Gould Eight-Line Asynchronous Communications Multiplexer Model 8512-2 Technical Manual

January 1985

Publication Order Number: 303-006180-000

GOULD
Electronics

This manual is supplied without representation or warranty of any kind. Gould Inc., Computer Systems Division therefore assumes no responsibility and shall have no liability of any kind arising from the supply or use of this publication or any material contained herein.

HISTORY

The Gould Eight-Line Asynchronous Communications Multiplexer, Model 8512-2 Technical Manual, Publication Order Number 303-006180-000, was printed January 1985.

This manual contains the following pages:

Title page
Copyright page
iii/iv through x
1-1 through 1-12
2-1/2-2
3-1 through 3-18
4-1 through 4-42
A-1 through A-3/A-4
IN-1 through IN-4

CONTENTS

ILLUSTRATIONS

TABLES

CHAPTER 1

INTRODUCTION

1.1 General Description

The Model 8512-2 Gould Eight-Line Asynchronous Communications Multiplexer controls communication between the central processing unit (CPU) of the Gould 32/7X Series computer, or Gould CONCEPT/32™ computers, and up to eight terminals and/or modems. Communication between the CPU and the external I/O device takes place through the Model 8000 or 8001 Input/Output Processor (IOP) via the multipurpose bus (MP Bus).

The multiplexer handles asynchronous data formats with five to eight bits per characters; with one, one and a half, or two stop bits; and even, odd, forced, or no parity.

The Eight-Line Asynchronous Communications Multiplexer interfaces with peripheral devices, with RS-232C or 20-milliampere current loop interfaces, via a distribution panel. Models 8580 and 8582 distribution panels are available to convert the signal levels and pinouts to the RS-232C or the 20-milliampere current loop protocols.

1.2 Features

The Eight-Line Asynchronous Communications Multiplexer has the following features:

1. Controls up to eight terminals and/or modems.
2. Sixteen data transmission speeds up to 19,200 bits per second.
3. Peak aggregate transfer rate of 76,800 bits per second and an average aggregate rate of 40,000 bits per second.
4. Selectable controller address.
5. Internal loopback mode for diagnostic testing.
6. Modem control.
7. Type ahead.
8. Full or half duplex.
9. Software configurable on a per line basis.

1.3 Applicability

The Eight-Line Asynchronous Communications Multiplexer can be used on the Gould 32/7X Series computers, and on CONCEPT/32 computers, designed and manufactured by Gould Inc., Computer Systems Division, which meet the required prerequisites.

1.4 Prerequisites

The computer must be equipped with the IOP. The Eight-Line Asynchronous Communications Multiplexer requires one multipurpose bus backplane slot.

1.5 Configurations

The following configurations are available with the Eight-Line Asynchronous Communications Multiplexer:

1. EIA RS-232C physical protocol.
   a. Model 8580 Distribution Panel Assembly.

   b. Model 8590 Cable P/N 145-103020-4XX - Interface cable between distribution panel and customer modem.

   c. Cable 145-103020-3XX - Interface between the distribution panel and CRT terminal (Televideo 910, Hazeltine 1500, or ADM-3).

   d. Cable 144-103416 - Two 60-pin flat cables for interfacing the eight-line asynchronous communications multiplexer and the Model 8580 Distribution Panel.

   e. Cable 145-103071-XXX - Flow Control Cables.

2. Twenty-milliampere current loop physical protocol.

   a. Model 8582 Distribution Panel Assembly.

   b. Cable 144-103437-001 - Power cable between eight-line asynchronous communications multiplexer and distribution panel.

   c. Cable 145-103020-2XX - Interface between Model 8582 Distribution Panel and the ADM-3 CRT terminal.

   d. Cable 145-103020-1XX - Interface between the Model 8582 Distribution Panel and the Televideo 910 or Hazeltine 1500 CRT terminal.

   e. Cable 144-103004-024 - Two 40-pin flat cables for interfacing between eight-line asynchronous communications multiplexer and the Model 8582 Distribution Panel.

1.6 Specifications

Table 1-1 lists the physical, electrical, and environmental specifications of the Eight-Line Asynchronous Communications Multiplexer.

1.7 Physical Description

The Eight-Line Asynchronous Communications Multiplexer is comprised of the eight-line asynchronous communication multiplexer printed wiring (PW) board, 160-103768 and one of two distribution panels, Model 8580 or Model 8582.

