CCSYSTM-022-SS

CONTROL DATA CORPORATION

CONSOLIDATED COMMUNICATIONS SYSTEM SYSTEM SPECIFICATION

CONTROL DATA®
MP-32
COMPUTER SYSTEMS

FLEET NUMERICAL OCEANOGRAPHY CENTER

CONSOLIDATED COMMUNICATIONS SYSTEM

SYSTEM SPECIFICATION

CCSYSTM-022-SS-00-A

CDRL A006

REVISION RECORD	
REVISION	DESCRIPTION
A	Original release; supersedes CDRL A004.
(03/15/85)	

Document Number
CCSYSTM-022-SS-00

Revision letters I, O, Q, and X are not used.

CONTROL DATA CORPORATION
SYSTEMS TECHNOLOGY DIVISION
215 Moffett Park Drive
Sunnyvale, California 94086

LIST OF EFFECTIVE PAGES

Page	Revision
Cover	-
ii through vii	A
1-1	A
2-1 through 2-3	A
3-1 through 3-11	A
A-1 through A-3	A
A-4	blank
A-5 through A-9	A
A-10	blank
A-11 through A-21	A
A-22	blank
A-23 through A-40	A
B-1	A
C-1 through C-5	A
D-1 through D-3	A
F-1	A

Page	Revision

PREFACE

This document is the System Specification document for the Consolidated Communications System (CCS). It represents Control Data Corporation's design of CCS to meet the requirements as stated in the CCS Statement of Work.

TABLE OF CONTENTS

APPENDIXES

1.0 GENERAL

1.1 Purpose

The System Specification for the Consolidated Communications System (CCS) is written to fulfill the following objectives:

a. To provide definition of the CCS system functions.

b. To define in detail the interfaces with other systems and subsystems and the facilities to be utilized for accomplishing the interfaces.

1.2 Project References

The list of documents contained in CCSYSTM-019-TIP-00-A Section 1.3 has pertinent information in understanding the Consolidated Communications Software. These documents, even if not directly referenced in the text, should be considered part of this document.

2.0 SUMMARY OF REQUIREMENTS

2.1 System Description

A general system description is contained the CCS-A004-MO-00-A, CCS Management Overview.

2.1.1 CCS MP-32 Applications Software

- Inter-Computer Network software (ICN) - This software provides a method of communication between tasks that span physical mainframes.

- CYBER Channel Coupler Link Driver (CCCLD) - This software interfaces with ICN on the MP-32 to the CYBER Channel Coupler Manager (CCCMGR) in MPX/OS, which handles the hardware interface to the IPS.

- Network Manager (NM) - This software statuses the IPS computers and responds to IPS NM requests to maintain the ICN Machine Table and ICN Link Table.

- Transfer Queue Processor (TQ) - This software creates CCS messages from IPS Transmit files. It will also perform character translation during the transfer when requested.

- Receive Queue Processor (RQ) - This software creates or adds CCS messages received from the remote circuits to either Permanent or Queue files on the IPS. This software will also perform character translation during the transfer when requested.

- Queuer Process (Q) - This software stacks short messages received from a remote circuit into a single summed message and then releases that message to the CCS message traffic based on time or size. It will also perform character translation.

- Message Routing Software (MRS) - This software directs the flow of message traffic through the CCS. It is composed of Message Routing Task (MRT), Message Routing Reading routines (MRR), and Bit Stream I/O (BSIO).

- Protocol Formatting Modules (PFM) - This software maps circuit format messages to CCS internal format messages and vice versa, and communicates with each circuit according to its specific protocol.

- System Operator Task (SOT) - This software interfaces with the Interactive Terminal Subsystem (ITS) to the CCS application to allow the operator to control message processing on the CCS.

- Message Editor - This software provides the CCS operator to interface with CCS message traffic.

- Report Generator - This software provides a utility for statistical summaries of dayfile messages.

2.1.2 MPCLA Software (Circuit Interface Software)

This software provides PFM with an interface to the circuits according to the specific protocol. Low-level synchronization and data enveloping is handled by this software.

2.1.3 IPS Software Modification

The IPS support software includes software modifications within the IPS environment made necessary by the interface requirements of CCS. this

includes software to interface with the CCS hardware. This software also provides the functional capabilities for CCS to access IPS time, date, files, and queue lists.

3.0 ENVIRONMENT

3.1 Equipment Environment

The CCS hardware is configured as stated in the Test Plan CCSYSTM-014-TIP-00-A.

3.2 Support Software

The support software required for the CCS system is divided into four categories and is described in the following four sections. Features of the MPX/OS system, the Interactive Terminal Subsystem (ITS), and the MPX/OS product set which are required by CCS are described in the first three sections. These sections include the description of the modifications needed to satisfy the CCS requirements. The last section describes certain features of the ICN software which will be required to support CCS. The CCS modifications to this software are discussed in section 4.2 of this document.

3.2.1 MPX Operating System Features

The MPX Operating System (MPX/OS) performs the System Control Functions. The MPX/OS has ultimate control over the access to CCS resources, and allocates resources to the CCS software modules based upon information supplied by the communicationse software modules. MPX/OS V3 is described in V3.0 Reference Manual (17329125).

3.2.2 Interactive Terminal Subsystem Features

The Interactive Terminal Subsystem provides an interface between terminal users (i.e. interactive terminals, remote batch terminals, and the system console) and the system. This interface allows users to communicate with jobs in the system, monitor and alter system input, output, and execution queues, monitor job execution, monitor the system dayfile, control external

peripheral devices, control remote batch input/output, and monitor system file management. The Interactive Terminal Subsystem Reference Manual (17529140) describes the ITS.

3.2.3 MPX/OS Product Set Features

This section lists the members of the MPX/OS Product Set which are required by the CCS. The reader is directed to the appropriate reference manual for a description of the features of each product.

FORTRAN Compiler	COMPASS
MP-60 Loader	COPYL
MP-60 Update	Utility
MICRO	MASS
Z-80 Relocatable Assembler	Z-80 Linking Loader

3.2.4 Inter-Copmputer Network Features

This section describes the features of the Inter-Computer Network (ICN) which are to be provided by the Government and are described here to define the ICN software required by CCS. These descriptions do not include the enhancements to the ICN modules necessary to satisfy the additional requirements imposed by CCS. These enhancements are described in section 4.2.

Postal Facility

The Network M.ICE Post Office functions are the vehicle through which process to process communication is established among the CDC CYBER 170/6000 series machines in the Inter-Computer Network. These functions provide a PPU process with the ability to:

- Obtain an ECS buffer for construction of a network message and to initialize the header fields of the message.

- Place message into the ECS buffer.
- Deliver the ECS buffer containing the message to the destination PPU.
- Receive a message addressed to this PPU from the network and extract information in the header of the message.
- Extract data from the body of the message.
- Return an ECS buffer to the list of unused buffers (free list).

The arguments to and information returned from these M.ICE functions are contained in the PP message buffer area associated with the particular PPU using the functions.

File Access Facility

The File Access Facility (FAF) is a process that is called into execution by a message arriving at its mailbox. The FAF will access and perform operations on the local file system on behalf of a process running in its own or some other machine. The process using the FAF will send a command and perhaps some data as a network message. FAF will interpret the command, perform the requested function, and return a reply message to the requestor. The major and minor modules of the FAF are described below, and include WRITE, READ, POSITION, RETURN, REQUEST, ROUTE, CATALOG, ATTACH, EXTEND, PURGE, DAYFILE, AND QUEUE. FAF begins by picking up a message from its mailbox and decoding the request.

Write

The logical file name of the file to be written is contained within the request and that, in conjunction with the requestor's network address, isused to search the File Name Table (FNT) for the local file. If the name is not found, an FNT entry is created. The file contained in the

message received is then written to the local file. A response is sent to the requestor.

Read

The local file name is obtained from the request. The FNT is searched for a match. If the specified file is not found, a "FILE DOES NOT EXIST" reply message is sent to the requestor. Otherwise, the data is read into the FAF buffer until the specified EOR, EOF, or EOI or end of the PPU buffer is encountered (as determined by the stop type). The Data Read is formatted into a network mesage. Dependent upon the stop type, the number of records encountered, number of files encountered, or the number of words transferred is added to the reply. The reply is sent back to the requestor.

Position

The logical file name is extracted from the request, the FNT is searched and if the file is found, the requested positional operation is performed (rewind, skip, and so forth). A reply message is then sent to the requestor.

Return

The FNT is searched for ALL logical file names present in the message. If a match is found, the file is returned. At completion a reply message is sent to the requestor.

Request

Extract the logical file name from the request. Search for the FNT for the specified file. If it does not exist, then build an FNT entry for the file on the device specified in the request. Send a reply message

to the requestor. If the file did exist, send an error reply message indicating the file already exists.

Route

The logical file name is extracted from the request. The FNT is searched and if the file is found, it is routed to an input or output queue with the specified routing parameters. A reply message is sent to the requestor. If the file was not found in the FNT search, an error reply is sent to the requestor.

Catalog

The logical file name is extracted from the request. The FNT is searched and if the file is found, it is routed to an input or output queue with the specified routing parameters. A reply message is sent to the requestor. If the file was not found in the FNT search, an error reply is sent to the requestor.

Attach

The logical file name is extracted from the request. A long form permanent file request is issued with the permanent file parameters supplied in the request. The appropriate bits are set in the request to ensure that requests are not queued and return codes are returned. If the ATTACH is successful, the FNT address of the file is added to the reply message. A reply message with the returned status information is sent to the requestor.

Extend

The logical file name is extracted from the request. The FNT is searched and if a file is foiund, the PFM routine for extends is called with permfile parameters supplied with the request. A reply message is

formatted with the status information returned from the PFM routine and sent to the requestor. If the file is not found, an error reply message is sent to the requestor.

Purge

The logical file name and other permanent file parameters are extracted from the request. The PFM routine for purging is called with the supplied parameters. If purge is successful, the local file is returned. A reply message is formatted from the status information returned from the PFM routine and sent to the requestor.

Dayfile

The data portion of the request received is formatted as a dayfile message and entered as a dayfile message as specified by the dayfile flag bits. A reply message is returned to the requestor.

Queue (Create, Evict)

The logical file name and TID (if non-zero) are extracted from the request and the FNT is searched for a file that matches. If one is found, 1QF is called to create or evict the entry in the PFO. A reply message is sent to the requestor indicating the status of the request.

Network Status Facility

The Network Status Facility (NSF) is designed as a network server process in that it will respond with a reply to each request received. This module is a collection of all of the requirements for status information by other elements of ICN. In particular, the following status subsystems are included in NSF to provide status information for:

- The Load Leveler Process
- Intercom Q Commands

- o Operator Display
- o Network Statistics Gatherer
- o Network Echo Process

The Network Status Facility is the type of server process that the postal system network will automatically call into execution each time a message is delivered to its mailbox. Upon entry NSF obtains the request message from its mailbox. A code field in the request identifies the type of request and therefore the form of reply desired.

3.3 Interfaces

The interfaces between the CCS and the following are described in Appendix F.

- o FNOC Transmit Queue
- o FNOC Receive Queue
- o MPX/OS Dayfile
- o FNOC Inter-Computer Network
- o MPCLA Subsystem
- o FNOC Circuit Protocols

3.4 Security

The Consolidated Communications System will be composed of unclassified computer programs which may have classified input and output data. The classified data is restricted to messages that CCS passes between the CDC CYBER/6000's and the remote circuits. The control of the classified messages will be discussed in paragraph 3.5, Controls.

