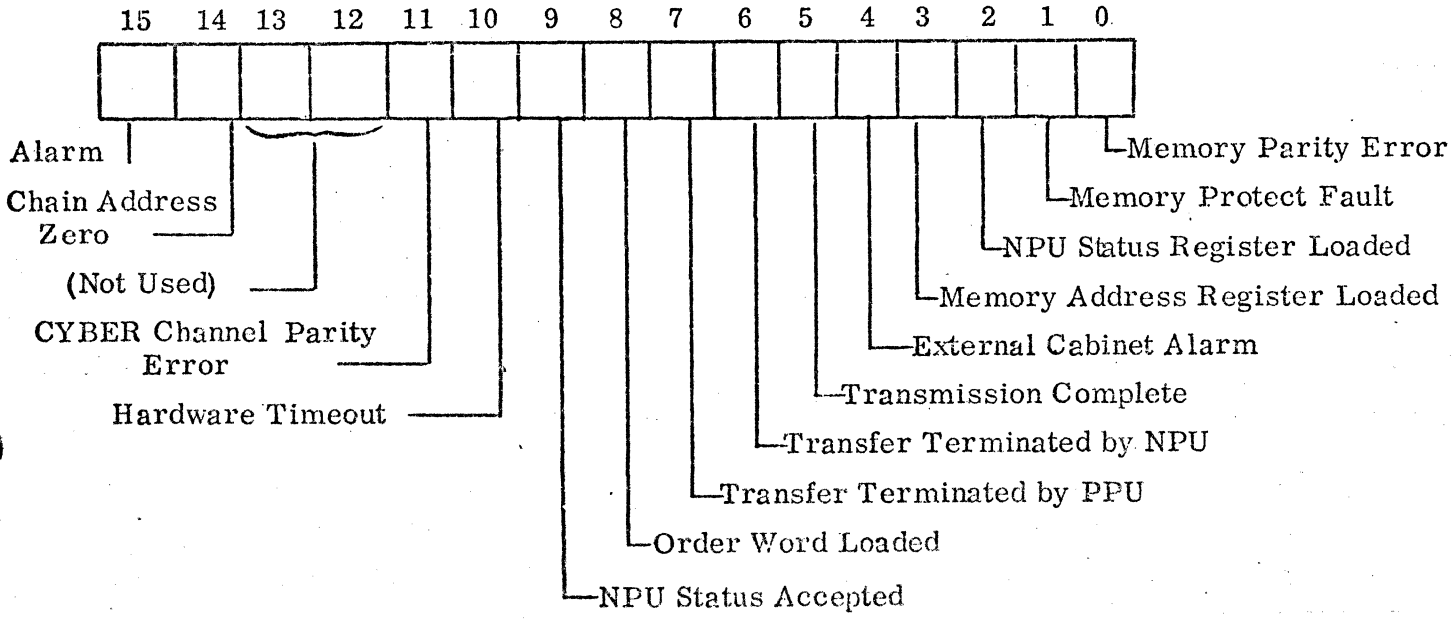
CLA Status Byte 1 (S1)



CLA Status Byte 2 (S2)



Coupler Status Word



Individual Service Messages

A.1.2 NPU Initialized

PFC = 1	SFC = 2	CCP Version	CCP Cycle	CCP Level

Response

NONE

A.3.0 Configure Line

PFC = 3	SFC = 0	P	SP	LT	TT	FN1	FV1	• • •	FN_n	FV_n

(See Table A-7)

(See Table A-2)

(See Table A-3)

Normal Response

PFC = 3	SFC = X'40	P	SP	LT	TT	RC = 0

X'40 = Configure

(See Table A-2)

(See Table A-3)

Error Response

PFC = 3	SFC = X'80	P	SP	LT	TT	RC	FN	FV

(FN/FV Pair returned if RC = 1)

(See Table A-3)

(See Table A-2)

X'80 = Configure

1 = Invalid FN/FV
2 = Invalid Line Number
3 = Line Control Block already configured
4 = Invalid Line Type
5 = Invalid Terminal Type

A.3.1 Delete Line

PFC= 3	SFC= 1	P	SP

Normal Response

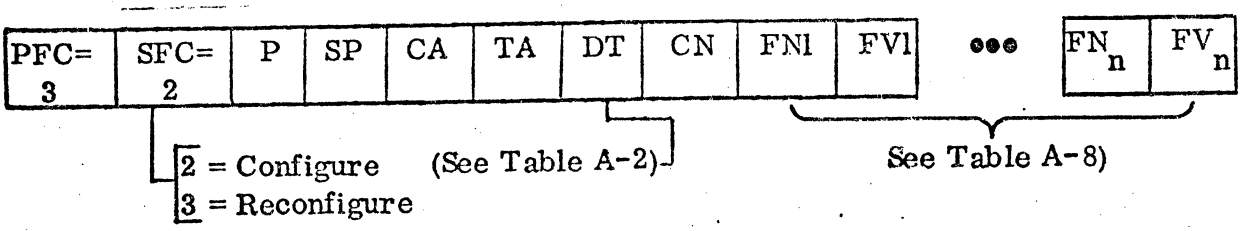
PFC= 3	SFC= X'41	P	SP	RC= 0

Error Response

PFC= 3	SFC= X'81	P	SP	RC=

2 = Invalid Line Number
3 = Line Control Block already deleted

A.3.2
A.3.3 Configure/Reconfigure Terminal



The table below shows the valid CA and TA values for each terminal.

	CA	TA
Mode 4A	X'70-X'7F	X-60
Mode 4C	X'70-X'7F	X'61-X'6F
TTY	0	0
HASP	0	1-7(1)
BSC	0	X'12-X'13(2)

(1) Equal to the stream ID of the device. The Interactive console must be 1, Card Reader(s) 1...7, Printer(s) 1...7, Punch(es) 1...7.

(2) Punch only, all other devices are zero.

Normal Response

PFC = 3	SFC	P	SP	CA	TA	DT	CN	RC = 0

X'42 = Terminal Configured
X'43 = Terminal Reconfigured

DT → See Table A-2

Error Response

PFC = 3	SFC	P	SP	CA	TA	DT	CN	RC	FN	FV

(FN/FV Pair returned if RC = 1 or 9)

SFC:
X'82 = Configure
X'83 = Reconfigure

DT → (See Table A-2)

RC:
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
5 = Invalid Terminal Type
6 = Line Inoperative or not Enabled
(Configure)
8 = Logical Link not established
9 = CN in Use
10 = Console not configured for a Mode 4 Device

A.3.4 Delete Terminal

PFC = 3	SFC = 4	P	SP	CA	TA	DT	CN

DT → (See Table A-2)

Normal Response

PFC = 3	SFC = X'44	P	SP	CA	TA	DT	CN	RC=0

DT → (See Table A-2)

Error Response

PFC = 3	SFC = X'84	P	SP	CA	TA	DT	CN	RC

DT → (See Table A-2

RC:
2 = Invalid Line Number
3 = Terminal not Configured

5.0 On-Line Diagnostics

Start Diagnostic Test Command

PFC = 5	SFC = 0	Overlay ID	1	P	SP	DM	CLA Type	MC

MC: (See Table A-6)

DM:
- 0 = CLA Internal Loopback
- 1 = Modem Loopback
- 2 = External Loopback

CLA Type:
- 0 = 2560-1
- 1 = 2561-1
- 2 = 2560-2 or 2560-3

Response

PFC = 5	SFC = X'40	Overlay ID	1	P	SP	RC

RC:
- 0 = Diagnostic Test Started
- 1 = Invalid Line Number
- 2 = Invalid CLA Type
- 3 = Invalid Test Mode
- 4 = Line not Out-of-Service
- 5 = Test Already in Process
- 6 = System Low on Buffers

A.6.0 Terminate Diagnostic Test Command

PFC = 5	SFC = 0	Overlay ID	2	P	SP

Response

PFC = 5	SFC = X'40	Overlay ID	2	P	SP	RC

RC:
- 0 = Diagnostic Test Terminated
- 1 = Invalid Line Number
- 2 = Diagnostic not in Progress

A.6.1 Diagnostic Test Results

PFC = 5	SFC = 0	Overlay ID	3	P	SP	EC	CLA Hardware Status Word

CLA Hardware Status Word: (See Table A-5)

A.6.2 Line Status Request

PFC = 6	SFC = 2	P	SP

If missing, return status on all lines.

Normal Response

PFC= 6	SFC= X'42	P	SP	RC	LT	CFS	NT

(See Table A-3) (See Table A-4)

0 - Line Operational
RC=4 - Line Inoperative
5 - No Ring Indicator or Auto Recognition in Progress
6 - Stop - CLA not responding, (CE intervention required)

Error Response

PFC= 6	SFC= X'82	P	SP	RC

1 = Invalid Line Number or No Lines Configured belonging to Requestor
2 = Line Status Request in Progress
3 = Illegal Configuration State

Unsolicited Response

Only for auto.-recog.

PFC= 6	SFC= 2	P	SP	RC	LT	CFS	NT	TT	CA

TA_1	DT_1	--	TA_i	DT_i

(See Table A-2)
(See Table A-4)
(See Table A-3)
(same as other Line Status responses)

For Auto Recognition responses, the TA DT pairs are repeated for each terminal that can be detected by the TIP. 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.

A.6.2 Cont.

For auto recognition bewteen BSC and HASP terminals, the TT field will be returned as a 3 or 4 if the terminal is configured for TT=4 with the auto recognition bit set (see Table A-2). The CA field will be zero and no TA, DT fields will be returned.

A.6.3 Terminal Status Request

PFC = 6	SFC = 3	P	SP

Normal Response

PFC = 6	SFC = X'43	P	SP	CA	TA	DT	RC	DN	SN	CN	TOT

(See Table A-2)

0 = operative
4 = terminal inoperative

Error Response

PFC = 6	SFC = X'83	P	SP	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 4 (no regulation)
Word 7 = Times at Regulation Level 3
Word 8 = Times at Regulation Level 2
Word 9 = Times at Regulation Level 1
Word 10 = Times at Regulation Level 0
Word 11 = Network Assurance Protocol Timeout

Response

NONE

A.7.1 Line Statistics

PFC = 7	SFC = 1	P	SP	0	00	Statistics Words 1 - 4

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	CA	TA	DT	CN		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 line

PFC = 8	SFC = 0	P	SP

Normal Response (Line Enabled)

PFC = 8	SFC = X'40	P	SP	RC	LT	CFS	NT=0

(See Table A-4)
(See Table A-3)
See Line Status Request Response Codes

Error Response (Line Not Enabled)

PFC= 8	SFC= X'80	P	SP	RC

(See Line Status Request Response Codes)

A.8.1 Disable Line

PFC= 8	SFC= 1	P	SP

Normal Response (Line Disabled)

PFC= 8	SFC= X'41	P	SP	RC = 0	LT	CFS	NT

(See Table A-4)

(See Table A-3)

Error Response

PFC= 8	SFC= X'81	P	SP	RC

(See Line Status Request Responses)

A.8.2 Disconnect Line

PFC= 8	SFC= 2	P	SP

Normal Response

Normal response is Line Enabled Normal Response SM.

Error Response

PFC = 8	SFC = X'82*	P	SP	RC

*SFC = X'80 for RC≥4

(See Line Status Request Response Codes)

A.A.0 CE Error

PFC = X'A	SFC = 0	ERC		• • •	

1 - 27 Bytes of Data

(See Table A-9)

Response

NONE

A.C.0 Host Broadcast One

PFC =X'C	SFC = 0	P	SP	CA	TA	DT	TEXT

Text must be 1-50 characters.