The board contains a Z80® CPU, peripheral logic to interface between the multipurpose bus and the peripheral device, programmable read only memories (PROMs) to store the control program, random access memories (RAMs) for data and parameter storage, and eight asynchronous communication elements (ACE) for data and signal control over the output lines. The ACE is also referred to in this manual as a Universal Asynchronous Receiver/Transmitter (UART).

The eight-line asynchronous board occupies one backplane slot in the I/O chassis and is interconnected to the distribution panel by two 40-pin flat ribbon cables and one 14-pin power cable for the 20-milliampere current loop operation, and by two 60-pin flat ribbon cables for the RS-232 operation.

The eight-line asynchronous distribution panel is mounted on the cabinet panel mounting rails and provides eight serial bidirectional ports for connection to peripheral devices.

The Eight-Line Asynchronous Communications Multiplexer is supplied with one of two distribution panels, depending on the physical protocol that is to be supported.

1.8 Functional Description

The following is a brief functional description of a communication system using the Eight-Line Asynchronous Communications Multiplexer.

1.8.1 Communication System

The block diagram of Figure 1-1 shows the data flow of a communication system using the Eight-Line Asynchronous Communications Multiplexer. The MP Bus is the interface between the IOP and the multiplexer. Information from the IOP is applied to eight parallel data lines of the MP Bus. The multiplexer accepts the data using the three-wire handshake protocol and performs a parallel-to-serial conversion. It also transmits the data (P0TXDAT through P7TXDAT) from the selected port (port 0 through port 7) to a distribution panel.

The distribution panel provides the signal levels and pinouts which are compatible with the protocol of the terminal or modem. The information is transferred to the appropriate terminal or modem, as determined by the selected port, using the established protocol. Information transfers can be initiated by the terminal or modem using the established protocol of the device. The distribution panel accepts the serial information on one of the receive data lines (RXDAT0 through RXDAT7).

The information is sent to the multiplexer on one of the serial port receive data (P0RXDAT through P7RXDAT) lines. The multiplexer performs a serial-to-parallel conversion and transmits the information to the IOP on the eight parallel data lines of the MP Bus, using the three-wire handshake protocol.

®Z80 is a registered trademark of Zilog, Inc.

Table 1-1
Eight-Line Asynchronous Communications Multiplexer Specifications

Characteristic	Specification
Physical	
PC Board	
Length	15 in. (38.1 cm)
Width	12 3/4 in. (32.2 cm)
Mounting	One backplane slot
Distribution Panels	
Length	4 in. (10.2 cm)
Width	19 in. (48.3 cm)
Height	3.5 in. (8.9 cm)
Mounting	Industry standard, 19-inch panel width with EIA hole spacing
Environmental	
Operating	
Temperature	0° to 50°C (32° to 122° F)
Relative humidity	20% to 80%, noncondensing
Storage	
Temperature	-25° to 70°C (-12.8° to 158°F)
Relative humidity	5% to 95%, noncondensing
BTUs/Min.	220
Electrical	
Voltage	+5 vdc -5 vdc +15 vdc, with Model 8582 Distribution Panel
Current	2.5 A at +5 vdc 100 mA at -5 vdc, with Model 8582 Distribution Panel 60 mA at -5 vdc, without Model 8582 Distribution Panel 320 mA at +12 vdc, with Model 8582 Distribution Panel
Performance	
Data Format	Asynchronous
Characters	5, 6, 7, or 8 bits
Stop bits	1, 1.5, or 2 bits
Parity	Even, odd, forced, or no parity
Data throughput	
Total (eight lines)	76,800 bits per second, peak (function of operational mode) 40,000 bits per second, average
Per line	19,200 bits per second, maximum
Protocol (Interface)	RS-232C and 20-milliampere current loop

1.8.2 Maximum Configuration

Figure 1-2, the communication system maximum configuration, shows the interconnection between the IOP and the terminals/modems, using up to 7 Eight-Line Asynchronous Communications Multiplexers.

The MP Bus is the interface between the IOP and up to 7 multiplexers. Each multiplexer can support eight port transmit data (P0TXDAT through P7TXDAT) lines to the distribution panel. This allows the computer to communicate with up to 56 external devices (terminals and/or modems).

The distribution panel can distribute up to eight transmit (TXDAT0 through TXDAT7) signals and eight receive (RXDAT0 through RXDAT7) signals to terminals or modems. The terminals or modems connected to the same distribution panel must have the same protocol.

1.8.3 Eight-Line Asynchronous Communications Multiplexer

Figure 1-3 is a simplified Eight-Line Asynchronous Communications Multiplexer block diagram that shows the functional blocks of the multiplexer, the major control signals, and the data and address lines.