The hardware devices which may contain or transfer classified information are listed below:

NEDN/NEDS remote communication lines.
Operator's Console and Hard Copy Device.
Validated Interactive Terminals.
Validated Remote Batch Terminals.
CDC CYBER/6000 Interface.
Mass Storage.
Magnetic Tape Units.
Printer Outputs.

3.5 Controls

The controls present in the CCS vary from external controls such as the operator to internal controls such as the MPX/OS task scheduler. The MPX/OS controls are described in section 3.2.1 MPX/OS Features and in the MPX/OS Reference Manual. This section will describe two CCS control areas: Message Transmission and System Security. The CCS transmission controls include the procedures required for each inter-connection to ensure the validity of the transmitted data. The CCS security controls are composed of tables and procedures for the protection of message data from unauthorized access.

Remote Network Controls

The two RS-232C signals "Data Carrier Detect" and "Data Set Ready" are used by the CCS to determine communication line and modem status. The "Data Set Ready" (DSR) signal is used by the CCS to determine communication line modem operability, while the "Data Carrier Detect" (DCD) signal is used for transmission control. If the DSR signal drops while data is being transmitted or received, then the communication line is considered down and the messages being received or transmitted must be re-started. For the current protocols the DCD signal must be "ON" for messages to be transmitted.

In addition to the RS-232C control signals the remote network protocols each have different control mechanisms to validate the reception of error free messages. These mechanisms include parity, checksums, and message sequence numbers.

Security Controls

The CCS security controls are intended to prevent access to MP-32 Computer System resources by non-validated users. These CCS controls are based on information obtained from three sources: initial system tables, system files, and operator entry.

The MPX Operating System maintains security information in the following tables and system files:

- o System security control (security control mask)
- o Peripheral control tables (security contrl mask)
- o Task Control Tables (task current security level)
- o Mass storage directory (file security level)
- o User validation file (interactive user security levels)
- o Port setup file (remote circuit security levels)

The System security mask is initialized at system installation but may be changed by the operator during the CCS operations. The System security mask is a global value levied on all system functions by the operator. No accesses will be allowed which are in conflict with the System security mask. If the operator attempts to alter the value while a CCS task requires resources which are in conflict with this value, the operator will be informed and the System security mask will not be altered.

The peripheral control tables contain the security mask for each peripheral device within the system, but only the unit record equipment security masks will be allowed to be altered by the operator. The others (mass storage, communications lines adapters, and so forth) will not be allowed to vary and are controlled by other system tables and files. If the operator attempts to alter a non-unit record device, the operator will be informed and no changes will be made.

The Task Control Tables contain security level information pertaining to an associated task. The CCS system will use this security level information to control resource allocation to a task. Both the task and the operator may change a task's security level but in no case will the level be allowed to conflict with the system security mask. If an attempt is made to alter the task's security level in a manner which makes it conflict with the system security level or with resources already obtained, then the task/operator will be informed, and no change will be made in the task's security level.

The Mass storage directory contains the information necessary to access the data contained within a file. This directory also contains a use parameter which contains the control level of the file. The use field may be one of the following control types:

Uncontrolled	Read/Write
Controlled	Read Only
Confidential	Secret
Top Secret	Limited to System Tasks

The user validation file is used to control the interactive user access to the MPX Operating System. This file contains the user validation parameters described in the ITS feature description. These parameters include facility access, security level, and username/password combinations. This user validation file is created and maintained by a special task within the ITS.

The MPX/OS port setup file contains configuration and validation parameters for each communications line adapter (CLA) within the CCS. These parameters are described in the ITS features described previously. The port validation mask parameter contains the security control mask and when the port validation mask is combined with the user validation mask, the resulting mask is used to limit the types of accesses the user/port can perform.

Consolidated Communication System

MPCLA
Subsystem Specification

May 1, 1980

Prepared by:
Digital Consulting Corporation.
DIGICOM
P.O. Box 32505
San Jose, California 95152

GLOSSARY

AQ interface	Programmed control interface between MPCLA and MP-32.
ASCII	American Standard Code for Information Interchange. The standard code, using a coded character set consisting of 7-bit coded characters. See American National Standard X3.4-1977.
Asynchronous	Refers to a type of serial transmission in which bit-synchronization is re-accomplished for each character.
Bit	A binary digit.
BYTE	Eight (8) bits.
CRC	Cyclic Redundancy Check. An error detection computation, similiar in usage to a checksum, in which the message (regarded as a polynominal) is divided by a specified polynominal in modulo arithmetic. The receiver should compute the same remainder as the transmitter. Achieves perfect detection of single, double, and odd numbers of errors, and very good detection of burst of error. The Z80 SIO provides a choice of two polynominals: CRC-16 = (x^16 + x^15 + x^2 + 1) and SDLC = (x^16 + x^12 + x^5 + 1). (Timing requirements force this to be a chip function.)
EOP	End of Operation. Typically, the requestor must be notified at this time.
FWA	First Word Address. Used in conjunction with table pointers and/or buffer addresses.
IPL	Initial Program load. The cold start load process: autoload, Deadstart.
LWA	Last Word Address.
LSB	Least sigificant bit, bit 2** 0
MPCLA	Microprogrammable Communication Line Adapter. Control Data equipment FJ130, several of which are included in CCS. Often used in this document to refer to the combination of hardware, plus the controlware defined herein.
MP-32	Primary hardware device for the CCS system.

	Refers to the Control Data primary hardware separate from the MPCLA.
MSB	Most significant bit or byte.
RAM	Random Access Memory. This document uses RAM to indicate the memory contained within the MPCLA.
SLAB	Shorthand for "AWN syllable". A SLAB is equal to 12 bits.
SIO	Serial input output. In particular a Z80 SIO chip which has two (2) serial ports.
Synchronous	Refers to a type of serial transmission in which bit-synchronization is maintained for an entire transmission block. Each character immediately follows the preceeding one. See ANSI X3.16-1976.
WORD	Within this document the meaning is 32 bits, which is one MP-32 word.
Z80	Zilog Z80 cpu, in this document it is used to refer to one side of a MPCLA.

SECTION 1. GENERAL

1.1 Purpose of System/Subsystem Specification.

This MPCLA/CCS System/Subsystem Specification for the CDC Consolidate Communication System (CCS) is written to fulfill the following objectives:

a. To provide a detailed definition of the System/Subsystem functions of the MPCLA parts of the CCS design.

b. To communicate details of the on-going analysis to the user's operational personnel.

c. To define in detail the interfaces between the MPCLA subsystem and the MPX and line adapters utilized in the CCS.

1.2 Project References.

The following list of documents represents the current history of the CCS project as related to the CCS/MPCLA design and development. Although this document may not make explicit references to all documents listed they are considered to offer a contribution to the understanding of the CCS project.

1. A Proposal to the Department of Navy,
Fleet Numerical Weather Central
by Control Data Corporation.
Volumes Two and Three.
in response to FNWC request N00123-78-R-1455.
December, 1978.

2. CCS Project Plan and control document.
Control Data Corporation
April, 1979.

3. AWN High Speed Interface Design Document
Appendix B from Reference 1.

4. NEDN/NEDS Interface Design Document
Appendix C from Reference 1.

5. FLENUMWEACEN-NMC Data Exchange Protocol Definition.
Appendix D from Reference 1.

6. System/Subsystem Specification documentation format.
D.O.D. Automated data System Documentation Standards Manual
Manual 4120.17-M
December 1972.

7. Mode 4A Data Communication Control Protocol
CDC Standard 1.10.020
Control Data Corporation, January 1975.

8. Microprogrammable Communication Line Adapter
FJ130-A Hardware Reference/Maintenance Manual,
Publication # 60475570,
Control Data Corporation, March 1979.

9. Z80-CPU, Z80-A-CPU Technical Manual
Part # 03-0029-00
Zilog, Inc., 1977
Cupertino, California

10. Z80-SIO Technical Manual
Part # 03-3033-01
Zilog, Inc., August 1978.
Cupertino, California

11. FNWC/NMC Data Exchange System
N00228-78-C-3302
Preliminary Design Document
Technology Development Corporation
Sunnyvale, CA
4 December 1978

SECTION 2. SUMMARY OF REQUIREMENTS.

2.1 System/Subsystem Description.

The MPCLA subsystem is one part of the CCS design that forms a working communications system. The MPCLA subsystems primary responsibility is to provide a software communications interface to the outside connection. This interface consists of line processors capable of receiving and transmitting data to and from the outside, and processing the connection to the CCS MPX operating system.

The MPCLA processors reside on a hardware module that is capable of interfacing to the outside connection via serial data hardware. The internal interface to the MPX system is provided by an A/Q hardware interface. The combination of these two hardware interfaces provide a means of meeting the requirements defined in the CCS requirements section of the CCS RFP.

2.2 System/Subsystem Functions.

The MPCLA software subsystem is divided into three major parts: 1)Scheduler, 2)Outside line processors, and 3)MPX interface processors.

Each of these parts reside in the RAM provided by the MPCLA hardware and are downloaded upon system start-up or reset by the MPX operating system. The actual line processor loaded into each MPCLA is defined by the MPX equipment status table.

2.2.1 Scheduler.

The Scheduler for the MPCLA subsystem provides a path for all subsystem requests. Its primary purpose is to maintain order in the system and to provide a convenient mechanism to handshake between the other parts of the MPCLA system.

2.2.2 Outside Line Processors.

There are four major Outside Line processor modules: 1)NEDN, 2)High Speed AWN, 3)NMC and 4)Control Data Mode-4A. Each of these modules contain software code to provide for translation of either incoming or outgoing data and software code to support the functioning of the Serial Input/output (SIO) device. Line processor modules are responsible for error detection & recovery only on the basis of a single transmission block where either the hardware provides a mechanism, or the protocol defines a means.

2.2.3 MPX Interface.

The MPX interface part of the MPCLA system consists of: 1)Data processor, 2)Status processor, and 3)Command Decoder.

2.2.3.1 Data Processor

The Data processor module supplies software to support the transfer of data packets between the MPCLA and the MP-32 hardware. The actual data transfer occurs via function code requests placed in Command & Reply Registers located in MPCLA RAM.

2.2.3.2 Status & Reply generator.

The status processor is responsible for issuing all function Replies to the MPX. This module also contains the low level drivers for functioning the interface.

2.2.3.3 Command Decoder.

The Command Decoder examines functions sent by the MP-32, and decides how they are to be processed.

2.3 Accuracy and Validity.

The accuracy and validity of the MPCLA system is divided between that supported by the hardware versus the software. The accuracy part of the question is answered by the SIO hardware interface of the MPCLA. The hardware used supports both parity error and data overrun detection. The software that exercises this hardware will on all SIO operations check these hardware conditions and upon finding an error execute a re-synch procedure that is supported by the protocol processor (Note if the protocol processor is unable to retransmit the 'frame' in error a data flag will be sent in the next packet to the MPX on incoming traffic. On outgoing traffic the error procedure is a function of the protocol processor).

Another hardware condition validated by the software is that the communication line is still active and ready. If the hardware reports a lack of Data Set Ready (DSR), the MPCLA software will notify the router, which will inform the MPX of a down line condition. The MPCLA will continue to monitor the bad (down) line until it becomes active again, or until the MPX indicates this device is not to be polled.