Normal Response

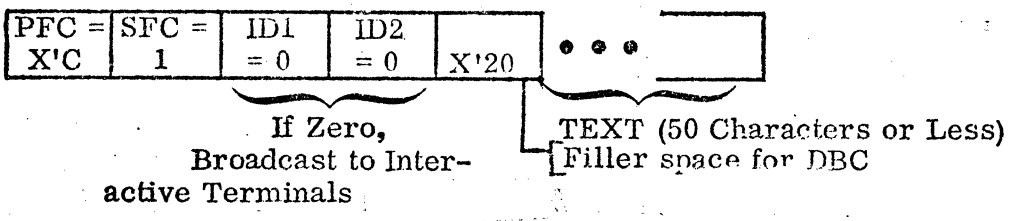
PFC X'C	SFC X'40	P	SP	CA	TA	DT	RC=0

Error Response

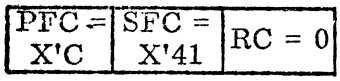
PFC X'C	SFC X'80	P	SP	CA	TA	DT	RC

1 = Invalid Line Number
2 = Invalid Device Type
3 = Terminal Not Configured
4 = Terminal Inoperative
5 = Host Broadcast In Process

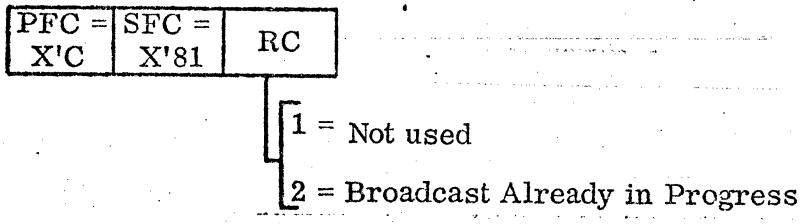
A.C.1 Host Broadcast



Normal Response



Error Response



FEATURE ABSTRACT

RFC0003 - Add 3270 support to REBS or INTERCOM 5.

1. Release Description

The 3270 feature release for REBS or INTERCOM 5 consists of this feature abstract plus a permanent file dump tape containing the files required to install the feature code on the host. Corresponding modifications are required for CCI to run the feature and are contained on a separate release. Three files with ID=RMK are included on the dump tape and described below:

File	Function
MODS3270PL1A	Modifications to CMR port entry definitions to allow definitions of 3270 ports.
MODS3270PL12	Modifications to be used to generate a REBS OLDPL that will contain 3270 code.
MODS3270PL14	Modifications to be used to generate an INTERCOM 5 OLDPL that will contain 3270 code.

2. Installation Procedures

PL1A must be updated and CMR on the deadstart tape must be replaced with a new CMR from the updated PL1A.

It is recommended that the standard PL1A update decks be used and an attach for the modification file plus an update read directive be added to update PL1A.

If the intent is to run 3270 under REBS the REBS installation decks should be modified to attach the modification file for REBS plus an update read directive to generate an updated REBS OLDPL.

If the intent is to run 3270 under INTERCOM 5, the PL14 installation decks should be modified to attach the modification file for PL14 plus an update read directive to generate an updated PL14.

3. Configuration Information

The BISYNC macro has been modified to allow definition of a 3270 port.

3. {Con't}

BISYNC LT=XX,CARR=YYYYY,MODE=3270,CL=C_1,...,C_N},
TA={T_1,...,T_N}

where XX=line type
DU for dial-up {default}
HW for hardwired,

YYYYY=carrier type {hardwired only}
CONTR for controlled carrier {default}
CONST constant carrier,

C_N=list of cluster addresses
{0-11,0=default}
T_N=list of terminal addresses
{1-11,1=default}

For a MODE=AUTO cluster address list and terminal address list may be specified and will be used if the terminal type is reported as 3270. Note - The CCI modifications do not include the ability to auto-recognize a 3270 terminal.

The downline load file name is calculated by assigning a value of 2**4 for the 3270 TIP. The last character of the load file is then calculated as now by interpreting the last digit as an alphabetic character. The load file for only the 3270 is ODG while 3270 and Mode 4 is ODI under this scheme.

The modification deck for PL1A is for use against a PL1A for REBS. To use the deck against a standard PL1A for INTERCOM 5 requires the following update directive changes:

REBS UPDATE DIRECTIVE	INTERCOM 5 UPDATE DIRECTIVE
*DELETE,RBS0034.121	*DELETE,F7220CM.162
*DELETE,RBS0034.124	*DELETE,F7220CM.165
*DELETE,RBS0034.136	*DELETE,F7220CM.179
*INSERT,RBS0034.139	*INSERT,F7220CM.182
*DELETE,RBS0034.142	*DELETE,F7220CM.185
*DELETE,RBS0034.146	*DELETE,F7220CM.195
*DELETE,RBS0034.148	*DELETE,F7220CM.197
*DELETE,RBS0034.159	*DELETE,F7220CM.208
*DELETE,RBS0034.162	*DELETE,F7220CM.211

3270 TIP ERS

By

R. C. Naken

H. J. Bots

April 26, 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 dedicated data links.

2.0 FEATURES

- 3270 control unit supporting
  - 3277 displays
  - 3284, 3286, 3288 printers
- 3275 control unit (no dial-feature)
- EBCDIC code set
- BSC protocol
- multidrop support
- up to 12 clusters on a line where each cluster may have a maximum of
  - 11 consoles,
  - 8 line printers.
- data protect is not supported

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.

4.0 NORMAL TRANSACTION CONTROL

Under normal conditions the TIP will issue a general poll to a cluster and then, if needed, issue selects or specific polls to all devices on that cluster.

The invitation to input or output will rotate around all the cluster controllers on the line.

The input process will terminate when the cluster controller indicates it has no more traffic.

5.0 OPERATIONAL OVERVIEW

The TIP will control activity on a 3270 by polling for input and delivering output.

The 3270 station is under control of the TIP in any 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 test mode. In test mode, the station is either the master or slave station with the TIP assuming the opposite role.

When the entry into test 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.

EBCDIC requires standard BSC CRC.

7.0 TRANSMISSION_BLOCKS

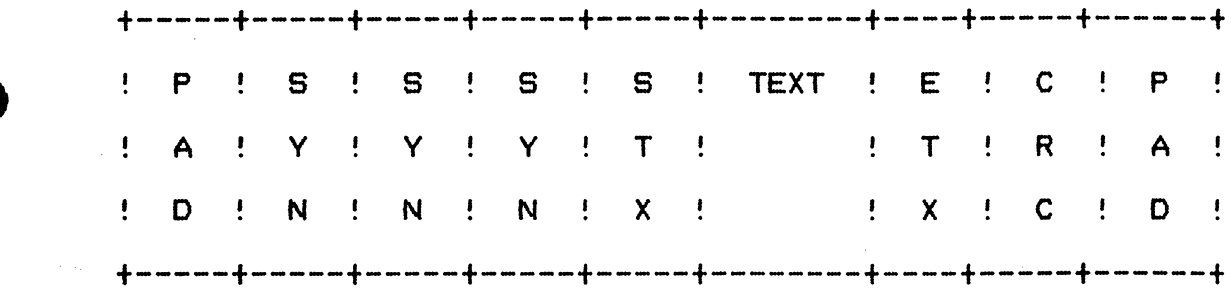
7.1 INPUT

PAD	SYN	SYN	STX	CU	DA	TEXT	ETB or ETX	CRC	PAD

CU = cluster unit poll address

DA = device address

7.2 OUTPUT

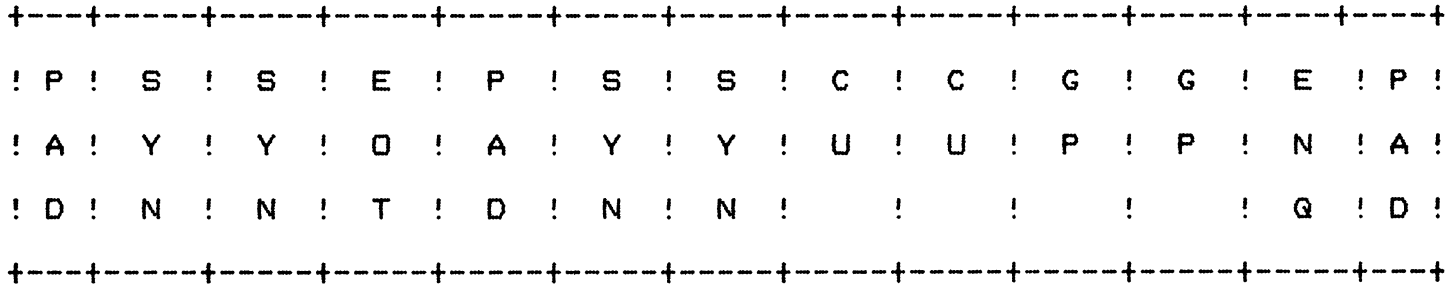


PAD	SYN	SYN	SYN	SOH	TEXT	ETX	CRC	PAD

- 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	ESC	PAD	SYN	SYN	CUU	CUU	GP	GP	ENQ	PAD

- first 4 characters clears any cluster controller from text mode.
- GP = general poll code in place of DA for general polls (x '22' is EBCDIC).

8.0.1.1 General Poll/Poll Responses

- Device Response

Any device having an input requirement may be 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 a random device and input all device messages, sequentially, as long as "ACK"s are received to blocks until all device have been serviced. The first block of each input message identifies the responding device.

The TIP will send blocks terminated by "ETB"s as "BLOCK" blocks; terminated by "ETX" as "MESSAGE" blocks to the host.

- No 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

P	S	S	E	P	S	S	C	C	D	D	E	P
A	Y	Y	O	A	Y	U	U	U	A	A	N	A
D	N	N	T	D	N	N					Q	D

8.0.2.1 Response to Selection

. ACKO

The device has entered the test 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.

- RVI

  The device has pending status or is unavailable. The TIP terminates the select sequence and polls the device to obtain status.

- Timeout or Bad Resposne

  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.

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.

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 roll 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 responds with an "EOT". The TIP performs a specific poll to obtain status.

- EOT

  The 3270 sends an "EOT" sequence to end a normal data transfer sequence. The TIP enters control mode and performs the next in turn task for the line.

- Bad Blocks

The TIP "NAK"s bad blocks. The 3270 does not count "NAK"s. The TIP must count "NAK"s and respond as the same as for general polls.

- 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.

10.0 UNUSED BSC FEATURES

- TTD
- Forward Abort
- Conversational Mode
- 3275 dial-feature

11.0 EXCEPTION_HANDLING

The 3270 has self diagnosis 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	%	RST	SCU	DAS	SS0	SS1	ECR	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. In addition to this primary use of status messages, the TIP will be required to detect end of busy condition for a printer.

The busy condition was 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 SLOW_POLL

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, 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, 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.

- Variable transmission blocks
- Variable page/line (screen) sizes
- Page wait
- PM messages support
- 125 character ASCii support to line printer devices
- Console interface is 64 character ASCii (Display Code)
- Configuration

  12 clusters

  11 consoles

  8 line printers
- There must always be a minimum of 1 console per cluster.
- Line printer status messages go to the lowest numbered console.
- Batch connections are PRU oriented e.g., display code.

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 of 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.

If a Selection Addressing sequence detects that the 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.
- A command attempted to start a printer but found it not ready. The printout is suppressed.

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 during a printout. 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 a printer character generator error occurred, the printer became mechanically disabled, or 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 3171 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).

SOFTWARE

SPECIFICATION

Document Type: External Reference Specification

Product Name: LOW & MEDIUM SPEED BATCH TERM./6000/3000 MODE IV - 200 UT PROTOCOL (731-12, 732-12)

Product Number: I065*1

This document is a working paper only and does not represent any official intent on the part of CONTROL DATA CORPORATION. Consequently, it is to be used for reference purposes by CONTROL DATA personnel only. The document or information contained therein may not be distributed to clients. Sales commitments are not to be made on the basis of this document.

REVISION STATUS

Change No.	SCN No.	Date
Original		3/24/72
A		6/09/72

II

CONTROL DATA CORPORATION

DESIGN DOCUMENT AUTHORIZATION

DIVISION: Terminal Systems Operations

P.O.D. NUMBER:

- [ ] PROD. DVLPMT. AUTH.
- [ ] GEN. INT. DESIGN
- [ ] OTHER
- [ ] PROD. OBJ.
- [X] EXT. REF. SPEC.
- [ ] GEN. EXT. DESIGN
- [ ] INT. MAIN. SPEC.

RECEIVED
JUN 09 1972
IDD ARDEN HILLS

DISCLAIMER: THIS DOCUMENT IS A WORKING PAPER ONLY AND DOES NOT NECESSARILY REPRESENT ANY OFFICIAL INTENT ON THE PART OF CONTROL DATA CORPORATION.

PRODUCT NUMBER/NOMENCLATURE: 1065*1.0
Low and Medium Speed Batch Terminal/6000/3000
Mode IV - 200 UT Protocol {731-12, 732-12}

APPROVAL SIGNATURES		DATE
PREPARED BY. W. C. Collette	William C Collette	5/30/72
PROJECT MANAGER W. C. Collette	William C Collette	5/30/72
SUPERVISOR J. W. Kueck	J W Kueck	6/1/72
DEPARTMENT MANAGER J. C. Kellogg	James C Kellogg	6/9/72
DIRECTOR - SOFTWARE M. J. McHale	M J McHale	6/9/72
- HARDWARE M. J. McHale	M J McHale	6/9/72
GENERAL MANAGER		
OTHER		

REVISIONS

LTR	DESCRIPTION	INITIAL	DATE
A	Changes for enhancements	mcc	5/30/72

ALL CHANGES MARKED By A BAR IN MARGIN.

	PAGE	REV.

AA3974

PRINTED

TABLE OF CONTENTS

1.0 SCOPE

This specification describes the external characteristics of the software that resides in the Low/Medium Speed Batch Terminal. This software employes the 200 USER Terminal Link Control procedure, thus allowing the Low/Medium Speed Batch Terminal to communicate with a 6000 or 3000 central computer employing 200 USER Terminal Link Control.

The Low/Medium Speed Batch Terminal hardware is a standard Low or Medium Speed Batch Terminal with an additional 4K of 1.1 micro-second memory and without a functional card punch or cyclic Encoder. It consists of the following devices:

1. Card Reader
2. Line Printer
3. Display
4. Keyboard
5. Micro Drum

It communicates, via a telephone line, with a 6000 or 3000 Computer System employing 200 USER Terminal Software.

1.1 Functional Description

The major functions of the PL10/200 UT software are:

a. Interpret messages transmitted by the high level computer.
b. Transmit message in response to messages received.
c. Direct output data to the correct device.
d. Accumulate data from input devices.
e. Transmit data to high level computer.
f. Check and generate character and message parity.
g. Check for correct control codes on messages received.
h. Attach correct control codes to messages transmitted.
i. Detect errors such as:

    1. Parity errors
    2. Unrecognizable control codes
    3. Non-existent station address
    4. Write message not ending with escape code and E code

j. Transmit error message in response to errors detected.

2.0 APPLICABLE DOCUMENTS

The following documents of exact issue shown, form a part of this specification to the extent specified herein:

2.1 L/MSBT Hardware

2.1.1 Buffer Controller {FR113}
Engineering Specification No. 53275100

2.1.2 Buffer Controller 750 Nanosecond
4096 word, 18 bit memory
Engineering Specification No. 53611200

2.1.3 Buffer Controller Normal Channel
Engineering Specification No. 53332300

2.1.4 Buffer Controller Block Transfer Channel
Engineering Specification No.

2.1.5 Buffer Controller Maintenance Console
Engineering Specification No. 58018900

2.1.6 Micro Drum Unit
Engineering Specification No. 64013400

2.1.7 Keyboard Unit
Engineering Specification No. 51769800

2.1.8 Display Unit
Engineering Specification No. 16000800

2.1.9 Remote Batch Terminal Low/Medium Speed
Engineering Specification No. 1600900

2.2 MSBT Peripheral

2.2.1 HR600 Line Printer
Engineering Specification No. 53125200

2.2.2 410 Card Reader
Engineering Specification No. 16006900

2.3 LSBT Peripherals

2.3.1 HR300 Line Printer
Engineering Specification No. 53125500

2.3.2 Low Speed Batch Card Reader - MARK II
Engineering Specification No. 16004600

2.4 200 USER Terminal

Hardware Reference Manual - Publication No. 82128000
Revision B-0-1

2.5 IMSBT

Interim Medium Speed Batch Terminal POD Number 131,
Product Number I045*1.

Low/Medium Speed Batch Terminal (L/MSBT) POD No. 253
Product Number I070*1.0.

3.0 DEFINITION OF TERMS

The following is a list of terms used throughout this document and their definitions:

1. Write E1 - A Write message directed to the Display
2. Write E2 - A Write message directed to the Line Printer
3. Write E3 - A Write message directed to the Card Reader
4. Read E1 - A Read message containing an E1
5. Read E2 - A Read message containing an E2
6. Read E3 - A Read message containing an E3
7. Line Turnaround - Switch the communication line from receive mode to transmit mode
8. MSBT - Medium Speed Batch Terminal
9. LSBT - Low Speed Batch Terminal
10. DSA - Data Set Adapter
11. LCDH - Low Cost Display Head

4.0 HARDWARE CONFIGURATION

Figure 4.0-1 shows a Low Speed Batch Terminal. The principle hardware elements are:

a. Buffer Controller - A small stored program computer used as a programmable control element in the terminal.

b. Memory - 8K, 1.1 microsecond, word size 16 bits.

c. Normal Channels - Interface to I/O devices and data set adapter.

d. Data Set Adapter - Acts as an interface between a data set (Modem) and Normal Channels. During input operation the DSA assembles the serial bit stream furnished by the data set, into 8-bit characters. During output operation the reverse disassembly operation occurs.

e. MARK II Card Reader - The Card Reader allows the user to submit jobs, in the form of Hollerith-coded punch cards.

f. HR300 Line Printer - The line printer allows the user to generate a hard copy output.

g. Keyboard/Display - The keyboard/display allows operator communication with the high level computer.

The Medium Speed Batch Terminal consists of an upgrade of peripherals.

a. Buffer Controller - A small stored program computer used as a programmable control element in the terminal.

b. Memory - 8K, 1.1 microsecond, word size, 16 bits.

c. Normal Channels - Interface to I/O devices and data set adapter.

d. Data Set Adapter - Acts as an interface between a data set (Modem) and Normal Channels. During input operation the DSA assembles the serial bit stream furnished by the data set, into 8-bit characters. During output operation the reverse disassembly operation occurs.

e. 410 Card Reader - The Card Reader allows the user to submit jobs, in the form of Hollerith-coded punch cards.

f. HR600 Line Printer - The line printer allows the user to generate a hard copy output.

g. Keyboard/Display - The keyboard/display allows operator communication with the high level computer.

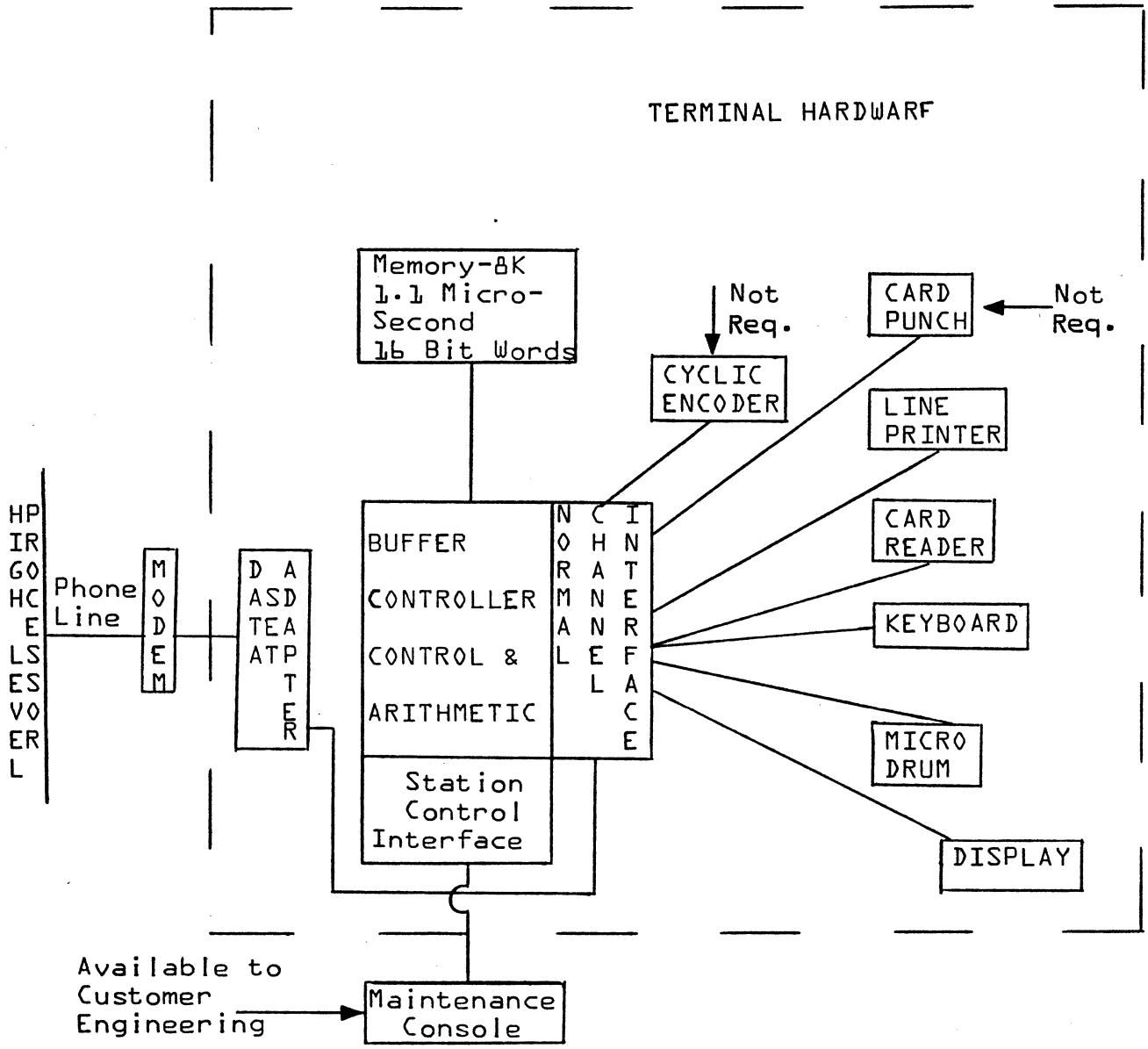


Figure 4.0-1: PL10/200 USER Terminal Configuration

5.0 HIGH LEVEL PROCESSOR INTERFACE

The interface to the Higher Level Processor is accomplished via a data set adapter, data set, and a telephone line. The general message format is shown in Figure 5.0-1.

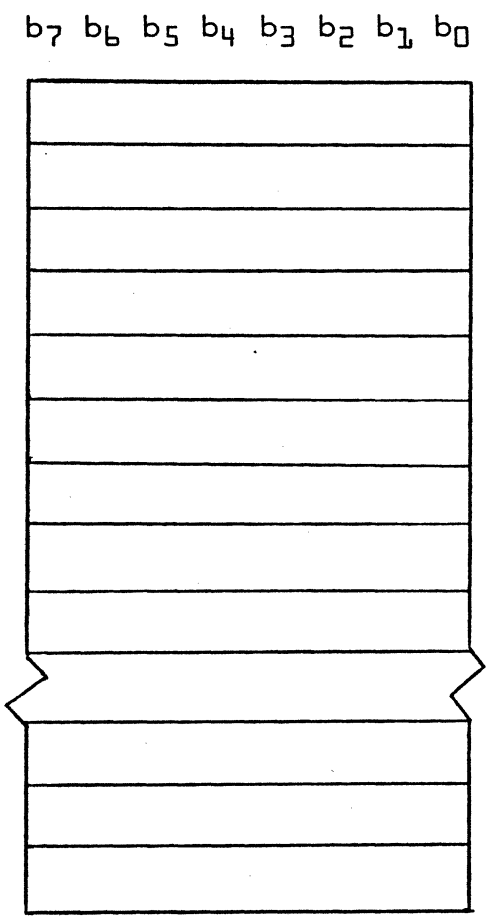


Figure 5.0-1: General Message Format

5.1 Message Format

All messages transmitted are preceded by four SYNC codes to assure synchronization recovery on the receiving end. The start of header code informs the receiving device that the following two codes are addresses. Site and station addresses follow in that order. The site address designates the remote site to which the message is addressed or from which a message is received. Since the terminal is a single-station device, the station address has little meaning and is defined mainly for purposes of program compatibility with multi-station devices. The address is normally 141 or 161 octal for this terminal and has special applications in the realm of write message and responses.

The control code defines the purpose of the message. In the case of messages for printout and display, the control code specifies conditions under which the write is to be performed. Read messages originating at the Display Station Keyboard or card reader are indicated as such by the control code. All other messages are used in the selection-response realm.

The USASCII end of text code designates the previous word as the last word of the message. A message parity word follows. Message parity is applicable from the start of header code through the USASCII end of text. With odd parity on each word and a final message parity word, vertical and longitudinal redundancy checking is provided.

Specific messages sent and received by the remote terminal are shown in Figures 5.1-1 and 5.1-2, respectively. The 7-bit octal translations are also listed.