2.3.1 Timing.

The MPCLA controlware is required to support a maximum transfer rate of 19,200 bits per second per Z80. There are thus two timing constraints to be considered in the MPCLA design: 1) Traffic loading/timing of outside lines, and 2) Traffic and timing conditions of loading and unloading the on-board RAM of the MPCLA to/from the MP-32.

In the initial CCS system, the worst case timing constraint per Z80 is one 19,200 bps line. This equates to one interrupt every 416 microseconds. The MPCLA operates on a 250 nanosecond clock with an average instruction taking 1.4 microseconds. The MPCLA system will process a byte save instruction in less than the required 297 instructions, thus guaranting the timing of the system.

For potential future implementation of 50,000 baud support, a Z80 is required to process each byte in 160 microseconds. In order to keep within the 114 instruction limit imposed by this rate, the Z80 will, for the duration of each transmission block, inhibit all processes except data transfer.

Line A of each Z80 has relative priority over line B. When configuring the system, the higher speed line should be assigned to A.

2.4 Flexibility.

The MPCLA provides the CCS system with a programmable external line interface adapter. Since the MPCLA software code executes in MP-32 down loadable RAM the MPCLA is flexible. The MPCLA code defined by this specification, on the other hand, is fixed to support only four basic synchronous line interfaces. The current software code is dedicated to processing these interfaces only. To improve the flexibility, but on the same hand support the CCS requirements, the software interface between the MP-32 and MPCLA is designed around a general command/response (a la controller) mechanism, thus providing a flexible design structure capable of supporting other disciplines.

SECTION 3. ENVIRONMENT.

3.1 Equipment Environment.

The MPCLA hardware is defined as Control Data Corporation equipment Model FJ130-A, described in Reference [8]. We term a logical MPCLA as one side of that equipment, which includes:

- 1 Zilog Z80-A CPU,
- 1 SIO/1 with 2 serial ports,
- 2 PIO with 16 parallel lines each,
- 16k RAM,
- 2 RS232-C interfaces,
- AQ Interface to MP-32.

A physical MPCLA consists of two such stations (the AQ hardware is shared) on one board.

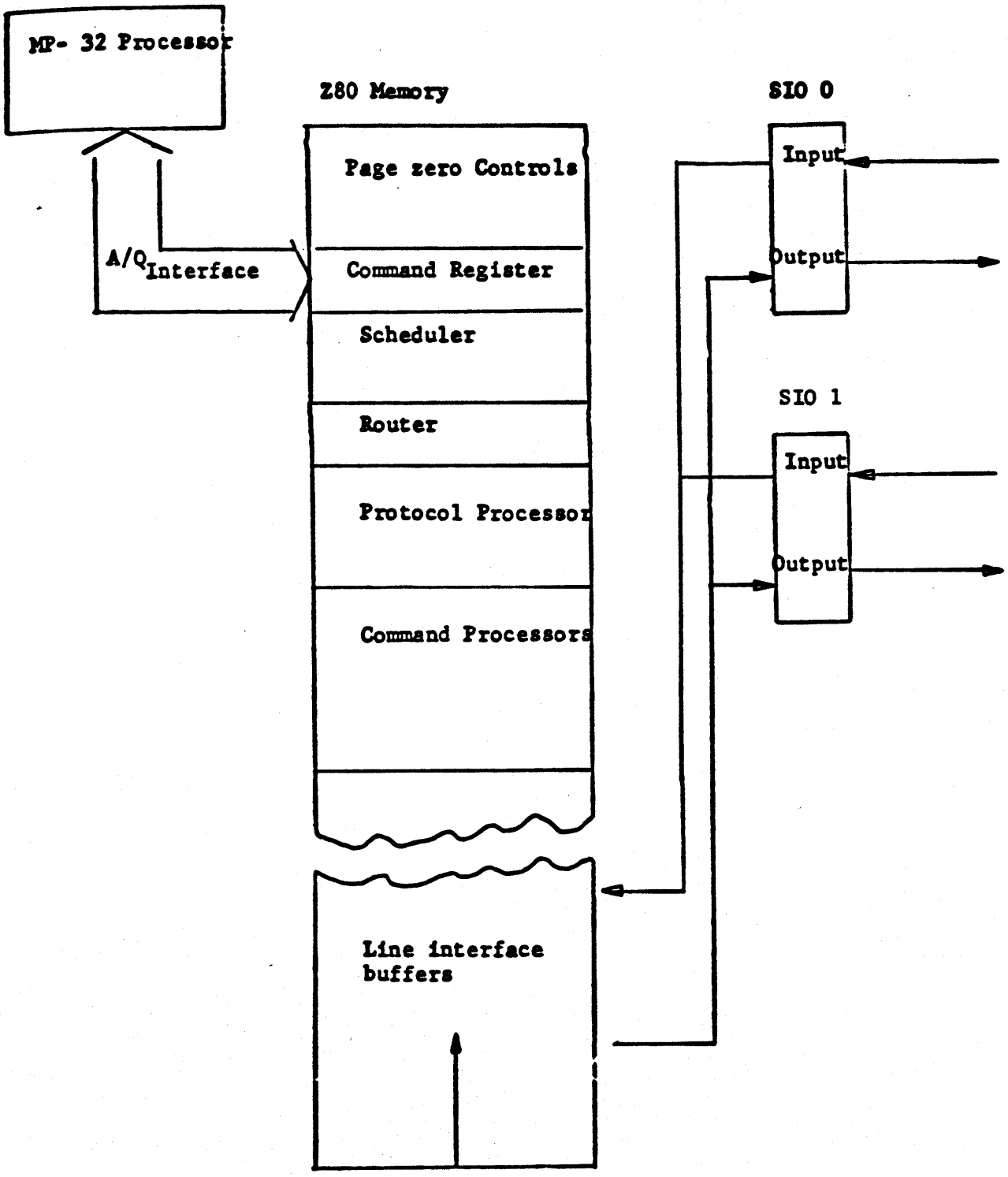
The Consolidated Communications System contains several MPCLA boards, each connected to communications lines as required. The other major CCS component is the MP-32, which is outside the scope of this document. All of the Z80's can access the memory of the MP-32 and interrupt it. Conversely, the MP-32 can access the entire Z80 RAM memory and send an interrupt to the Z80.

The communications device connected to a serial port (SIO) of an MPLCA is either a Modem or an "ADCAP" box. The device connected presents to the SIO an interface equivalent to one of the following:

- LSI 9600 Codex Modem
- Bell 202 Modem
- Milgo 4800 Modem

For all external interfaces the following signals are required to be configured on the EIA RS-232C 25 pin connector:

Pin	Name	Description
1	-	Chassis Ground
2	TxD	Transmitted Data
3	RxD	Received Data
4	RTS	Request to Send
5	CTS	Clear to Send
6	DSR	Data set ready
7	SG	Signal Ground
8	DCD	Data Carrier Detect
15	TxC	Transmitter Clock
17	RxC	Receiver Clock
20	DTR	Data Terminal Ready



MPCLA (Z80) MP-32 Environment

3.2 Support Software Environment.

The MPCLA operates as a slave to the MP-32, which is in overall control of the CCS. The MPLCA functions essentially as a smart device controller, responding to commands received from the MP-32.

3.3 Interfaces

The MPCLA connects inward, tightly, to the MP-32. It's outward connections are over communications lines, and are therefore less intimate, being constrained by hardware and protocol definitions.

3.3.1 External Interfaces.

Each of the External Interfaces exists to transfer data. The MPLCA is not concerned with the content or purpose of any of the data. The MPLCA is not cognizant of any security implications of the data transferred.

The maximum non-interruptible data block is limited to 4000 bytes.

The communications lines supported are detailed in the CCS Statement of Work [reference 1]. There are five types of protocols supported by the MPLCA:

1) NEDN - Naval Environmental Data Network. Per Appendix C of [ref 1], Para 1A: "NEDN is a Full Duplex Synchronous communication system... utilizing a transparent interface unit (ADCCP)". Per Para 1E:

> "The ADCCP units are controlled by the RTS line. When the CCS has data to transmit the RTS line is raised, this sets the ADCCP to receive mode. The CCS will then transfer data to the ADCCP. The ADCCP will accept data until all transmit buffers are full. At this point the ADCCP will stop the transfer clock (TxC), inhibiting the transfer of data from the CCS to the ADCCP. When the ADCCP has successfully transmitted a buffer, the clock will be restarted and the ADCCP will accept more data from the CCS. When a complete message has been transferred the CCS will drop the RTS line and the ADCCP will transmit the partial buffer."

2) AWN - High Speed Automated Weather Network. Full duplex, Synchronous protocol; details in Reference [3]. The MPLCA provides the MPX with a convienent method of transmitting & receiving some of the canned control messages. The MPX is in charge of all message sequencing and retransmission.

3) NMC - National Meterological Center. Full duplex, Synchronous

protocol, 8 bit data, no parity. Uses standard ASCII control codes for definition of transmission mode. Original specification in Appendix D of the Statement of Work has been superceded by Reference [11]. See pp. 42 ff. for message formats and ASCII control code usage.

4) CDC Mode 4A - Control Data Corporation's two-way alternate protocol. Details in Reference [7]. Supports both switched and dedicated lines.

5) Diagnostic - Full transparent data bitstream. Allows MP-32 control of all synch and redundancy check info.

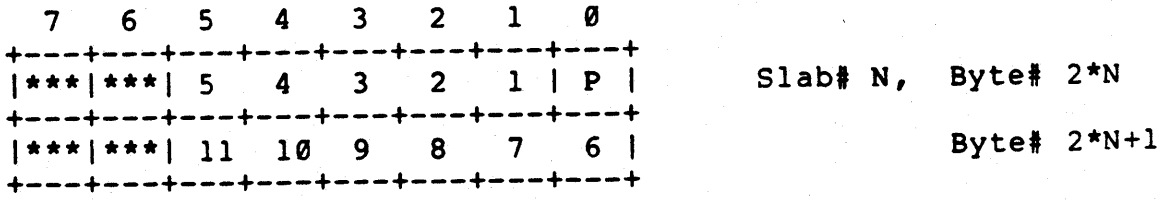
3.3.2 Internal Interface.

The MPCLA interfaces with the MP-32 in order to transfer data as directed, and report transfer completion & status. The MP-32 may also obtain the hardware status of the communication ports. The MP-32 prepares blocks of data to be transmitted, essentially as a memory image of the bytes which are to be output. The MP-32 performs all code conversions, data compression/expansion, byte re-arrangement, and sequencing.

All bytes are padded out to 8 bits for passing between MPCLA & MP-32. Since AWN is oriented toward 12-bit bytes (called "syllables" herein), blocks of AWN data are passed between MP-32 & MPCLA in a "6-of-8" format. The 2 most significant bits of each 8-bit byte are ignored; the lower 6 bits contain half of an AWN syllable, the least significant half being first.

For all bytes of an AWN message horizontal parity is the responsibility of the MPCLA. (The MP-32 should clear bits 7 and 6 of each outbound byte.) The MPCLA conditions the communications hardware for 6-bit transfers in a fashion such that 12-bit slabs are properly transmitted & received. The MPCLA also computes the AWN slab parity for transmit into slab bit Ø (bit Ø of odd bytes). Slab bit 6 (bit Ø of odd bytes) is controlled by MP-32: when it is not a data bit (in Baudot slabs) it must be cleared (zero) for proper parity computation. For receive, slab parity is checked by the MPCLA; the MP-32 ignores the least significant bit of each slab.