MESSAGE	7-BIT CODE	FORMAT
Acknowledge	026	Sync
	⋮	⋮ 4 total
	026	Sync
	001	Start of Header
	16X	Site Address
	141 or 161	Station Address
	006	Acknowledge
	003	USASCII End of Text
	XXX	Message Parity
Reject	026	Sync
	⋮	⋮ 4 total
	026	Sync
	001	Start of Header
	16X	Site Address
	141 or 161*	Station Address
	030	Reject
	003	USASCII End of Text
	XXX	Message Parity
Error	026	Sync
		4 total
	026	Sync
	001	Start of Header
	16X	Site Address
	141 or 161	Station Address
	025	Error
	003	USASCII End of Text
	XXX	Message Parity
Read	026	Sync
	⋮	⋮ 4 total
	026	Sync
	001	Start of Header
	16X	Site Address
	141 or 161	Station Address
	023	Read
	XXX	Data
		0 to 1040 with 80 by
	XXX	13 format}
	076	Escape
	102, 040, 041	E1, E2, or E3
	003	USASCII End of Text
	XXX	Message Parity

* 140 or 160 in response to a poll.

Figure 5.1-1: Messages Sent by Terminal

MESSAGE	7-BIT CODE	FORMAT
Poll	026	Sync
		4 total
	026	Sync
	001	Start of Header
	16X	Site Address
	140 or 160	Station Address
	005	Poll
	003	USASCII End of Text
	XXX	Message Parity
Alert	026	Sync
		4 total
	026	Sync
	001	Start of Header
	16X	Site Address
	141 or 161	Station Address
	007	Alert
	003	USASCII End of Text
	XXX	Message Parity
Write	026	Sync
		4 total
	026	Sync
	001	Start of Header
	16X	Site Address
	141 or 161	Station Address
	021	Write
	XXX	Data
		{0 to 1040 with 80 by
	XXX	13 format}
	076	Escape
	102, 040, or 041	E1, E2, or E3
	003	USASCII End of Text
	XXX	Message Parity
Reset-Write	026	Sync
		4 total
	026	Sync
	001	Start of Header
	16X	Site Address
	141 or 161	Station Address
	014	Reset-Write
		{0 to 1040 with 80 by
	XXX	13 format}

Figure 5.1-2: Messages Received by Terminal

MESSAGE	7-BIT CODE	FORMAT
	076	Escape
	102, 040, or 041	E1, E2, or E3
	003	USASCII End of Text
	XXX	Message Parity
Clear-Write	026	Sync
		4 total
	026	Sync
	001	Start of Header
	16X	Site Address
	141 or 161	Station Address
	022	Clear
	XXX	Data
	XXX	{0 to 1040 with 80 by 13 format}
	076	Escape
	102, 040, or 041	E1, E2, or E3
	003	USASCII End of Text
	XXX	Message Parity
Diagnostic Write	026	Sync
		4 total
	026	Sync
	001	Start of Header
	16X	Site Address
	140, 141, 160, or 161	Station Address
	020	Diagnostic Write
	XXX	Data
	XXX	{0 to 1040 with 80 by 13 format}
	076	Escape
	102, 040, or 041	E1, E2, or E3
	003	USASCII End of Text
	XXX	Message Parity

Figure 5.1-2: Messages Received by Terminal {concluded}

5.2 Message Processing

The equipment is normally in the receive mode. Data is input, via the normal channels. After each character is received, it is examined. At least two consecutive sync codes followed by the start of header code and correct site address must be observed to allow further progress. If this does not occur, the Low and Medium Speed Batch Terminal again searches for the same pattern. Once synchronization has occurred, the Message Sequence continues. An incorrect sequence results in a message aborting error sequence. A correct station address allows interpretation of the subsequent control code. If the terminal attempts to transmit a message and it is unsuccessful, it returns to the receive mode.

5.2.1 Poll Message

Since a LSBT or MSBT can be one of many remote sites, each site is scanned in search for read messages. The poll message provides the scanning means. Each remote site receives a poll message. If no read message is available at the terminal a reject message is transmitted. If a read message is available it is transmitted as a reply to the poll message.

A read message is available if one or more of the following conditions {Read Requests} exist.

1. The Line Printer has acknowledged receipt of a message and the printer is not ready.

2. The Keyboard has detected the ETX Key {has been depressed} and it has read the displayed message from the drum into a buffer.

3. The Line Printer has acknowledged receipt of a message and the printer is ready.

4. The Terminal has received a Write E-3.

If condition 1 above exists upon receipt of a Poll message a Read E-2 without data will be transmitted and the condition will be cleared.

If condition 2 above exists upon receipt of a Poll message a Read E-1 containing the keyboard message will be transmitted and the condition will be cleared.

If condition 3 above exists upon receipt of a Poll message a Read E-3 without data will be transmitted and the condition will be cleared.

If condition 4 above exists upon receipt of a Poll message and

1. The Card Reader Interface has read a block of 12 cards and there are more cards to read, a Read E-3 containing the card data will be transmitted.

2. The Card Reader Interface has not read any cards or has found the Card Reader to be not ready a Read E-2 without data will be transmitted.

3. The Card Reader Interface has read some cards and has detected a card read error or a hopper empty status a Read E-2 with the card data will be transmitted.

The condition will be cleared in all three cases.

If conditions 1 and 2 above exist upon receipt of a Poll Message a Read E-2 without data will be transmitted. Condition 1 will be cleared and condition 2 will be saved for a succeeding Poll message.

If conditions 2 and 3 above exist upon receipt of a Poll message a Read E-1 containing the keyboard message will be transmitted. Both conditions, 2 and 3 will be cleared.

If conditions 2 and 4 above exist upon receipt of a Poll message a Read E-1 containing the keyboard message will be transmitted. Condition 2 will be cleared and condition 4 will be saved for a succeeding Poll message.

5.2.2 Alert Message

The purpose of the alert message is to provide a method of informing the terminal that the data source has a write message for transmission. When the terminal receives a correct alert message, the alert will be automatically answered by software and an acknowledge message is transmitted. Upon receipt of a poll message, a read message is transmitted.

5.2.3 Acknowledge Message

Upon correct receipt of a write, reset-write, clear-write or alert message an acknowledge message is transmitted.

5.2.4 Reject Message

A reject message is transmitted by the terminal when one of the following conditions exist:

a) When the terminal is polled and no read request is pending

b) Upon receipt of a write message and a buffer is not available from the buffer pool.

5.2.5 Read Message

A read message may consist of data from the display station keyboard, punched cards read by the card reader, or data from a diagnostic write. It may also be a non-data message indicating the state of the line printer ready status or indicating the card reader is not ready. A read message is transmitted only in response to a poll message or a diagnostic write message.

The read messages that are transmitted in response to a poll message are explained in Section 5.2.1.

A read message containing the same data received is transmitted in response to diagnostic write.

5.2.6 Write Message

Information from the data source to be displayed or printed is sent in the form of write, reset-write, clear-write, or diagnostic write messages. The write message directed to the display will cause the display to begin displaying the message at the current position of the cursor. The reset-write message directed to the display will cause the cursor to be reset to the upper left corner of the display, and the display will begin displaying the message at that point.

The data following the cursor after the message is displayed will not be altered. The clear-write message directed to the display will cause the cursor to be reset to the upper left corner of the display and the display will begin displaying the message at that point. The data following the cursor after the message is displayed will be cleared.

The diagnostic write message resets the entry marker but does not clear the CRT. The data in the message will be written on the drum {for display}. Upon completion of writing it on the drum it will be read from the drum and transmitted to the Central System in a read message. Data from the beginning of the diagnostic write message up to the first ▽E▽ code is returned in the read response. If no ▽E▽ code exists, the read response continues to the end of the display and terminates.

Station addresses used in write message to a particular site should alternate between 141 and 161 octal. For example, if the first write message to the site contained a station address of 141 octal, the next write message to that site should contain a station address of 161. This restriction enables the terminal to remember and report the accuracy of the preceding write message. All messages transmitted from the terminal

following correct receipt of a Write message will contain the same station address as received in the Write message until another correct Write message is received.

Escape codes followed by ▽E▽ codes also serve controlling functions in write, reset-write, and clear-write messages. However, only the first of any ▽E▽ codes is treated as normal data and stored in memory.

The following is a list of write messages and their description:

1. Write E-1 - Upon completion of a Write E-1 message, the message is given to the Display Interface.

2. Write E-2 - Upon completion of a Write E-2 message, the message is given to the Line Printer Interface.

3. Write E-3 - Upon completion of a Write E-3 message, a flag is set so when a Poll message is received the terminal will look for a read request from the card reader interface.

Three other control code types may be present in any write message preceded by an escape code. These are the end of line, carriage return, and compression codes. Only carriage return performs an operation of the display page.

The other two have meaning to the line printer only and are functional in all write messages with the exception of the diagnostic write. The carriage return code erases remaining data to the right of the cursor, and advances the cursor to the next line.

The write message normally solicits a response from the equipment in the form of an acknowledge message. However, should the terminal be busy at the time, a reject message is transmitted instead. Any error conditions result in an error sequence described later. As stated before, a diagnostic write message obtains a read message response.

Messages processed by the line printer differ somewhat from other messages. It has line truncation, format control, and character compression features which are governed by control codes in the message.

A. Line Truncation

Print lines can be truncated through the use of an escape code followed by an end of line or carriage return code. These two codes enable the rest of the current line to be filled with blanks.

B. Format Control

The first data subset code of a message is recognized as a format control code and is not printed. The first code following an end of line or carriage return control code is also interpreted as a format control code. The four format control codes are shown in table 5.6-1.

TABLE 5.6-1

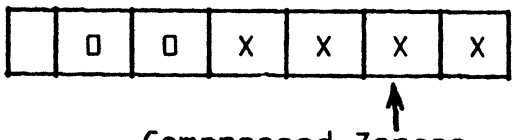
FORMAT CONTROL

EXT BCD	SYMBOL		FUNCTION
120	Blank		Single Space {Vertical}
112	0		Double Space {Vertical}
101	1		Page eject - advance to beginning of next form
060	+		Suppress space - maintain current line {do not space}

Any codes, other than those shown, are interpreted as blanks; i.e., the single space operation is performed.

C. Character Compression

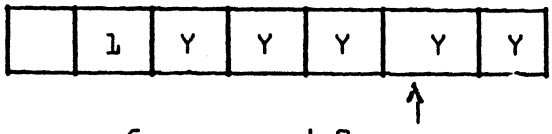
The line printer allows data source compression of strings of zeroes or spaces. This means that the data source need only send an escape code followed by one other code to enable printout of a specified number of zeroes or spaces in succession. The data compression code is interpreted as shown in Figure 5.2.1.



Compressed Zeroes

XXXX {Binary} = number of compressed zeroes

3 ≤ number of compressed zeroes ≤ 17 {octal}



Compressed Spaces

YYYYY {Binary} = number of compressed spaces

3 ≤ number of compressed zeroes ≤ 37 {octal}

See Appendix A.

Figure 5.2.1

5.2.7 Error Message

The following is a list of error conditions that cause an error message to be generated:

1) Word Parity Error on Characteristics Beginning with Station Address through MPC - An error message is transmitted in place of an acknowledge message. No data is transferred to the printer or from the card reader.

2) Message Parity Error - treated same as word parity error.

3) Unrecognized Control Code - Transmit error message on end of message. No data is transferred to the printer or from card reader.

4) Nonexistent Station Address - Treated same as a word parity error.

5) Write Message not ending with Escape Code and E1, E2, or E3 - Treated same as word parity error.

Error conditions arising in diagnostic write messages are treated the same as those listed above with the following exceptions:

a) A word parity, on characters beginning with first data characters through MPC, does not enable transmission of an error message.

   Instead, a parity error code is substituted in the subsequent read message.

b) Missing °E° codes or incorrect message parity do not result in error message transmission. Instead, data is transmitted in the read message exactly as it is received from the data source, up to end of page if there is no °E° code.

6.0 TERMINAL OPERATION

6.1 Control Panel

The following switches are on the control panel:

1. Power On Switch

   A momentary switch which controls power to the buffer controller, display, the micro drum, terminal console and cooling systems. This switch also performs the function of a lamp test switch. When the switch is depressed all lamps will be turned on.

2. Power Off Switch

   A momentary switch to turn off the above equipment.

3. Master Clear Switch

   A momentary switch which causes the BC to stop operation, clears all addressable registers and its controls, as well as clearing memory registers and controls, all normal channels and coupler controls.

4. Channel Clear Switch

   A momentary action switch which, in its activated position, releases clear on all channels.

5. Run Switch

   A momentary switch which causes the BC to begin executing instructions, starting with the instruction stored at the location contained in the program address register.

6. Stop Switch

   A momentary switch which causes the BC to stop executing the stored program.

7. Stop on Memory Parity Fault

   An alternate action switch which, in its activated position, causes the buffer controller to stop when a memory parity fault is detected. An indicator is illuminated when this switch is in the activated position.

8. Reset Memory Parity Fault Switch

A momentary switch which causes the memory parity error indication to be cleared.

9. Autoload Switch

A momentary switch which causes the BC, under hardware control, to load one track of data from the micro drum into core storage starting at location zero.

10. Reset Drum Parity

A momentary switch that clears the drum parity error indication.

6.2 Line Printer

The Line Printer will print all correctly received Write, Reset-Write, or Clear-Write messages containing an Escape Code followed by an E2 code.

The START/STOP key in the Line Printer must be in the START position at all times when printing. The Printer may be stopped by depressing the STOP key.

The Page Eject key is used to move the paper to top of form.

6.3 Card Reader

The Card Reader will read only Hollerith Card Decks like the 200 User Terminal.

To operate the MSBT Card Reader, load the cards into the input hopper and put the START/STOP key in the START position. To operate the LSBT Card Reader, load the cards into the input hopper, depress Clear Key and depress the Offline Key.

6.4 Keyboard/Display

Messages will be displayed on the display as a result of the following operations:

1. A Write, Reset-Write, or Clear-Write message is received from the High Level Processor containing an Escape followed by an E1 code.

2. A diagnostic write is received from the High Level Processor.

3. A message is keyed in on the keyboard.

The keyboard is locked-out when a message from the High Level Processor is being displayed. At all other times, characters can be keyed-in on the keyboard and they will be displayed. However, if the terminal receives a message for the display while a message is being keyed-in on the keyboard, the keyboard will be locked-out and the message from the High Level Processor will be displayed. When the terminal operator is keying-in a message, a buffer from the buffer pool is not required. A buffer is only assigned to the keyboard when the ETX key is depressed. The following Function Key descriptions further explain the operation of the keyboard:

1. Clear - The Clear Key clears the complete display and resets the cursor to the upper corner of the screen.

2. Line Clear - The Line Clear Key clears the line that is currently displaying the cursor and resets the cursor to the beginning of that line.

3. Reset - The Reset Key resets the cursor to the upper left corner of the display without destroying the data.

4. ETX - The ETX Key causes the terminal to get a buffer from the buffer pool. The data is then read from the low cost display head memory, converted to External BCD and stored in the buffer. Upon completion of storing the displayed data in the buffer, a read request is generated.

5. Alert - The Alert will automatically be answered by the software.

6. New Line - The New Line Key clears the current line from the cursor to the right side of the screen and moves the cursor to the left most position of the next line. Depression of the New Line Key does not cause a character to be displayed.

7. Right Arrow - The Right Arrow Key moves the cursor one character position to the right without changing the data.

8. Left Arrow - The Left Arrow Key moves the cursor one character position to the left without changing the data.

9. Up Arrow - The Up Arrow Key moves the cursor up one line without changing the character position or the data.

10. Down Arrow - The Down Arrow Key moves the cursor down one line without changing the character position or the data.

11. Shift - When two symbols share a key, depression of the Shift Key enables the upper symbols. The Shift Key has no affect on single keys.

12. Space - The Space Key moves the cursor right one character position and clears that position.

13. F0 - The F0 Key resets the cursor to the upper corner of the display. This Key is used to simulate the Interrupt Key on the 200 UT.

14. F7 - The F7 Key will only be used when the terminal is in line mode. It will move the active mark down one with the exception of line 13.

15. F6 - The F6 Key initiates a display to print operation. The displayed data is read and formatted in a buffer so the Line Printer Interface routine treats the buffer like a buffer from the Central System.

16. F5 - F1 - Not Used.

6.5 Off-Line Card to Print

The Card to Print operation is strictly off-line and, therefore, requires that the Central System be aware of it. It performs by executing the following steps:

1. Depress Autoload
2. Depress the F4 Key
3. Depress the Clear Key
4. Depress the F2 Key
5. Insert cards in the reader
6. Depress the Clear Key on the reader for LSBT or depress the Start Key for MSBT
7. For LSBT the ON-LINE key must be depressed on the card reader.

7.0 SYSTEM FLOW

There are two basic elements to this Software package, a buffer pool and a scanner. The buffers in the buffer pool are used to accumulate data from the communication line for the output devices and to accumulate data from the input devices to be transmitted to the High Level Processor. The scanner is used to distribute control to the various routines. See Figure 7.0-1.

7.1 Buffer Pool

There is a pool of three buffers. The pointers to these buffers are contained in the table COMBUF. Bits 0, 1, and 2 in BSTAT indicate the status of the buffers. If Bit 0 in BSTAT is clear, it indicates the buffer identified by Word 0 in COMBUF is in use. If Bit 0 in BSTAT is set, it indicates the buffer identified by Word 0 in COMBUF is available for use. Bits 1 and 2 correspond to Word 1 and 2 in COMBUF as Bit 0 does to Word 0 described above.

A buffer is taken from the pool and the corresponding Bit in BSTAT is cleared under the following conditions.

1. When the Line Input receives a Write, Reset-Write, Clear-Write or a Diagnostic Write code. If a buffer is not available, a Reject message is the response to the Write message rather than an Acknowledge message.

2. When the Keyboard Interface detects the Send Key. If a buffer is not available, the Keyboard Interface waits until one is made available.

3. When the Card Reader Interface gains control from the Card Reader Driver. If a buffer is not available, the Card Reader Driver waits until one is made available.

A buffer is returned to the pool and the corresponding bit in BSTAT is set under the following conditions:

1. The Line Printer Interface has completed printing a message.

2. The Display Interface has completed displaying a message.

3. The Line Output has completed transmitting a Read message from one of the buffers.

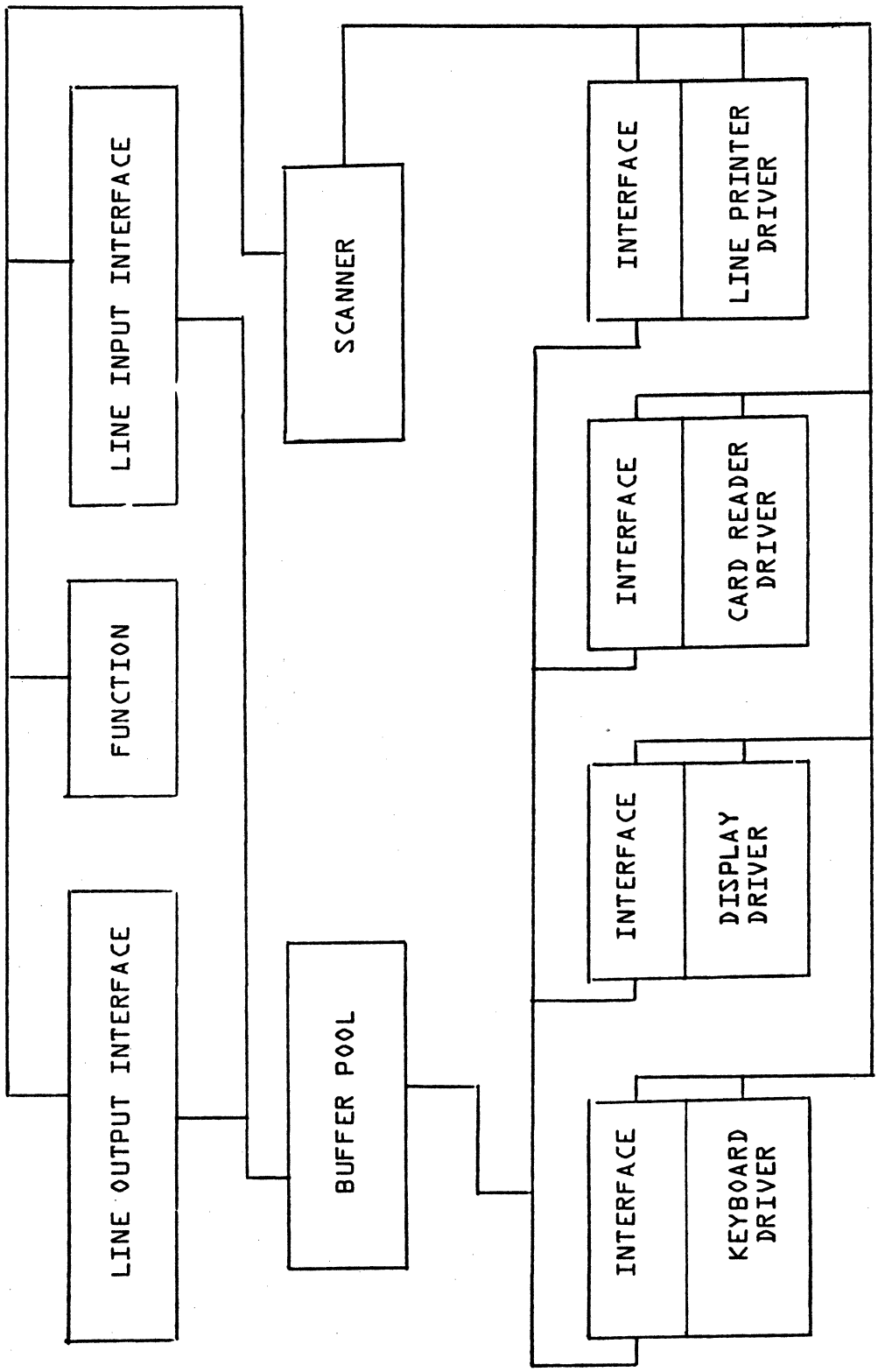


Figure 7.0-1: Block Diagram of Software

7.2 Scanner

The Scanner consists of three parts; an Address Table, a Bit Table, and a Scan Routine. The Scanner has three levels, each having one third of the Address Table and one-third of the Bit Table.

The Address Table is a table of the entry point-addresses of the software routines.

The Bit Table contains a bit position for each address in Address Table.

The Scan Routine scans the Bit Table in search of a bit present. A bit is present in scan level one and scan level three if the corresponding bit is set on the normal channel, clear in the Toggle Mask and Set in the Product Mask or the corresponding bit is clear on the normal channel, set in the Toggle Mask and set in the Product Mask. A bit is present in scan level two if the corresponding bit is set in the Toggle Mask and Set in the Product Mask.

Some of the bits in the Bit Table represent hardware status lines and the remainder of the bits in the Bit Table represent software status. There are two ways bits get set in the Bit Table - a) hardware status line gets set, or b) software routine sets a bit.

Each time the Scan Routine gains control it scans the Bit Table until it finds a bit present. The position within the Bit Table of the bit detected provides an index into the Address Table. The Scan Routine then gives control to the address found at that position in the Address Table.

The following routines have a bit position in the Bit Table and an entry-point address in the Address Table:

1. Line Input Interface - Bit 5, Scan Level 1
2. Line Output Interface - Bit 6, Scan Level 1
3. Function - Bit 4, Scan Level 2
4. Line Printer Driver Interface - Bit A, Scan Level 2
5. Card Reader Driver Interface - Bit 8, Scan Level 2
6. Keyboard Driver Interface - Bit B, Scan Level 2
7. Display Driver Interface - Bit C, Scan Level 2

Each of the devices have drivers that utilize bit positions in the Bit Table. The routines within a driver set and clear each others bits in the Bit Table to perform their respective I/O. However, for the purpose of this document each driver will be considered one routine with one bit position in the Bit Table.

7.2.1 Routine Linkages

7.2.1.1 Line Input Interface

The Line Input Interface routine will get control from the scanner when its bit is set in the product mask, clear in the toggle mask, and its bit is present on the normal channel. The bit in the product mask is set by the Initialize routine. The bit on the normal channel is set by the Data Set Adapter when it has a character assembled for input. The Line Input Interface routine clears its own bit in the product mask when it completes an input operation. The toggle mask bit is clear at all times.

7.2.1.2 Line Output Interface

The Line Output Interface routine will get control from the scanner when its bit is set in the product mask, clear in the toggle mask, and its bit is present on the normal channel. The bit in the product mask is set by the Function routine upon exit. The bit on the normal channel is set by the Data Set Adapter when it is ready to accept a character. The Line Output Interface routine clears its own bit in the product mask when it completes an output operation. The toggle mask bit is clear at all times.