The slab format, pictorially is:



The other protocols are either 8-bit, or 7-bits+parity. MPCLA

generates/checks horizontal parity, as required.
Longitudinal checksums are generated/checked by the MPCLA for:
AWN - checksum in slab 95 of each segment.
Mode 4A - Longitudinal parity follows ETX.
NEDN inbound - checksum follows EOM.
ID50 - Block Check Character follows ETB/ETX on data blocks.

The MPLCA is responsible for idle-line time fill characters transmitted to maintain character synchronization. They are transmitted when required by a particular protocol. On the receive side, synch/fill characters are discarded by the MPLCA, and not passed to the MP-32.

3.3.3 MPCLA/MP-32 Command Interface.

For each Station (Z80) of each MPCLA attached to the system, an area of MPCLA RAM is reserved, called the Command/Reply area. This constitutes the major functional interface between the MP-32 and the MPLCAs. Format of this area, definition of Commands & Replies, and the possible responses are contained in Section 3.5 and in Section 4.

The MP-32 directs an MPCLA (station) to perform some function by setting the Command Register and then sending an interrupt (over the AQ interface) to the Z80. When the MPCLA honors the interrupt, it is required to read up the contents of the Command Register before re-enabling interrupts. At this point the MP-32 is free to issue another Command, and the MPCLA must be capable of honoring and stacking further commands without regard to other tasks partially completed. The Commands are defined such that this is possible. For instance, the MP-32 is not permitted to issue any Transmit command for a line with an incomplete prior Transmit.

The MPCLA informs the MP-32 of command completion by issuing a Reply, analogous to the End-of-Operation interrupt available with some controllers. A reply is issued by setting the Reply Register non-zero, and then sending an AQ interrupt to the MP-32. The Reply Register is interlocked by prohibiting an MPCLA from writing into it until AQ interface status shows that the MP-32 does not have an interrupt outstanding from this Z80, and it is zero. Therefore, the MP-32 is required to sample Reply Registers before clearing the "AQ interrupt pending" status.

An MPCLA only writes into his Command Register after sampling it due to AQ interrupt (and then only zero is written). Similiarly, the only time the MP-32 writes into a Reply Register is to zero it to indicate that he has seen the Reply.

Note that the Reply caused by most Commands may be delayed indefinitely. Any number of other Commands (and corresponding Replies) may intervene. For instance, on an NEDN line, the MP-32

might issue a Read at the beginning of a day, and then begin a series of Transmits. Each of the Transmits would cause an appropriate Reply upon completion. It might be desired to shut down the CCS at the end of the day, say, while the Read was still active. To accomplish this cleanly, the MP-32 can issue a "Turn OFF Line" command. The MPCLA would first respond to the pending Read, with Status = "Complete due to Receiver Shutdown". When all activity ceased, the MPCLA would issue a Reply to the OFF command.

The point of all this is that a processor never really has to wait for a desired function to be honored. If he has something to do in the future which requires action from the other (MP-32/Z80), his attention will be explicitly directed to that fact by a Reply (Command).

3.4 Security.

The MPCLA modules will be unaware of any security in the CCS system. The responsibility for all security checking of incoming or outgoing data is the responsibility of the MP-32 system.

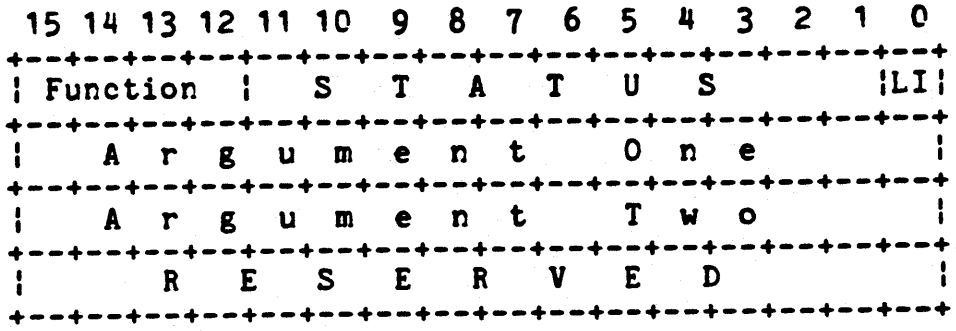
3.5 Controls.

As stated in section 3.3.3 the interface between the MPCLA and the MP-32 is via a command register and the MPCLA low page of RAM memory. This section describes each of these control definitions.

3.5.1 Command Register

The command register resides in the MPCLA low page of RAM memory and is accessable via the MP-32 directory functions to read or write. After IPL this command register is the only vehicle providing control over the MPCLA.

COMMAND REGISTER FORMAT
(One register per Z80)



Format description:

Function	These bits define the function code to be performed by the MPCLA. The legal values for a function are 1 to 16.
Status	These bits are used by the MPCLA to indicate the status of the current operation.
LI	This bit is reserved to indicate which SIO line attached to the requested Z80 the function is associated with. A value of 0 refers to SIO line 0, a value of 1 is for SIO line 1.
Argument One	This argument is always 16 bits long and is generally defined to contain the beginning address of an MP-32 buffer. This address is always a 16 bit half word address. The most significant bit is always set.
Argument Two	This argument is always 16 bits long and is currently defined to contain the LENGTH of a block. This length is always in bytes.

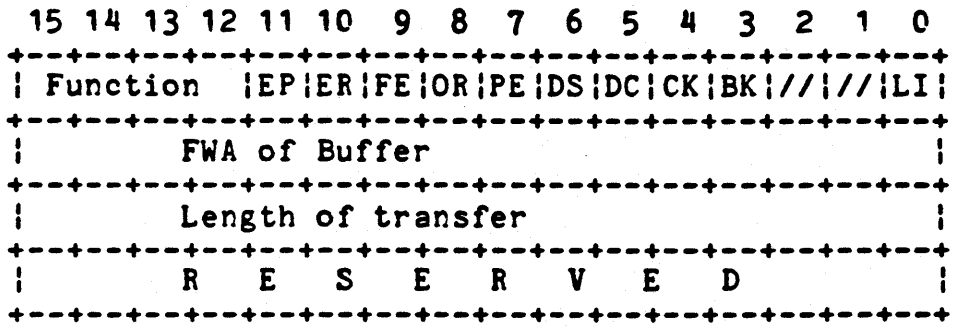
The legal request commands are:

Number		Description
1	Recv Enable	MPCLA to turn on receiver and inform MP-32 of any line changes, including carrier on. The receiver enable also supplies the FWA of a memory buffer for data in argument one.
2	Xmit Enable	MPCLA to turn on transmitter and inform MP-32 of any line changes, including external link not responding.
3	Recv Data	MPCLA to put received data into MPX memory at Argument 1 address.
4	Xmit Data	The MPCLA will start to transmit the data contained at starting address pointed to by Argument One on line stated in Status bit LI. The MPCLA will inform the MP-32 when EOP for this block is sensed.
5	Step	Informs MPCLA to put the Z80 processor in step mode. The Z80 will execute one trip of its master loop. There is no specific Reply to a Step command (other than that the Command Register is cleared). an EOP.
6	Hang	Stops current Z80 process. There is NO response to this request.
7	Reset	The Z80 will stop all transmissions and turn off all external data ports and respond with an EOP to the MP-32 (Any pending transmit or receive requests are ignored by the Z80).
8	Off On	The Z80 will turn off or on the requested Z80 SIO line. Unlike the Reset command any pending transmits or receive blocks which have actually started will be completed.
9	Define Line	Directs the Z80 to initialize communication parameters: baud rate, internal/external clock, data bits and or parity for the SIO line specified by the LI bit. (See 4.2.9 for format of Command Register.)
10	Xmit Pad	Special form of transmit command, where the Z80 appends filler characters to the specified block.

3.5.2 Reply Register.

Like the command register, the Reply Register resides in the MPCLA low page of RAM memory and is accessable via the MP-32 directory functions to read or write. This register is used to advise the MP-32 of completion and status of requested transactions.

REPLY REGISTER FORMAT



Format description:

Function Specifies which Command is being Replied to.
Fwa Echo of Fwa in Command.
Length Specifies Length actually transferred. For an error-free operation, should be equal to the length in the Command.

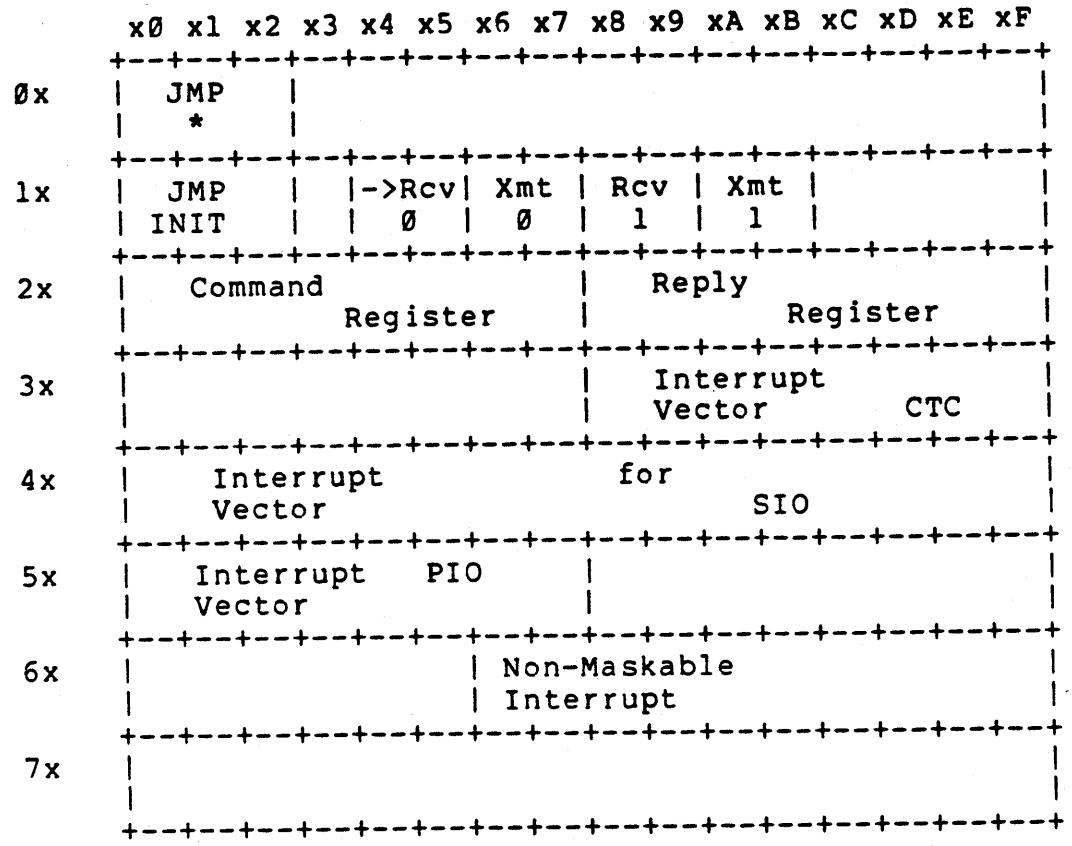
The possible status conditions and their meaning are:

EP End of Operation. If this bit is not set, the operation was prematurely terminated; the data should be discarded or re-transmitted.
ER Error occurred = OR of bits 9 thru 5.
FE Framing Error. Asynchronous = no stop bit present. Synchronous = character synchronization not achieved.
OR Overrun/Underrun. Characters arrived faster than the Z80 could respond; or Z80 was not able to transmit fast enough.
PE Parity Error. Horizontal parity not equal equal to that computed by Z80.
DS Data Set not ready. Modem is not working.
DC Data Carrier has dropped. Probably due to phone line hung-up, or a bad line hit.
CK Checksum error. Longitudinal check value not equal to that computed by Z80.
BK Break detected. Not meaningful in Synchronous operation. Bit is defined only -- not implemented. Reserved for future enhancement.