7.2.1.3 Function

The Function routine will get control from the scanner when its bit is set in the Toggle Mask and Product Mask. The bit in the Toggle Mask is set by the Line Input Interface routine upon correct receipt of complete message. The Function routine clears its own bit in the Toggle Mask when it completes turning the Communication line around. The Product Mask is set at all times.

7.2.1.4 Line Printer Driver Interface

The Line Printer Driver Interface routine will get control from the Scanner when its bit is set in the Toggle Mask and Product Mask. The bit in the Toggle Mask is set by the Line Input Interface routine upon correct receipt of a message for the Printer. Then upon preparing a line of data for the Line Printer the Line Input Interface routine sets a bit for the Line Printer Driver routine in the Toggle Mask and clears its bit in the Toggle Mask. Upon completion of printing the line of data the Line Printer Driver routine sets the bit in the Toggle Mask for the Line Printer Driver Interface routine and clears its bit in the Toggle Mask. This continues back and forth until the complete message is printed at which time the Line Printer Driver Interface routine looks to see if there is another message to be printed. If there is another message to be printed the Line Printer Driver Interface routine clears its bit in the Toggle Mask and does not set any other bits. The Product Mask is set at all times.

7.2.1.5 Card Reader Driver Interface

The Card Reader Driver Interface routine will get control from the Scanner when its bit is set in the Toggle Mask and Product Mask. The bit in the Toggle Mask is set by the Card Reader Driver routine after reading a card. The Card Reader Driver Interface routine puts the card data into a buffer from the buffer pool. Then the Card Reader Driver Interface routine sets the bit for the Card Reader Driver routine in the Toggle Mask and clears its bit in the Toggle Mask. This continues back and forth until 12 cards are read or until an error is detected at which time the Card Reader Driver Interface routine clears its bit in the Toggle Mask and does not set any other bits. The Product Mask is set at all times.

7.2.1.6 Keyboard Driver Interface

The Keyboard Driver Interface routine will get control from the Scanner when its bit is set in the Toggle Mask and Product Mask. The bit in the Toggle Mask is set by the Keyboard Driver Interface routine upon detection of a function key. After the Keyboard Driver Interface routine performs the function it clears its bit in the Toggle Mask. The Product Mask is set at all times.

7.2.1.7 Display Driver Interface

The Display Driver Interface routines will get control from the Scanner when its bit is set in the Toggle Mask and Product Mask. The bit in the Toggle Mask is set by the Line Input Interface routine upon correct receipt of a message for the display. Then upon preparing some data to be displayed the Display Driver Interface routine will set the bit for the Display Driver in the Toggle Mask. Upon completion of displaying the data the Display Driver routine clears its bits in the Toggle Mask. This continues back and forth until the complete message is displayed at which time the Display Driver Interface routine clears its bit in the Toggle Mask and does not set any other bits. The Product Mask is set at all times.

7.2.2 Routine Functions

7.2.2.1 Line Input Interface

The Line Input Interface routine performs the following functions:

1. Inputs each character from the Data Set Adapter
2. Checks Character Parity
3. Checks Message Parity
4. Checks for Proper Sequence of Characters
5. Obtains buffers from the buffer pool to accumulate incoming data
6. Converts incoming characters, EXT BCD, to ANSII See Appendix A

7. Sets up branch to correct output message in the Line Output Interface Routine

8. Combines Escape codes and following character into one character for output device

9. Informs the Line Printer Driver Interface or Display Driver Interface which buffer has the message and the length of the message

7.2.2.2 Line Output Interface

The Line Output Interface routine performs the following functions:

1. Outputs each character to the Data Set Adapter

2. Generates Character Parity

3. Generates Message Parity

4. Generates proper sequence of characters

5. Returns buffers to the buffer pool after outputing data

7.2.2.3 Function

The Function routine turns the communication line around (issues the Write function to the Data Set Adapter which puts it into a transmit mode) each time it is entered when the terminal is in 2-wire mode. The Function routine turns the communication line the first time it is entered after initialization when the terminal is in 4-wire mode.

7.2.2.4 Line Printer Driver Interface

The Line Printer Driver Interface routine performs the following functions:

1. Removes the data from the buffer passed to it by the Line Input Interface routine and builds a line buffer for the Line Printer Driver.

2. Returns the buffer to the buffer pool upon completion of printing the message.

3. Generates read requests for the Poll message (See 5.2.1)

7.2.2.5 Card Reader Driver Interface

The Card Reader Driver Interface performs the following functions:

1. Obtains a buffer from the buffer pool
2. Removes the data from the card buffer filled by the Card Reader Driver
3. Converts the data, as it moves it in {2} above, from Hollerith code to EXT BCD
4. Informs the Line Output Interface which buffer has the message and the length of the message
5. Informs the Line Input Interface the status of the Card Reader. See 5.2.1 {4}

7.2.2.6 Keyboard Driver Interface

The Keyboard Driver Interface routine performs the following functions:

1. Executes the keyboard functions as described in Section 7.4.
2. Converts data from ANSII to EXT BCD. {See Appendix A}
3. Informs the Line Output Interface routine which buffer has the message and the length of the message.

7.2.2.7 Display Driver Interface

The Display Driver Interface routine performs the following functions:

1. Removes the data from the buffer passed to it by the Line Input Interface routine, formats it for the display and passes it to the Display Driver.
2. Returns the buffer to the buffer pool upon completion of displaying the message.

APPENDIX A

CODE SETS

LOW AND MEDIUM SPEED BATCH TERMINAL

CODE SETS

Communication Line Codes (Octal)	Card Reader Punches	Line Printer and Display Characters
061	12-1	A
062	12-2	B
063	12-3	C
064	12-4	D
065	12-5	E
066	12-6	F
067	12-7	G
070	12-8	H
071	12-9	I
041	11-1	J
042	11-2	K
043	11-3	L
044	11-4	M
045	11-5	N
046	11-6	O
047	11-7	P
050	11-8	Q
051	11-9	R
122	0-2	S
123	0-3	T
124	0-4	U
125	0-5	V
126	0-6	W
127	0-7	X
130	0-8	Y
131	0-9	Z
113	8-3	=
114	8-4	#
115	8-5	"
116	8-6	%
117	8-7	¢
120	Blank	Space
121	0-1	/
132	0-8-2	}
133	0-8-3	`
134	0-8-4	(
135	0-8-5	@

136	0-8-6	&
040	11	~
137	0-8-7	
052	11-0	\
053	11-8-3	$
054	11-8-4	*
055	11-8-5	!
056	11-8-6	?
057	11-8-7	>
060	12	+
072	12-0	<
073	12-8-3	.
074	12-8-4	)
075	12-8-5	/
None	None	None
077	12-8-7	;
100	8-2	:
112	0	0
101	1	1
102	2	2
103	3	3
104	4	4
105	5	5
106	6	6
107	7	7
110	8	8
111	9	9
120	None	_*
127	7-8-9	None
126	6-7-8-9	None

* Displayed for Parity Error on Diagnosic Writes.

FUNCTION CODES

Communication Line Codes	Function Codes
120	End of Line
101	Carriage Return
102	E1
076	Escape
040	E2
041	E3
127	End of Record
126	End of File
103	Compress 3 Zeros
104	Compress 4 Zeros
105	Compress 5 Zeros
106	Compress 6 Zeros
107	Compress 7 Zeros
110	Compress 8 Zeros
111	Compress 9 Zeros
112	Compress 10 Zeros
113	Compress 11 Zeros
114	Compress 12 Zeros
115	Compress 13 Zeros
116	Compress 14 Zeros
117	Compress 15 Zeros
043	Compress 3 Spaces
044	Compress 4 Spaces
045	Compress 5 Spaces
046	Compress 6 Spaces
047	Compress 7 Spaces
050	Compress 8 Spaces
051	Compress 9 Spaces
052	Compress 10 Spaces
053	Compress 11 Spaces
054	Compress 12 Spaces
055	Compress 13 Spaces
056	Compress 14 Spaces
057	Compress 15 Spaces
060	Compress 16 Spaces
061	Compress 17 Spaces
062	Compress 18 Spaces
063	Compress 19 Spaces
064	Compress 20 Spaces
065	Compress 21 Spaces
066	Compress 22 Spaces
067	Compress 23 Spaces
070	Compress 24 Spaces
071	Compress 25 Spaces
072	Compress 26 Spaces
073	Compress 27 Spaces
074	Compress 28 Spaces
075	Compress 29 Spaces
076	Compress 30 Spaces
077	Compress 31 Spaces

APPENDIX B

INSTALLATION

&

AUTOLOAD

INSTALLATION

The purpose of an installation is to write the current software on the autoloading device of the subsystem.

The software may be installed from paper tape, or from cards, or the combination of paper tape and cards. If a current system is not installed, a maintenance console is necessary for installation. The following is an explanation of the three methods of installation.

A. Installation from Paper Tape

1. Disable the write protect on the micro-drum.
2. Load the paper tape via the console paper tape reader.
3. Depress the Autoload button for the paper tape reader.
4. The reader should stop at the end of a group of overlays, this can be detected by the presence of a leader after the section loaded.
5. Depress reader Halt, Master Clear, Channel Clear, and Go.
6. Set P to °400°, repeat steps 3 thru 5.
7. Enable the write protect feature on the micro-drum.
8. Autoload the system.

B. Installation using Paper Tape Loader and Cards

1. Disable the write protect feature on the micro-drum.
2. Set P to °1000°, load the paper tape loader and depress the Autoload button on paper tape reader or for paper tape reader.
3. At the card reader: If a MARK II, depress Clear and On-Line. If a 410 or 430, depress the Start switch. All cards from the input hopper should be read.
4. The system is now loaded.
5. Enable the write protect feature on the micro-drum
6. Autoload the system.

C. Installation from Cards

1. Autoload the off-line system.
2. Disable the write protect feature on the micro-drum.
3. Depress the F7 Function Key on the keyboard.
4. At the card reader: If a MARK II, depress Clear and On-Line. If a 410 or 430, depress the Start switch. All cards from the input hopper should be read.
5. The new system will now be loaded.
6. Enable the write protect feature on the micro-drum.
7. Autoload the new system.

AUTOLOAD

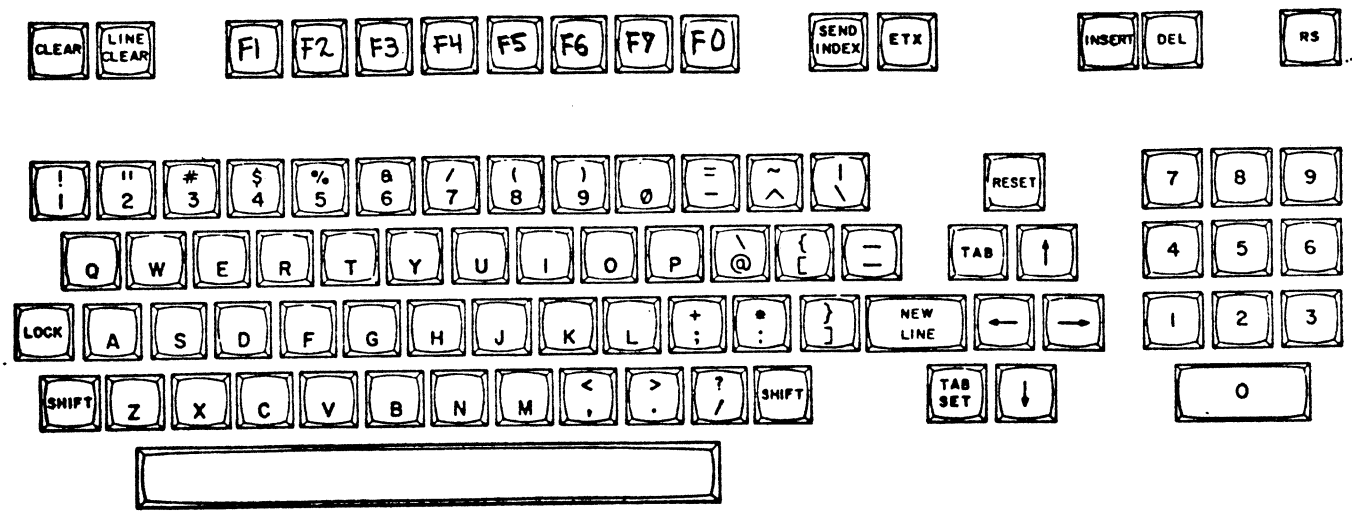
Once the software has been installed on the drum, it can be loaded into the BC (Buffer Controller) by depressing the AUTOLOAD key on the operator's console. Then depress the F1 key on the keyboard and the software is loaded into the BC memory.