3.5.3 MPCLA Control page.

The first page of the MPCLA's RAM is dedicated to providing a fixed definitions for both the MPCLA and the MP-32. This page of memory, the Control page, contains all interface words used by both the MP-32 and the MPCLA.

Consult a current listing for exact Page Zero addresses. The general format and layout is as follows:



The definitions for each of the low page cells is as follows:

JMP *

This location is used during autoload time to insure that the MPCLA is running so that memory refresh cycles maintain RAM contents. Locations (0-2) are initially set to be a JMP to self (*) by the MP-32, and the MPCLA started executing at P=0 before the loading process begins. After MPCLA has been loaded by the MP-32 this address will be changed to JMP 10h which fires-up the MPCLA.

JMP INIT

This jump instruction contains the main entry point address of the Z80 code module. The Initializer will change location 0 back into a JMP *.

-> cells

These cells (14h to 1Bh) contain pointers to the SIO receive and transmit line status cells. (See Section 5, MPCLA Internal Specification, for Line Information Table format.)

SECTION 4. EXTERNAL DESIGN DATA

4.1 System Logical Flow.

The MPCLA is a programmable communication line adapter that unlike most commercially available processors uses RAM instead of ROM as the memory for CPU instructions. This feature of the MPCLA allows for dynamic changes in the structure of the the code being executed, but also requires additional internal design structures to be present in order to interface with a host system. The following paragraphs provide a general flow of the MPCLA, including the design structure, with Section 4.2 describing the details of each interface.

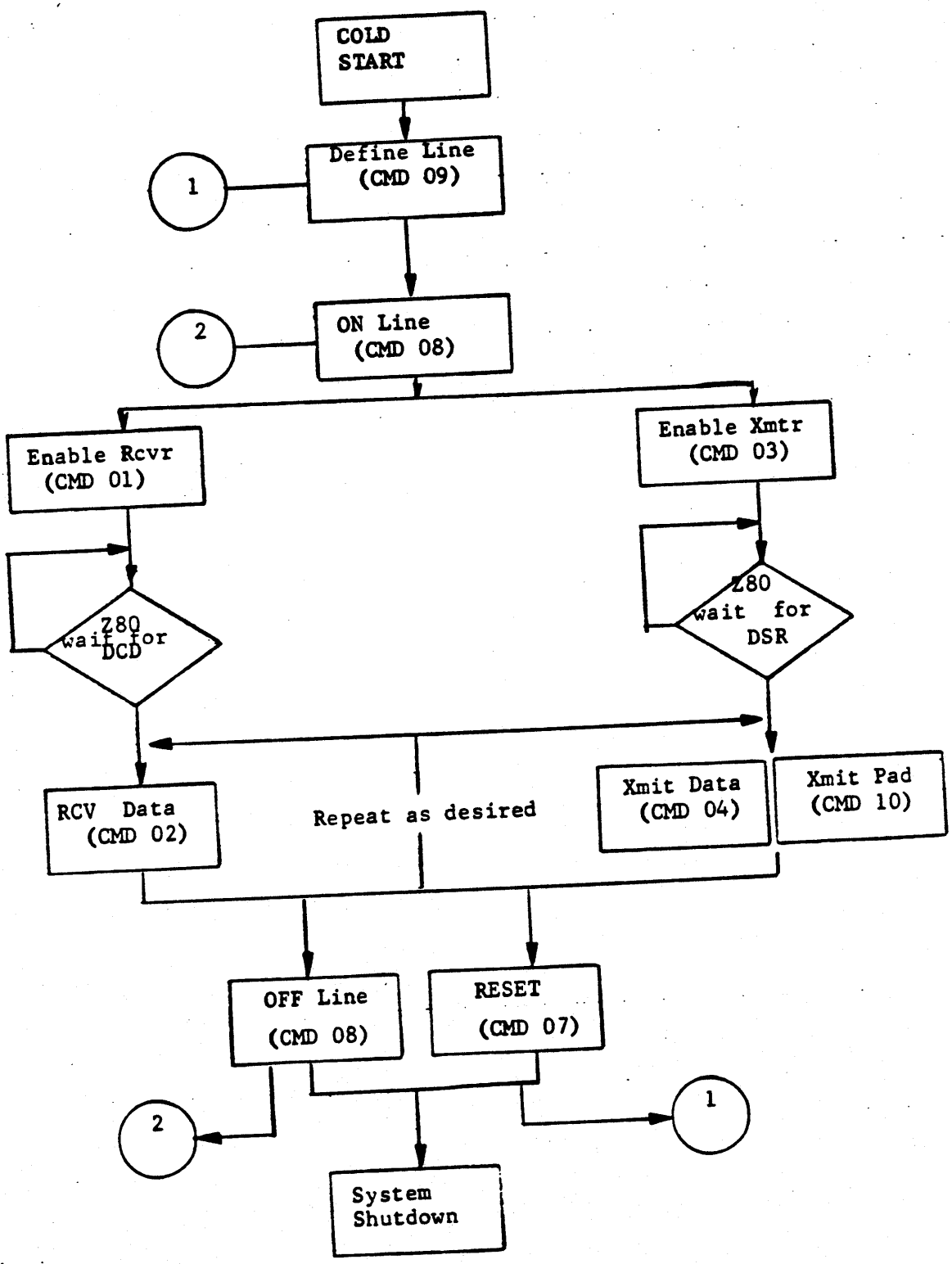
The MPCLA(s) upon system reset or cold start is loaded via MP-32 AQ-directory functions. This load will consist of loading the MPCLA with the appropriate code to process the protocol defined as connected on the various ports. The definition of the port configuration resides in the MPX system. Once the MPCLA is loaded it enters its idle loop awaiting a command from the MPX system. Each MPCLA has a command register located in the MPCLA low page of memory, this command register is fixed and is the main interface between the MPCLA and the MP-32.

The MPCLA after initial load will NOT be capable of receiving or transmitting data. Before the MPCLA can process any external communications data the MP-32 must perform a command function to define the serial interface. This function defines to the MPCLA the baud rate, and the clock for the port requested. Once the port definition has occurred the MPCLA must receive either a Transmit and/or Receive enable request from the MP-32. These requests inform the MPCLA to turn on their appropriate communication lines and to monitor the condition of the line. Once the MPCLA sees a line change it will respond with an EOP indicating line data and conditions.

Once the MPCLA has established line connection it will continue to run as a slave to the MP-32 performing command requests to receive or transmit data.

The MPCLA must also process the Transmit request intiated by the MP-32. This command fires up the transmission of data on the Z80 SIO line defined in the request. The data to be transmitted is defined to be in transmission layout, with pads to fit in 8 bit bytes. The Z80 processor will process the xmit request adding the necessary 'SYNCH' characters, parity and or checksum.

The MPCLA, besides processing data transfers commands, can receive from the MP-32 a second class of commands called control commands. These commands are used by the MP-32 to change the conditions of the MPCLA, or to status the hardware of the MPCLA. A complete description of these commands follows.

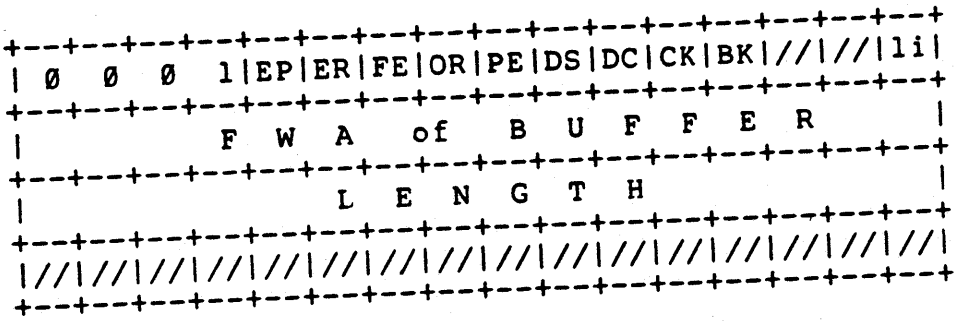


MPCLA Command Logical Flow.

4.2 MPCLA Program descriptions.

The following paragraphs describe in detail the data flow of each Command processor including the definition of the function and all possible replies.

4.2.1 Receive Enable (Command 01)



When the MPX commands a Receive enable, the MPCLA will set up the hardware on the Z80 processor requested, to receive data according to the line descipline. This function can not be performed before a Define-line command has been sent, or if the line is OFF.

If the modem or line is not ready when this command is issued, the Reply is held until the line becomes ready.

4.2.1.1 INPUTS.

The Receive enable command also provides the MPCLA with a FWA of a MP-32 data buffer of maximum length to be used for the next block of data received. Once the Z80 receives a character at the SIO, it will begin saving the data, and transmitting the data to the MP-32 memory in background. All data will be put into the MP-32 memory 'as is', i.e. in transmission line format.

The Z80 processing of the received data is restricted to the following definitions:

AWN
The Z80 processing AWN data will discard the SYNCH bytes, check horizontal parity (MPX ignores parity bits) and validate the segment Checksum. The AWN checksum bytes are set to the difference between the expected and received values, so that the MP-32 can detect which Segment(s) had errors. (A correctly received Segment will have checksum difference = zero.) The buffer word format is as described in Section 3.3.2.

CDC Mode-4A
The Z80 processing of MODE-4A will discard leading and embedded SYN characters, check & clear horizontal parity bit, and test & discard the Longitudinal Parity check (LPC). Parity bit (2**7) of each character will be zeroed. The SOH character is passed on to the MP-32 in order to align the first byte of message data on a (32-bit) word boundary. All bytes (except SYN) from SOH to but not including ETX are passed to the MP-32.

NEDN
The Z80 processing of NEDN will pass received data straight thru. The 'NNNN' or 'nnnn' terminator is passed to MPX. Inbound checksum (8 bit, arithmetic) is validated and discarded.

NMC
The Z80 will discard the DLE,DC1,DLE header and the ETX terminator before passing the block to the MP-32 system. Also, the n*DLE,3*SYN fillers are discarded.

DIAGNOSTIC
The Z80 processing in this mode is defined as pass through. The Z80 will transfer all data through the SIO as defined by the 'define-line' command, NO processing of the data is performed.