RESTART

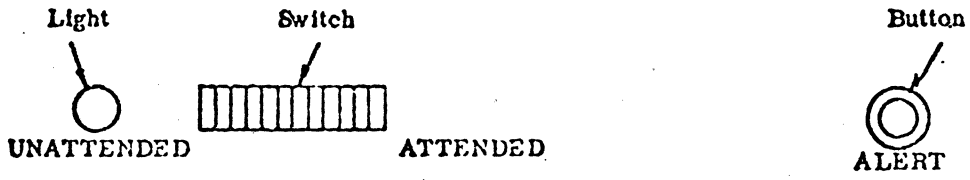
Restart and Autoload are synonymous for this product.

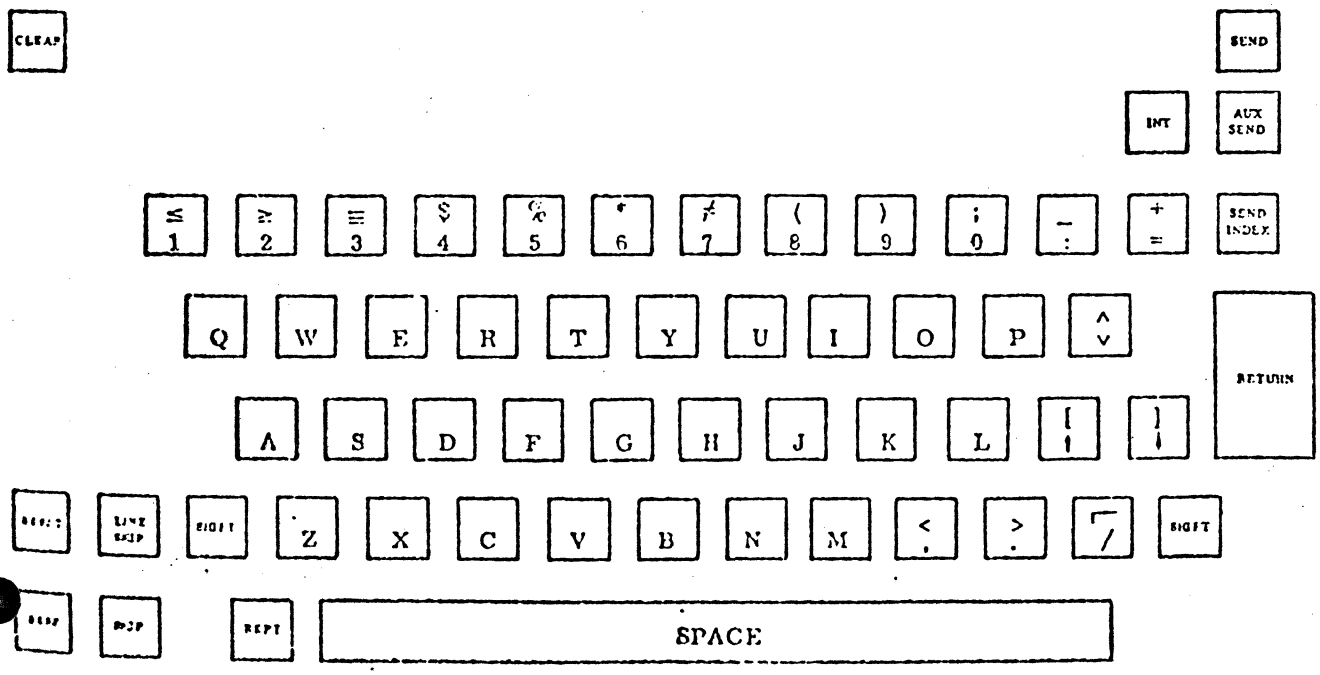
APPENDIX C

KEYBOARD



LOW/MEDIUM SPEED BATCH KEYBOARD





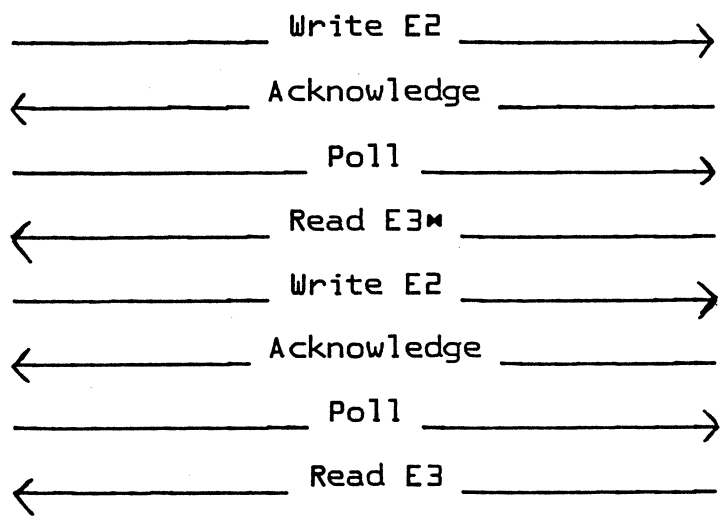
200 USER TERMINAL

APPENDIX D

TYPICAL MESSAGE SEQUENCES and STATION ADDRESS SEQUENCES

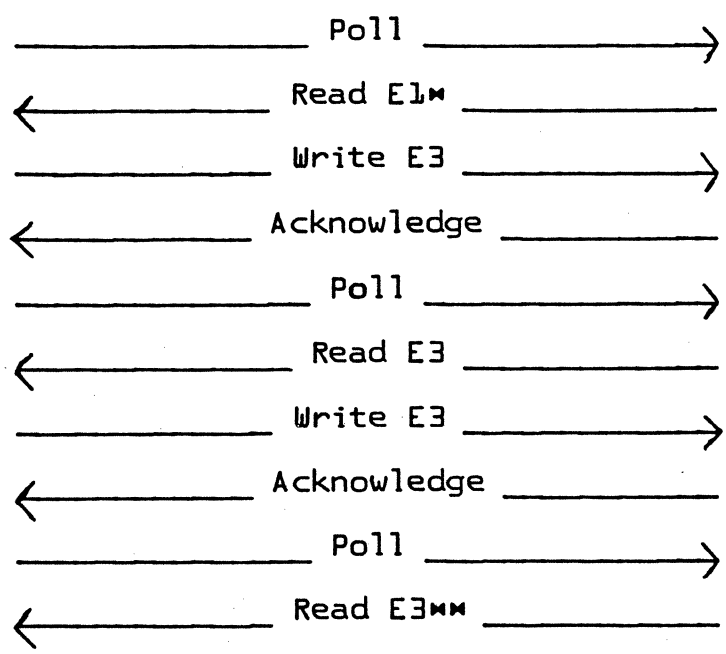
TYPICAL MESSAGE SEQUENCES

1. Typical message sequence when printing.



M Indicates printer is ready. A Read E2 would be transmitted if the printer is not ready.

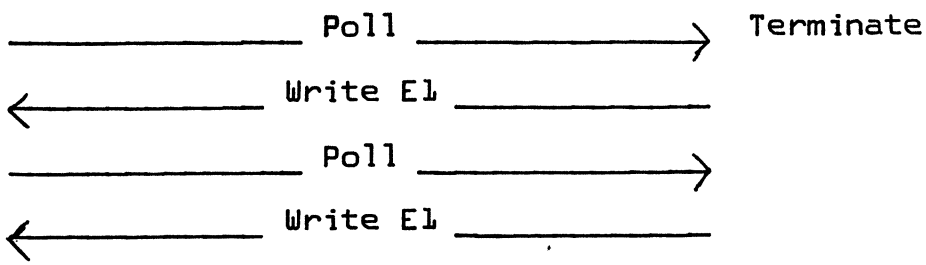
2. Typical message sequence when reading cards.



M Message from the keyboard informing the central site there are cards to be read.

MM Indicates there are more cards to be read. A Read E2 will be transmitted when the hopper is empty.

3. Typical message when displaying messages.



STATION ADDRESS

The Station Address must alternate between 141 and 161 octal for each correct successive write message. When the write message is correctly received the Low/Medium Speed Batch Terminal will return the same Station Address (as received in the write message) in the acknowledge or reject message. When the write message is incorrectly received the Low/Medium Speed Batch Terminal will return the alternate Station Address in the error message.

The following list shows the station address situation for both incorrect and correct receipt of write messages. Normally, the data source retransmits a write message if it is incorrectly received. Message 5 indicates this procedure.

	Transmitted Message	Station Address	Response Message	Station Address
Message 1	Write	161	Acknowledge	161
Message 2	Write	141	Acknowledge	141
Message 3	Write	161	Acknowledge	161
Message 4	Write	141	Error	141
Message 5	Write	141	Acknowledge	141

If the data source failed to receive a response correctly, another message (poll or alert) may be transmitted in an effort to determine the status of the preceding write message. This new message may use either of the station addresses. If the original write message had been correctly received, the response to this second transmission supplies the original station address.

In the following list, the acknowledge message response to the write message marked with an asterisk (*) is assumed to have been destroyed by a telephone channel error burst. By transmitting a poll message, the data source receives a reject message response (assuming no read requests) containing the station address used by the preceding write (however, 160 is used instead of 161 in reject response to a poll). This station address tells the data source that the previous write message was received and processed correctly and retransmission is not necessary.

	Transmitted Message	Station Address	Response Message	Station Address
Message 1	Write	161	Acknowledge	161
Message 2	Write	141	Acknowledge	141
Message 3	Write *	161	Acknowledge	161
Message 4	Poll	140 or 160	Reject	160
Message 5	Write	141	Acknowledge	141

If the write message had been incorrectly received, the error message response contains the alternate station address as shown by the following list. Assuming a telephone channel error burst destroys this error response, the data source transmits a poll message to determine the status. An alternate station address in the reject message response informs the data source of the error condition in the previous write. Retransmission of the write message is the normal procedure.

	Transmitted Message	Station Address	Response Message	Station Address
Message 1	Write	161	Acknowledge	161
Message 2	Write	141	Acknowledge	141
Message 3	Write	161	Acknowledge	141
Message 4	Poll	140 or 160	Reject	140
Message 5	Write	161	Acknowledge	161

If a Read Request (see Section 5.2.5) is available when the Poll message is received, a Read message will be transmitted instead of the Reject message in all the above examples. The Station Address in the Read message will be the same as that shown for the Reject messages above.

The Alert message can have a 141 or 161 octal Station Address any time.

APPENDIX E

TEMPORARY FIX FACILITY

TEMPORARY FIX FACILITY

To allow field {small temporary fixes without reassembling} changes it is necessary to have a map of the software on the drum. When it has been determined what area on the drum is going to be modified, the changes can be made using the following procedure.

1. Depress the Autoload key {in Buffer Controller cabinet}.
2. Depress the F4 Function key. {The off-line system is now loaded.}
3. Depress the Clear key on the keyboard.
4. Key in S5,02XX {XX = Drum track address of data to be read}.
5. Depress New Line key.
6. S6, will appear on the screen.
7. Key in GGHH {GG = Starting sector address of data to be read, HH = Starting word address of data to be read}.
8. Depress New Line key.
9. S7, will appear on the screen.
10. Key in JJJJ {JJJJ = Starting word address of where data is going to be written into the Buffer Controller Memory}.
11. Depress New Line key.
12. S8, will appear on the screen.
13. Key in KKKK {KKKK = Length of the data transfer, in words}.
14. Depress New Line key.
15. The right hand side of the display must begin with address 0.
16. Depress F0 key.
17. The data is in core beginning at location JJJJ as defined in statement 10 above.
18. Depress Clear key.
19. Display the area in the Buffer Controller core that is going to be modified by using the following procedure:

a. Key in RRRR {RRRR is address to be displayed}.

b. Depress New Line key.

1. The screen will appear with RRRR as the first address displayed.

c. It is possible to change only the left half of the screen by preceding the RRRR {paragraph 19 {a} above} with a °<° character. The right half of the screen can be changed by preceding the RRRR {paragraph 19 {a} above} with a °>° character.

1. Example: Key in <RRRR and depress New Line; the first address displayed will be RRR0 and the screen to the right of second column of ¤'s will remain unchanged.