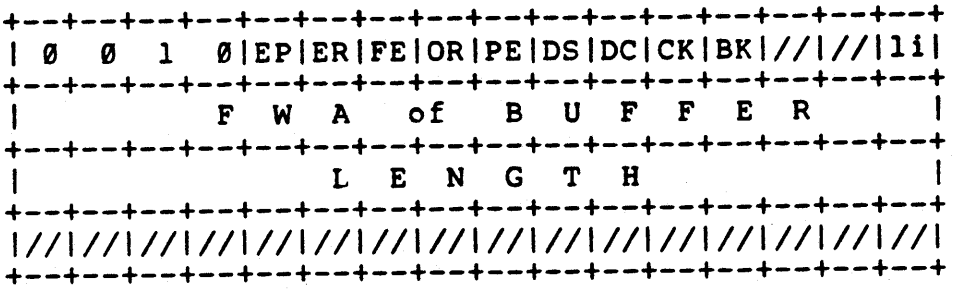
4.2.1.2 OUTPUTS.

Once the MPCLA has received a complete block of information, it will issue an End of Operation (EP) to MPX and pass a status indicating any error encountered.

The possible status conditions and their meaning are:

EP End of Operation.
ER Error occurred = OR of bits 9 thru 5.
FE Framing Error. Asynchronous = no stop bit present. Synchronous = character synchronization not achieved.
OR Overrun. Characters arrived faster than the Z80 could respond.
PE Parity Error. Horizontal parity or longitudinal redundancy check not equal to that computed by Z80.
DS Data Set not ready. Modem is not working.
DC Data Carrier has dropped. Probably due to phone line hung-up, or a bad line hit.
CK Checksum error. The checksum defined by the line protocol is in error.
BK Break detected (Asynch only). Bit reserved - not implemented.

4.2.2 Receive Data (Command 03)



The receiver must have been previously enabled (by Cmd 01) before this command is issued. The Receive Data command is used to read communication blocks subsequent to the first one. Unlike Recv_Enable (01), it does not reset the modem or communication line. Thus, receive overrun and other line problems can be detected.

Except as noted, this command functions identically to Recv_Enable. Refer to Cmd 01 for rest of definition.

4.2.2.1 INPUTS.

Recv_Data commands must be given before the first frame of a communications block arrives at the SIO.

4.2.2.2 OUTPUTS.

A Reply 03 is issued when a complete block has been received.

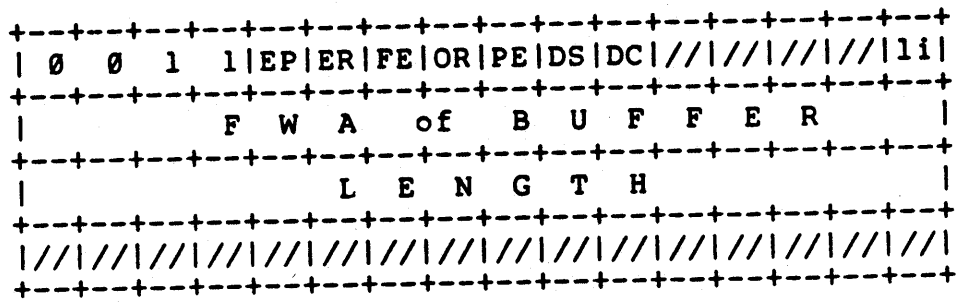
In addition to the status conditions of Cmd 01, the following:

OR Overrun. Characters arrived faster than the Z80 could respond. Or, a communications block began before the Recv_Data command was issued. The MPCLA subsystem cannot begin to process an arriving message until it has an MPX memory area in which to put the data.

DS Data set (modem) dropped Ready status subsequent to receipt of the previous block.

DC Data Carrier has dropped since the previous block. (Except for Controlled Carrier disciplines.)

4.2.3 Transmit Enable (Command 02)



When the MPX commands Transmit Enable, the MPCLA will set up the hardware in condition to transmit according to the line discipline. This function can not be performed before a Define_Line (09) has been done. After a successful Transmit_Enable, the MPX can issue Transmit_Data (04) and Transmit_Pad (10) commands.

For Constant Carrier lines, the carrier is turned on by Cmd 02. (All CCS lines are currently specified as Constant Carrier.) The MPCLA switches the transmitter from the "Off Line" state into "On Line Idle", and waits for the modem to become ready, and conditions the line, as necessary.

4.2.3.1 INPUTS.

No data is passed with the Transmit_Enable. The FWA and LENGTH parameters should be zero, to allow for future extension.

4.2.3.2 OUTPUTS.

A Reply 02 is issued when setup is complete (or failed).

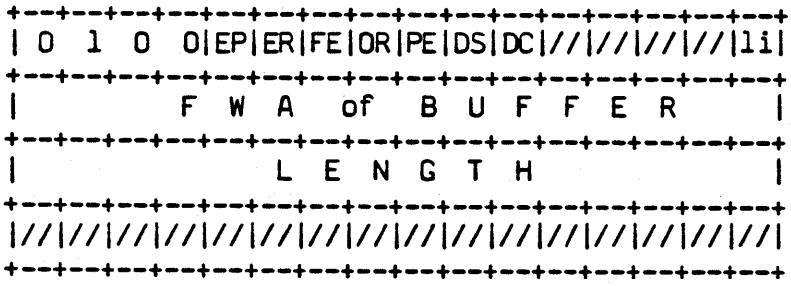
The possible status conditions and their meaning are:

EP End of Operation.
ER Error occurred = OR of bits 9 thru 5.
DS Data Set not ready. Modem is not working.
DC Data Carrier did not come up properly. Probably due to phone line hung-up, or a bad line hit.

4.2.3.3 DATA BASE.

none.

4.2.4 Transmit Data (Command 04)



The transmitter must have been previously enabled before this command is issued. The MPCLA switches the transmitter from the "On Line Idle" state into "Active" (needed, for example, for Controlled Carrier disciplines), initializes longitudinal check computations, and ensures that the message is preceded by a sufficient number of character-synchronization frames, as defined for this line.

4.2.4.1 INPUTS.

The data supplied with the Transmit command must constitute one complete communication block, as required by the protocol. The Z80 transmits the data unchanged, subject to only the following transformations:

AWN
Horizontal parity is completed as detailed in Section 3.3.2. The segment Checksum is generated and transmitted at the end.

CDC Mode-4A
The SOH character must be generated by the MP-32. The MPCLA appends the ETX character, as well as the Longitudinal Parity Check (LPC). The most sigificant bit of each data byte is replaced by horizontal parity.

NEDN
Data is transmitted unchanged. Message Pad of 50 bytes of binary zero is transmitted after the last block of a sequence (at "NNNN" EOM). There is no outbound checksum.

NMC
The Z80 will add the necessary DLE,DC1,DLE header and ETX terminator to all blocks. The 3*SYN group is inserted every 188 bytes.

that the block begins with sufficient Synch codes. The Z80 will inhibit the normal SIO CRC transmission, so that the MP-32 is in complete control of the transmitted block.

4.2.4.2 OUTPUTS.

In order to improve transmit timing, a Transmit Data command generates two replies:
1) After the data block has been read into Z80 RAM, a "Transmit Data Block Copied "reply (EP=0), is sent to the MP-32. This signifies that the Z80 has no further need for the MP-32 memory image of the data, and, further, that the MP-32 may now issue another Transmit Data command.
2) When the block is completely transmitted, a "Transmit EOP" Reply (EP=1) is sent to the MP-32. The status field of the EOP indicates whether the block was successfully transmitted, or whether an error occurred.

Note that in unusual circumstances (for example, Line Down), the transmit operation may be terminated before the "Copied" reply is generated. When this happens, the EOP reply will be the only one given. The MP-32 should be able to cope with an omitted "Copied" reply. However, if multiple transmits are outstanding, each one will get an EOP reply.

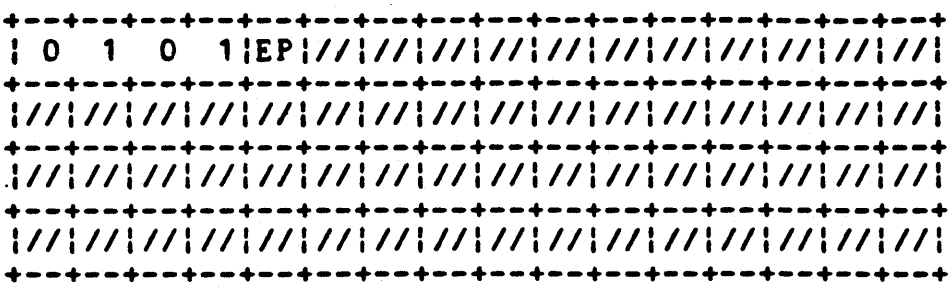
The possible status conditions and their meaning are as follows:

EP	End of Operation.
ER	Error occurred = OR of bits 9 through 5.
FE	Framing Error. Asynchronous = no stop bit present. Synchronous = character synchronization not achieved.
OR	Underrun. Z80 got behind transmitter.
PE	Parity Error. Horizontal parity or longitudinal redundancy check not equal to that computed by Z80.
DS	Data Set not ready. Modem is not working.
DC	Data Carrier has dropped. Probably due to phone line hung-up, or a bad line hit.

4.2.4.3 DATA BASE.

none.

4.2.5 Step (Command 05)



The Step command causes the MPCLA to enter "Step mode". While in Step mode, the MPCLA will wait for another Step command before generating each Reply interrupt to the MP-32.

The Step command is not compatible with normal system operation. Communication lines attached to a Z80 which is being Stepped may not be serviced properly. This command should only be used for diagnostic purposes, or if on-line reconfiguration is attempted.

The MPCLA will be capable of honoring a Hang command (06) while it is waiting for a next-Step command.

4.2.5.1 INPUTS.

The EP bit in the Command indicates whether to begin or terminate Step mode. If EP=0 the MPCLA enters Step mode. If EP=1 the MPCLA resumes normal high speed operation.

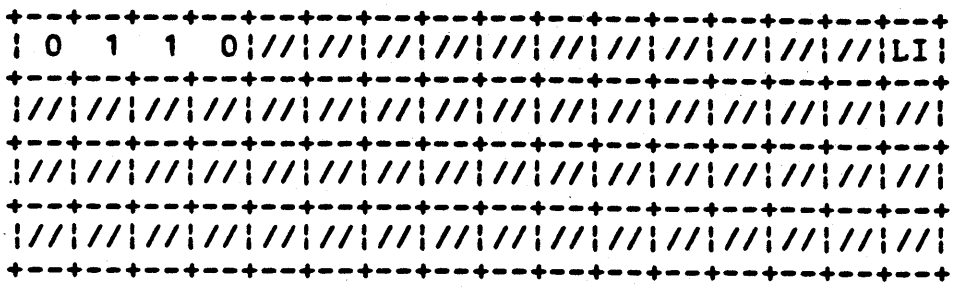
4.2.5.2 OUTPUTS.

There is no Reply to the Step command, as such. However, after each Step command is issued, the MPCLA may issue a pending Reply to some other command.

4.2.5.3 DATA BASE.

None.

4.2.6 Hang (Command 06)



The Hang command forces the MPCLA to stop all execution. The MPCLA will stop execution and jump to location zero. Location zero will have been restored to contain JMP *, by MPCLA initialization. Thus, after issuing a Hang command, Z80 RAM refresh will still be operating so the MP-32 can read/dump MPCLA memory without using destructive AQ-director functions.

This command is not compatible with normal system operation. Any transactions in progress when the MPCLA is stopped will not be completed. Active communication lines will no longer be serviced, nor are blocks properly terminated. The MPCLA must be restarted in order to resume communication service.

NOTE - This function will never be honored if the Z80 is looping with interrupts disabled. (The MPCLA code will set location 66h such that a Non-Maskable Interrupt will also jump to location zero. This can be used if the NMI/ pin of the Z80 is accessible.)

4.2.6.1 INPUTS.

There are no other input fields for this command except the command itself.

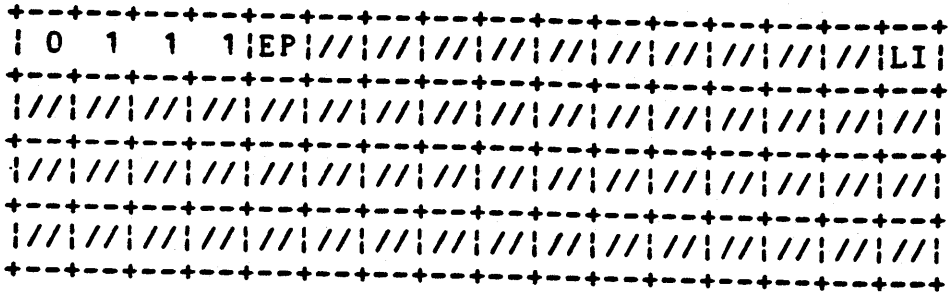
4.2.6.2 OUTPUTS.

There is NO output for this command (The processor is stopped). However, all of Z80 memory is intact and available to the MP-32.

4.2.6.3 DATA BASE.

None.

4.2.7 Reset (Command 07)



The RESET command is executed upon request and releases any and all pending command requests (including transmit or receive).

4.2.7.1 INPUTS.

None.

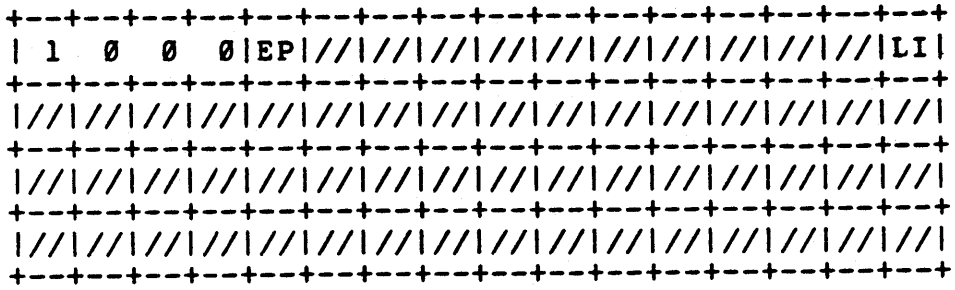
4.2.7.2 OUTPUTS.

The MPCLA will send a Reply with the EP bit set when it has completed its clean-up and is ready to receive a new command. The MPCLA state after receiving a RESET command is the same as that after an initial load.

4.2.7.3 DATA BASE.

None.

4.2.8 OFF-ON Request (Command 08)



The OFF/ON command informs the Z80 to change the condition of the requested line. This process is NOT performed until all transmit and receive queued requests have been completed.

4.2.8.1 INPUTS.

LI Specifies which line is to be affected.

EP 0= Turn line ON. 1= Turn line OFF.

4.2.8.2 OUTPUTS.

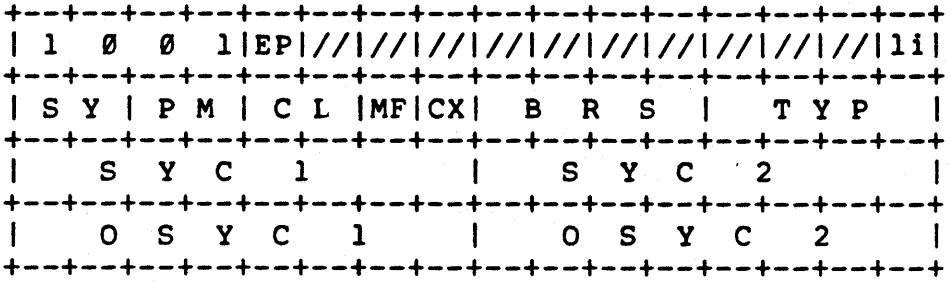
The MPCLA will send a Reply with the EP bit set after this function request is complete.

4.2.8.3 DATA BASE.

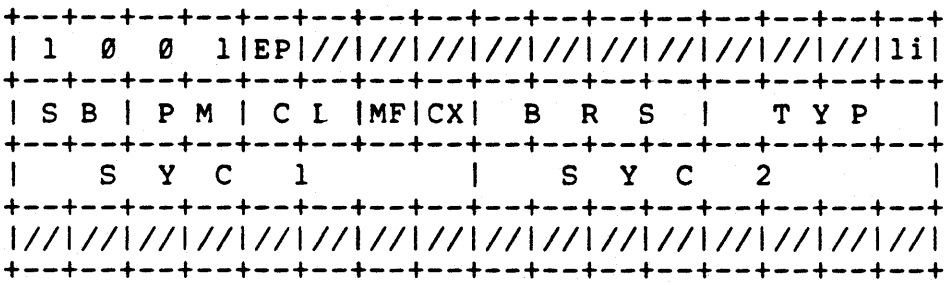
After an OFF command is completed, the specified line is Disabled. Commands 01 and/or 02 must be issued to resume traffic.

4.2.9 Define Line (Command 09)

SYNCHRONOUS Select Format:



ASYNCHRONOUS Select format:



The Define_Line (DF) command request sets the operating mode of the SIO line requested (0 or 1) for the Z80. This command is required only one per line following power-on or a Master Reset. If this command is issued while the Z80 is still processing data unknown results will occur. It is the responsibility of the MP-32 to insure that this command is used properly.

4.2.9.1 INPUTS.

The Define_Line command selects the line discipline for the MPCLA including the definition of whether the line is Asynchronous or Synchronous. The Define-line command word field definitions for Synchronous are as follows:

SY — Define SYNCH type. This field defines the number of internal SYNCH characters or external synch select. The legal values are:

- 0 — One synch character. Definition defined by SYC1 field
- 1 — Two synch characters. Definition defined by SYC1 and SYC2 field.
- 2 — RESERVED.
- 3 — External synch control. Synchronization is achieved by applying a level on the SYNDET pin.

PM — Parity mode. This field defines the parity

mode the SIO line is to execute in. The legal values are:

00	NO parity.
01	ODD parity.
10	(not used)
11	EVEN parity.

CL — Character length. This field defines the character length of the SIO line. The legal values are:

00(0)	5 bits
01(1)	7 bits
10(2)	6 bits
11(3)	8 bits.

CX — Clock select. In Synchronous mode the bit data rate is equal to the selected clock rate. If this field is zero the internal baud rate clock will be used, see BRS field specification. If CX=1, the clock is supplied by the external line via RS-232C pins 15 and 17.

BRS — Baud Rate Select. The SIO line, if running on an internal clock, will select the appropriate baud rate defined by this field. (The column labeled "Actual" is the 1x clock rate for Synchronous CX=0.) The legal values are:

BRS Value	Nominal Baud	Actual Clock = Nominal x 16
0	50 bps	0.8 kHz (bps=Hz)
1	75	1.2
2	110	1.76
3	134.5	2.152
4	150	2.4
5	300	4.8
6	600	9.6
7	1200	19.2
8	1800	24.7438
9	2000	31.9168
10	2400	38.4
11	3600	57.8258
12	4800	76.8
13	7200	114.306
14	9600	153.6
15	19200	307.2

TYP — This field defines the protocol to be used for the port selected. The currently defined protocols and their value is:

0	(DIGICON DEBUG package)

	1	Diagnostic mode
	2	AWN
	3	NEDN
	4	Mode 4A
	5	NMC
	6	(Reserved for ASynch)
	7	(Reserved for ID50)
	8-15	(Reserved)

MF — The MF field equals 0 for Synchronous definition.

SYC1/2 — Synch words. These bytes define the two possible SYNCH characters that are to be used when the EX definition is internal.

OSYC1/2 — Outbound Synch words. These bytes provide for the case where the outbound Synch differs from inbound (AWN only). The MPCLA sets them equal to SYC1&2 for normal protocols.

If the MF field is set to a 1, the define-line is requesting that the SIO line for this Z80 run in Asynchronous mode. The field definitions for Asynchronous mode are:

SB — Stop bits. The legal values are:

00(0)	Invalid
01(1)	One (1) Stop bit
10(2)	1 1/2 Stop bits
11(3)	2 Stop bits

PM — Parity mode. This fields definition is the same as Synchronous.

CL — Character length. This fields definition is the same as Synchronous.

CX — Clock Select. If CX=0, clock is internally supplied, and BRS field specifies the rate. If CX=1, the clock is supplied externally via RS-232 pins 15 and 17; the external clock is divided as specified by BRS to determine the bit data rate.

BRS — Baud rate select. When CX=0, the data rate times 16 = clock rate, and the values of BRS are given above under Synchronous in the column labelled "Nominal".
When Asynch CX=1, the external clock divisor is specified by BRS:
Clock rate = Data rate if BRS=0; However RxD and RxC must maintain proper phasing, as detailed in SIO manual, reference [10].

Clock = 16x Data if BRS=1.
Clock = 32x Data if BRS=2.
Clock = 64x Data if BRS=3.

MF — Mode flag. Asynchronous select is MF=1. When the Z80 is in Asynchronous mode all lines are considered half duplex, it is the responsibility of the MP-32 to echo in full duplex mode.

TYP — Must be = 0, 1, or 6 for Asynchronous line.

SYC1/2 — Termination characters for Asynchronous receive. Normally set to Ascii CR.

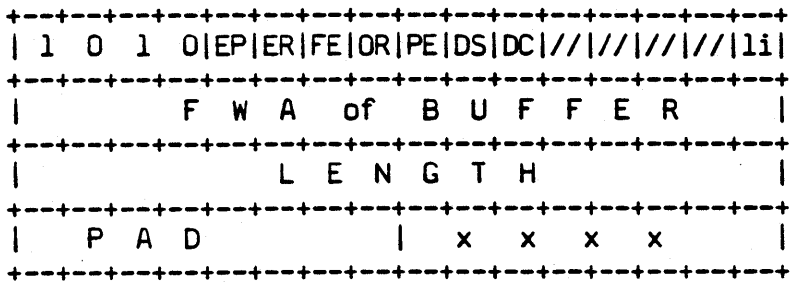
4.2.9.2 OUTPUTS.

The MPCLA will reply with an EOP set upon completion of processing the DF request. The DF reply will raise the interrupt line to the MP-32 and set EOP, the DF command does not have any error conditions.

4.2.9.3 DATA BASE.

The parameters specified in the Command Register are set up in the SIO control registers, and the Line Information Table (see Section 5).

4.2.10 Transmit Pad (Command 10)



This command enables the MP-32 to send predefined handshake messages conveniently and efficiently. Several different padding options are provided, as detailed below. The transmitter must have been previously enabled before this command is issued. Transmitter initiation is the same as for normal Transmit Data (Command 4).

4.2.10.1 INPUTS.

There must be at least one byte of data supplied by the MP-32. The Z80 suffixes the MP-32 data with internally generated bytes to form a complete transmission block. The MP-32 data is transmitted as per Transmit Data Command (4). The value of the PAD field determines which predefined pattern is used. Each protocol processor supports only the PAD options which are useful with that protocol.