20. Depress the Clear key.

21. Key in STTTT,VVVV {TTTT = Address to be modified, VVVV = Modified contents of TTTT}.

22. Depress New Line key.

23. Continue steps 20, 21 and 22 until all modifications are made.

a. TTTT will be incremented by one each time the New Line key is depressed automatically so it is not necessary to depress the Clear key and key in STTTT, if contiguous locations are modified.

24. Write the modified program back on the drum using the following procedure.

a. Depress the Clear key.

b. Key in S5,03XX {XX = Drum track address where data is to be written}. Extreme care must be exercised to assure that the modified program does not extend into another program area on the drum. If the modified program is written in a different area on the drum than it was removed from, the overlay tables must be changed to reflect the new location on the drum.

c. S6, will appear on the screen.

d. Key in GGHH {GG = Starting sector address of where data is to be written. HH = Starting word address of where data is written}.

e. Depress New Line key.

f. S7, will appear on the screen.

g. Key in JJJJ {JJJJ = Starting word address of where data is going to be read from in the Buffer Controller Memory}.

h. Depress New Line key.

i. S8, will appear on the screen.

j. Key in KKKK {KKKK = Length of the data transfer, in words}.

k. Depress New Line key.

l. The right hand side of the display must begin with address 0.

m. Depress F0 key.

n. Depress Autoload key.

o. Select off-line or 200 UT mode by depressing F1 or F4

F1 = 200 UT Mode
F4 = Off-Line

p. The modified system is executing.

APPENDIX F

OFF-LINE FEATURES

A. AUTOLOAD

Once the software has been installed on the drum, the off-line system can be loaded into the BC by depressing the AUTOLOAD key on the operator's console. Then depress the F4 key on the keyboard. Next clear the keyboard with the Clear key and the off-line system is ready to perform any of the following functions.

	Function	Function Key To Depress
1.	Read/Write the drum	F0
2.	Card to Print	F2
3.	Memory to Print Dump	F5
4.	Installation of new system from cards	F7

APPENDIX G

200 UT Emulation

OPTIONS

A. The following is a list of the selectable options available with the 200 UT Emulation software:

1. External BCD
2. Internal BCD
3. ASCII
4. 026 Punch Card
5. 029 Punch Card
6. Print 6 lines per inch
7. Print 8 lines per inch
8. 2 wire
9. 4 wire
10. Line mode
11. Block mode
12. Site Address

B. To select the options:

1. Autoload
2. Depress the F2 Function Key
3. The display will appear with the message

   200 UT OPTIONS Y = YES N = NO ___.

   Depression of N will select the standard options which are:

   a. 4 Wire
   b. Block Mode
   c. 026 Punch Card
   d. Ext BCD

4. If Y is typed in the display will appear with the message:

   {2-WIRE = 1, 4-WIRE = 2 ___}

   Select 1 or 2.

5. The next message displayed is:

   {BLOCK MODE = 1, LINE MODE = 2 ___}

   Select 1 or 2.

6. The next message displayed is:

   {026 = 1, 029 = 2 ___}

   Select 1 or 2.

7. The next message displayed is:

   {CODE EXT BCD = 1, INT BCD = 2, ASCII = 3 ___}

   Select 1, 2 or 3.

8. The next message displayed is:

   {PRINT 6 LPI = 1, 8 LPI = 2 ___}

9. The next message displayed is:

   {SELECT SITE ADDRESS 160-177 ___}

   Select site.

   Depressions of site will complete the selection. The system is now loaded and ready for use.

APPENDIX H

ERROR CODES

ERROR CODES

The following is a list of all error codes used in the 200 UT software.

1. Printer

XX2	NOT READY {STOP}
XX4	SYNC ERROR
XX8	PAPER RUN AWAY
X1X	PAPER FAULT
X2X	DRUM LATCH SWITCH
X4X	HAMMER FUSE or 29 VOLT FAULT
X8X	ILLEGAL CHARACTER CODE

2. Card Reader

a. MARK II

XX2	NOT READY
XX4	STROBE COUNT ERROR
X1X	DARK CHECK FAIL
X2X	LIGHT CHECK FAIL
1XX	READ HEAD JAM
2XX	MOTOR OFF
4XX	HOPPER EMPTY or STACKER FULL
8XX	HOPPER FEED JAM

b. 410/430

XX1	CLUTCH MALFUNCTION
XX2	NOT READY {STOP}
XX4	COLUMN COUNT ERROR {CARD SLIP}
XX8	SKEW ERROR or SKEW TEST FAILURE
X1X	DARK CHECK FAIL
X2X	LIGHT CHECK FAIL
X4X	0^{o} SWITCH MALFUNCTION
X8X	321^{o} SWITCH MALFUNCTION
1XX	READ STATION JAM/EXIT JAM
2XX	MOTOR OFF/INTERLOCK MODE ERROR
4XX	STACKER FULL or HOPPER EMPTY
8XX	FAIL TO FEED

3. BG300 DRUM

XX1	HEAD SELECT FAULT
XX2	POWER FAULT
XX4	DRUM FAULT
X1X	DRUM BLOCK ERROR

4. COMMUNICATION INTERFACE

XX1	DID NOT RECEIVE 4 SYNC CODES
XX2	EXCEEDED TOTAL NUMBER OF CONTROL CODES
XX4	DID NOT RECEIVE START OF HEADER
XX8	UNIDENTIFIED SITE ADDRESS
X1X	UNIDENTIFIED STATION ADDRESS
X2X	UNIDENTIFIED CONTROL CODE
X4X	DID NOT RECEIVE ESCAPE CODE
X8X	INPUT CHARACTERS EXCEEDED 1040
1XX	DID NOT RECEIVE END OF TRANSMISSION
2XX	INCORRECT MESSAGE PARITY CHECK
4XX	CARRIER ON ERROR
8XX	LINE OUT OF SYNCHRONIZATION

APPENDIX I

CHARACTER SET

200 USER Terminal L/MSBT

ASCII	BCD INT	BCD EXT	DELAY LINE CODE	DISPLAY SYMBOL	LINE PRINTER SYMBOL	LINE PRINTER CODE	HOLLERITH CODE	ASCII CODE	029 PUNCHES	026 PUNCHES	ASCII ARRAY
101	121	061	61	A	A	61	12-1	41	12-1	12-1	A
102	122	062	62	B	B	62	12-2	42	12-2	12-2	B
103	123	063	63	C	C	63	12-3	43	12-3	12-3	C
104	124	064	64	D	D	64	12-4	44	12-4	12-4	D
105	125	065	65	E	E	65	12-5	45	12-5	12-5	E
106	126	066	66	F	F	66	12-6	46	12-6	12-6	F
107	127	067	67	G	G	67	12-7	47	12-7	12-7	G
110	130	070	70	H	H	70	12-8	48	12-8	12-8	H
111	131	071	71	I	I	71	12-9	49	12-9	12-9	I
112	041	041	41	J	J	41	11-1	4A	11-1	11-1	J
113	042	042	42	K	K	42	11-2	4B	11-2	11-2	K
114	043	043	43	L	L	43	11-3	4C	11-3	11-3	L
115	044	044	44	M	M	44	11-4	4D	11-4	11-4	M
116	045	045	45	N	N	45	11-5	4E	11-5	11-5	N
117	046	046	46	◊	◊	46	11-6	4F	11-6	11-6	◊
120	047	047	47	P	P	47	11-7	50	11-7	11-7	P
121	050	050	50	Q	Q	50	11-8	51	11-8	11-8	Q
122	051	051	51	R	R	51	11-9	52	11-9	11-9	R
123	062	122	22	S	S	22	0-2	53	0-2	0-2	S

200 USER Terminal — L/MSBT

ASCII	BCD INT	BCD EXT	DELAY LINE CODE	DISPLAY SYMBOL	LINE PRINTER SYMBOL	LINE PRINTER CODE	HOLLERITH CODE	ASCII CODE	029 PUNCHES	026 PUNCHES	ASCII ARRAY
124	063	123	23	T	T	23	0-3	54	0-3	0-3	T
125	064	124	24	U	U	24	0-4	55	0-4	0-4	U
126	065	125	25	V	V	25	0-5	56	0-5	0-5	V
127	066	126	26	W	W	26	0-6	57	0-6	0-6	W
130	067	127	27	X	X	27	0-7	58	0-7	0-7	X
131	070	130	30	Y	Y	30	0-8	59	0-8	0-8	Y
132	071	131	31	Z	Z	31	0-9	5A	0-9	0-9	Z
75	113	113	13	=	=	13	8-3	3D	8-6	8-3	=
100	114	114	14	≠	≠	14	8-4	23	8-4	8-4	#
47	115	115	15	≤	≤	15	8-5	22	8-5	8-5	//
45	116	116	16	%	%	16	8-6	25	0-8-4	8-6	%
42	117	117	17	{	{	17	8-7	5B	8-7	8-7	{
40	060	120	00	Blank	Space	00	Blank	20	Blank	Blank	Space
57	061	121	21	/	/	21	0-1	2F	0-1	0-1	/
175	072	132	32	}	}	32	0-8-2	5D	11-8-2 0-11	11-8-2 0-11	}
54	073	133	33	, comma	,	33	0-8-3	2C	0-8-3	0-8-3	,
50	074	134	34	(	(	34	0-8-4	28	12-8-5	0-8-4	(
137	075	135	35	↱	↱	35	0-8-5	40	0-8-5	0-8-5	@
136	076	136	36	≡	≡	36	0-8-6	26	11-8-7	0-8-6	&

✖ EOF for 3000 EXT BCD

ASCII	BCD INT	BCD EXT	DELAY LINE CODE	DISPLAY SYMBOL	LINE PRINTER SYMBOL	LINE PRINTER CODE	HOLLERITH CODE	ASCII CODE	029 PUNCHES	026 PUNCHES	ASCII ARRAY
55	040	040	40	- minus	-	40	11	2D	11	11	-
77	077	137	37	∧	∧	37	0-8-7	5E	0-8-7	0-8-7	∧
175	052	052	52	V	V	52	11-0	5C	0-11 11-8-2	0-11 11-8-2	V
44	053	053	53	$	$	53	11-8-3	24	11-8-3	11-8-3	$
52	054	054	54	*	*	54	11-8-4	2A	11-8-4	11-8-4	*
41	055	055	55	↑	↑	55	11-8-5	21	12-8-7	11-8-5	!
43	056	056	56	↓	↓	56	11-8-6	3F	8-3	11-8-6	?
76	057	057	57	>	>	57	11-8-7	3E	0-8-6	11-8-7	>
53	120	060	60	+	+	60	12	2B	12-8-6	12	+
74	132	072	72	<	<	72	12-0	3C	12-8-4	12-0	<
56	133	073	73	.period	.	73	12-8-3	2E	12-8-3	12-8-3	.
51	132	074	74	)	)	74	12-8-4	29	11-8-5	12-8-4	)
46	135	075	75	≥	≥	75	12-8-5	27	12	12-8-5	'
None	None	None	76	Parity Error		76	None				
73	137	077	77	;	;	77	12-8-7	3B	11-8-6	12-8-7	;
72	112	100	20	:	:	20	8-2	3A	8-2	8-2	:
60	100	112	12	0 Zero	0	12	0	30	0	0	0
61	101	101	01	1	1	01	1	31	1	1	1
62	102	102	02	2	2	02	2	32	2	2	2

AA 3777

PRINTED IN U.S.A.

CONTROL DATA CORPORATION

DIVISION

DOCUMENT CLASS ERS PAGE NO. I-4

PRODUCT NAME Low and Medium Speed Batch Terminal/6000/3000

PRODUCT MODEL NO. MACHINE SERIES PL10

200 USER Terminal — L/MSBT

ASCII	BCD INT	BCD EXT	DELAY LINE CODE	DISPLAY SYMBOL	LINE PRINTER SYMBOL	LINE PRINTER CODE	HOLLERITH CODE	ASCII CODE	029 PUNCHES	026 PUNCHES	ASCII ARRAY
63	103	103	03	3	3	03	3	33	3	3	3
64	104	104	04	4	4	04	4	34	4	4	4
65	105	105	05	5	5	05	5	35	5	5	5
66	106	106	06	6	6	06	6	36	6	6	6
67	107	107	07	7	7	07	7	37	7	7	7
70	110	110	10	8	8	10	8	38	8	8	8
71	111	111	11	9	9	11	9	39	9	9	9
74	67	127					7-8-9				
73	66	126					6-7-8-9				

These codes taken from 200 UT Hardware Ref. Man. Pub. No. 82128000
Secondary Source - 200 UT USASCII Compatible Pub. No. PD82136000