AWN
No PAD options.

CDC Mode-4A.
No PAD options.

NEDN
No PAD options.

NMC
No PAD options.

DIAGNOSTIC
The entire data block is transmitted unchanged. No PAD options.

4.2.10.2 OUTPUTS.

Unlike the normal Transmit Data command (4), the Transmit Pad command does not generate an early reply ("Transmit Data Block Copied"). Thus, this command should only be used for short data lengths, or when no further transmission is contemplated. When the block is completely transmitted, a normal "Transmit EOP" reply is sent to the MP-32. See 4.2.4.2 for other reply and status definitions.

4.2.10.3 DATA BASE.

None.

Appendix B

System Design and Requirements-Software

Intentionally Omitted

Appendix C

Glossary

Glossary

Abort	The premature termination of a process whenever an irrecoverable situation occurs.
Absolute	Refers to actual machine address.
ASCII	American Standard Code for Information Interchange. The standard code, using a coded character set consisting of 7-bit coded characters. See American National Standard X3.4-1977.
Assemble	The process by which an object (binary) module is created from the symbolic language program.
Asynchronous	Refers to a type of serial transmission in which bit synchronization is accomplished for each character.
Batch	Class of tasks which run on a time available basis.
BCLA	Buffered Communications Line Adapter is the controller interface to the RS-232C asynchronous communications lines.
Bit	A binary digit.
Block	A group of machine words or bytes. Usually a collection of one or more records used in I/O to reduce the number of physical operations.
Buffer	A portion of memory used to collect data in order to compensate for speed differences between the processor and peripheral devices.
Byte	Eight (8) contiguous bits.
CALL	The tranfer of control to a closed routine or task. A Executive Service Request, CALL, is used to activate a specific task.
Callee	The task initiated by the CALL Executive Service Request.

Caller	The task initiating another routine or task.
CLA	Communication Line Adapter, the hardware interface between the CPU and the communications line.
Compile	The process by which an assembly module is created from a problem solving language. A compiler usually generates several machine instructions from a single symbolic statement.
Common	An area of memory that may be shared between tasks or routines. Tasks may communicate through common areas.
Control Point Zero	CYBER System Control Point.
CPU	Central Processing Unit.
CRC	Cyclic Redundancy Check.
Data	An area of memory that may be prestored with information at load time and may be shared between routines of a task.
Dispatcher	An operating system routine that unthreads a task from the top of the ready list and places the task into execution.
ESR	Executive Service Request, name given to the MPX Operating System routines invoked by the MON instruction.
File	A collection of blocks and/or records, usually of related data. Each mass storage file has an entry in the system Label File.
Interrupt	A break in the normal processing flow usually caused by a hardware signal. Interrupts can be enabled or disabled and occur with an associated priority. Processes that are interrupted are later resumed at the point of interruption.

ITS	Interactive Terminal Subsystem.
IOC	Input/Output Controller.
Job	The sequential and/or parallel execution of tasks. Begins with a *JOB card and ends with a *EOJ card.
JCT	Job Control Table is an area of memory containing information controlling a given job.
Job Manager	A system task that processes the input stream of the job. The Job Manager is a set of reentrant routines shared by all user jobs.
Library	A collection of frequently used, checked-out programs maintained on an external device that can be loaded and executed separately or in conjuction with a user's program. Libraries must be arranged to minimize searching.
Linkage	The interconnection between routines or devices. The loader matches externals and entry points to establish program linkage. The Interactive Terminal Subsystem match job terminal requests with terminal user connection requests to establish port linkage.
Loader	A system task that is used to load, relocate, and link binary object modules.
LU	Logical Unit, a number representing the user connection to a device.
MPX/OS	The MPX Operating System.
MP-32	MP-60 CPU Emulation device.
MPP	MP-60 CPU Emulation device (Militarized, or Ruggidized).

Ordinal	The relative location of an entry in a table. The absolute location of an entry can be obtained by multiplying the ordinal by the number of machine words per entry and adding the starting address of the table.
Page	A 4096 word block of contiguous memory. Paging is a technique where a logical address is transformed via a set of page registers into a physical address.
PMAC	Programmable Multiple Access Controller.
Port	The communications line between the CPU and the user.
PPU	CDC CYBER/6000 Peripheral Processor Unit.
Process	A software program equivalent to the MPX/OS terms tasks and job.
Priority	A value assigned to an item in the system which facilitiates scheduling and processing within the operating system.
Queue	A list used to control the processing to be done.
Ready list	A prioritized list of tasks waiting for control of the CPU.
Relocatable	Refers to a program that has been prepared by a source language compiler or assembler to be loaded intc any area of available memory.
Resident	The portion of the operating system which resides permanently in memory.
RETURN	A Executive Service Request that terminates a task and transfers control to the point in the caller where the call originated. A task may return with or without release of memory.
Schedule	The determination of which processing is required next.

Stack	A last-in, first-out queue.
Status	A stage or condition of an I/O request or a task.
Synchronous	Serial transmission in which characters are sent bitwise without start and stop bits.
System Initialize	Refers to the intial system load process where the resident is loaded, memory initialized, and the system tasks are started.
Task	An independent unit of work that can compete for the resources of the system. A task may call ard be called by other tasks.
TCT	Task Control Table is an area of memory containing information used to control a task.
Terminal	The device connected to the user end of the PORT.
Termirate	The process of completing a job. A job may terminate normally or abnormally.
Thread	A linked list of elements, the contents of each thread cell contains the address of the next thread cell and so on until a thread cell contains a value indicating the end of the list.
Utility	A routine, procedure or program that supports the operation of a system.

Appendix D

Conversion Tables

Appendix D Conversion Tables

ASCII HEX	ASCII CHAR	DC OCT	DC CHAR	EXT BCD OCT	BAUDOT OCT	FLEET BCD OCT	FLEET BCD CHAR
0	NUL						
1	SOH						
2	STX						
3	ETX						
4	EOT						
5	ENQ						
6	ACK						
7	BEL				F5	32	]
8	BS						
9	HT						
A	LF				F2/L2	0	
B	VT						
C	FF						
D	CR				F10/L10	13	=
E	SO						
F	SI						
10	DLE						
11	DC1						
12	DC2						
13	DC3						
14	DC4						
15	NAK						
16	SYN						
17	ETB						
18	CAN						
19	EM						
1A	SUB						
1B	ESC						
1C	FS						
1D	GS						
1E	RS						
1F	US						
20	SPCE	55	SPCE	20	L4/F4	20	SPCE
21	!	66	!	52	F15	66	F
22	"	64	"	14			
23	#	60	#	36	F24	70	H
24	$	53	$	53	F11	64	D
25	%	63	%	16			
26	&	67	&	37	F32	67	G
27	'	70	'	55		55	*
28	(	51	(	34	F17	42	K
29	)	52	)	74	F22	43	L

	2A	*	47	*	54								
	2B	+	45	+	60		F21	60	+				
	2C	,	56	,	33								
	2D	-	46	-	40		F3	61	A				
	2E	.	57	.	73		F34	73	.				
	2F	/	50	/	21		F35	21	/				
	30	0	33	0	12		F26	12	0				
	31	1	34	1	1		F27	1	1				
	32	2	35	2	2		F23	2	2				
	33	3	36	3	3		F3	3	3				
	34	4	37	4	4		F12	4	4				
	35	5	40	5	5		F5	5	5				
	36	6	41	6	6		F25	6	6				
	37	7	42	7	7		F7	7	7				
	38	8	43	8	10		F6	10	8				
	39	9	44	9	11		F30	11	9				
	3A	:	0	:	0		F16	63	C				
	3B	;	77	;	77		F36	53	$				
	3C	<	72	<	72								
	3D	=	54	=	13								
	3E	>	73	>	57								
	3F	?	71	?	56		F31	62	8				
	40	@	74	@	15								
	41	A	1	A	61		L3	61	A				
	42	B	2	B	62		L31	62	B				
	43	C	3	C	63		L16	63	C				
	44	D	4	D	64		L11	64	D				
	45	E	5	E	65		L1	65	E				
	46	F	6	F	66		L15	66	F				
	47	G	7	G	67		L32	67	G				
	48	H	10	H	70		L24	70	H				
	49	I	11	I	71		L6	71	I				
	4A	J	12	J	41		L13	41	J				
	4B	K	13	K	42		L17	42	K				
	4C	L	14	L	43		L22	43	L				
	4D	M	15	M	44		L34	44	M				
	4E	N	16	N	45		L14	45	N				
	4F	O	17	O	46		L30	46	O				
	50	P	20	P	47		L26	47	P				
	51	Q	21	Q	50		L27	50	Q				
	52	R	22	R	51		L12	51	R				
	53	S	23	S	22		L5	22	S				
	54	T	24	T	23		L20	23	T				
	55	U	25	U	24		L7	24	U				
	56	V	26	V	25		L36	25	V				
	57	W	27	W	26		L23	26	W				
	58	X	30	X	27		L35	27	X				
	59	Y	31	Y	30		L25	30	Y				
	5A	Z	32	Z	31		L21	31	Z				

5B	[	61	[	17							
5C	\	75	\	75							
5D	]	62	]	32							
5E	~	76	~	76							
5F	_	65	_	35							
60	~										
61	a	1	A	61		L3		61		A	
62	b	2	B	62		L31		62		B	
63	c	3	C	63		L16		63		C	
64	d	4	D	64		L11		64		D	
65	e	5	E	65		L1		65		E	
66	f	6	F	66		L15		66		F	
67	g	7	G	67		L32		67		G	
68	h	10	H	70		L24		70		H	
69	i	11	I	71		L6		71		I	
6A	j	12	J	41		L13		41		J	
6B	k	13	K	42		L17		42		K	
6C	l	14	L	43		L22		43		L	
6D	m	15	M	44		L34		44		M	
6E	n	16	N	45		L14		45		N	
6F	o	17	O	46		L30		46		O	
70	p	20	P	47		L26		47		P	
71	q	21	Q	50		L27		50		Q	
72	r	22	R	51		L12		51		R	
73	s	23	S	22		L5		22		S	
74	t	24	T	23		L20		23		T	
75	u	25	U	24		L7		24		U	
76	v	26	V	25		L36		25		V	
77	w	27	W	26		L23		26		W	
78	x	30	X	27		L35		27		X	
79	y	31	Y	30		L25		30		Y	
7A	z	32	Z	31		L21		31		Z	
7B	{										
7C	|									
7D	}										
7E	~										
7F	DEL										
	SKIP					F0/L0		0			
	FIG					F33/L33		0			
	LET					F37/L37		0			
	ORC					F14		45		N	
	DNA					F13		41		J	

FIG = FIGURES (F) LET = LETTERS (L)
ORC = OR-CIRCLE DNA = DOWN ARROW

Appendix E
AWN High Speed Interface
Intentionally Omitted

Appendix F

Interface Definition Document

COMMENT SHEET

TITLE: MP-32 Computer Systems

Consolidated Communications System
System Specification

PUBLICATION NUMBER: CCSYSTM-022-SS REVISION: A

NAME:

COMPANY:

STREET ADDRESS:

CITY: STATE: ZIP CODE:

Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND TAPE

TAPE

TAPE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION

Systems Technology Division
215 Moffett Park Drive
Sunnyvale, California 94086

CUT ALONG LINE

FOLD

FOLD