The functional blocks are as follows:

1. MPC interface (interfaces with the MP Bus)
2. I/O address decode
3. Interrupt logic
4. Memory
5. MPC EOI controller
6. DMA controller
7. Microprocessing unit (MPU)
8. Port (0 through 7)

The entire operation of the multiplexer is under the control of the microprocessor unit (MPU). The MPU controls the information transfers between the MPC interface and the MPU, between the MPC interface and memory, between the MPU and memory, between the MPU and the direct memory access (DMA) controller, and between the ports (0 through 7) and the MPU. The DMA controller can control the information transfers between the MPC interface and memory when it is programmed by the MPU to do so. The microprocessor has control of the bus (DMA single cycle mode) between DMA transfers.

1.8.3.1 MPC Interface

The mnemonic MPC refers to the nomenclature multipurpose controller. As used in this manual it is synonymous to the I/O Processor (IOP).

The MPC interface, which connects to the MP Bus, contains the logic necessary for information transfers between the MP Bus and the multiplexer. The interface also contains the logic necessary for controlling information transfers between the microprocessor unit (MPU) and the interface and between memory and the interface.

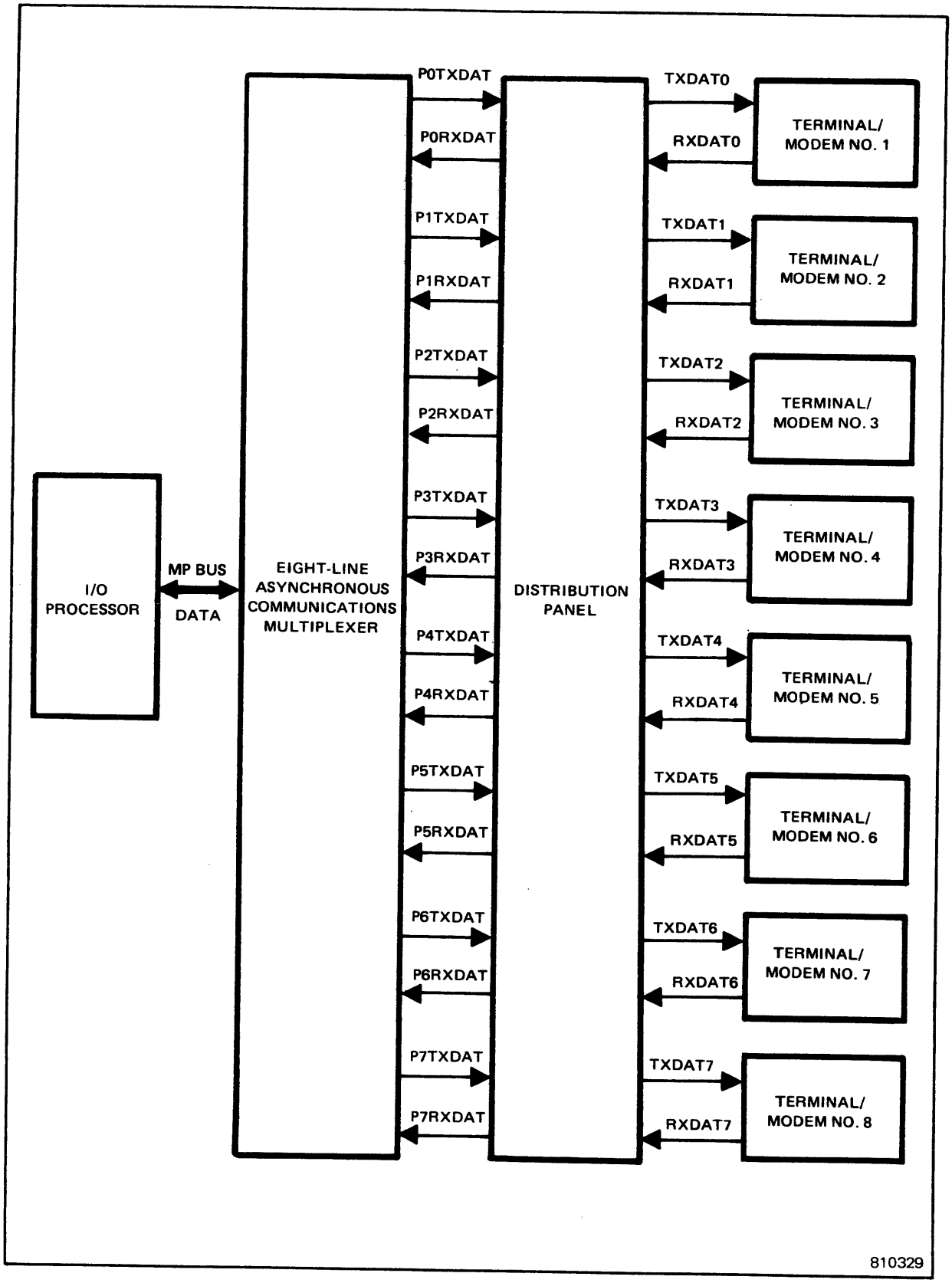


Figure 1-1. Communication System, Block Diagram

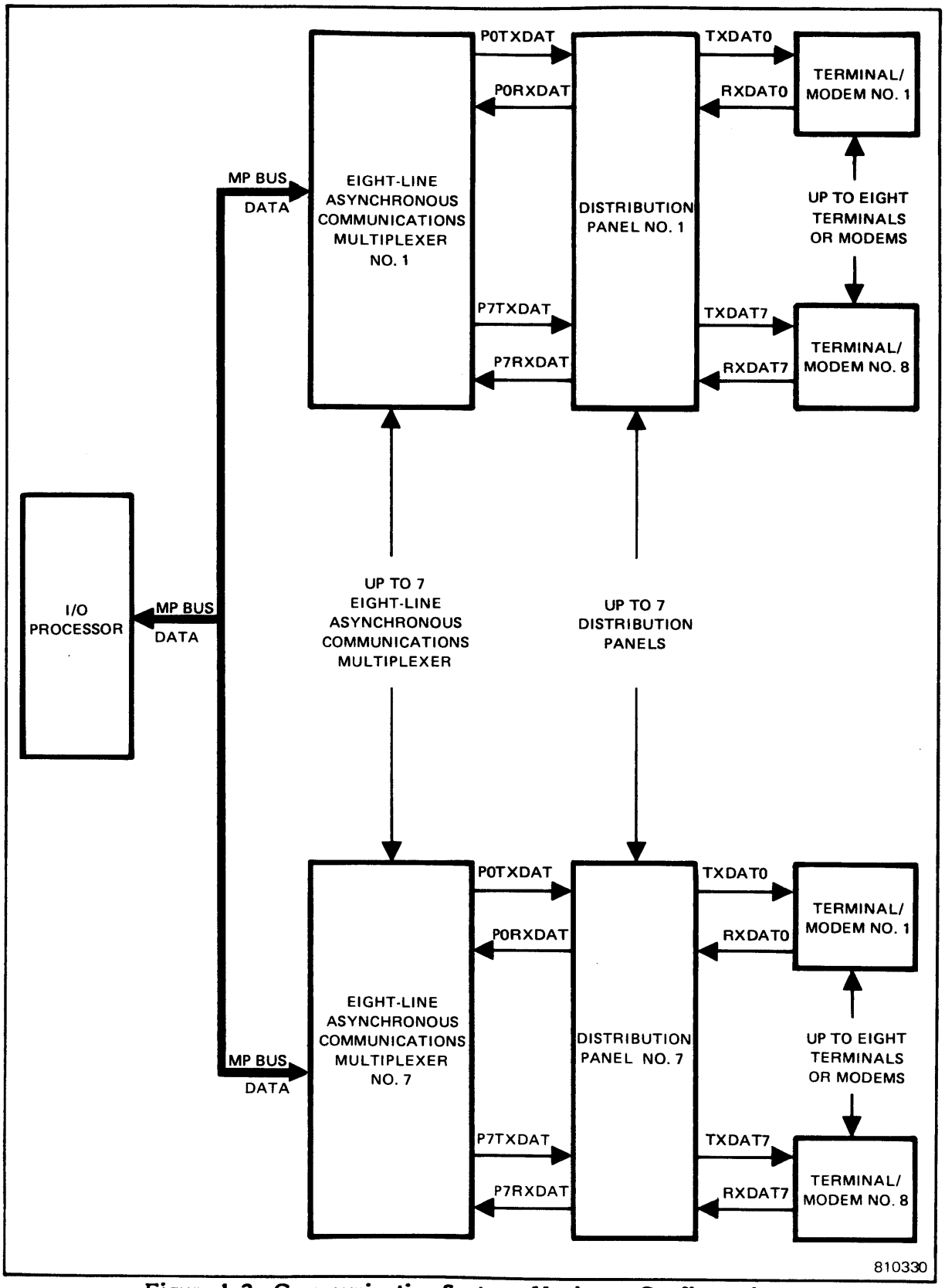


Figure 1-2. Communication System, Maximum Configuration, Block Diagram

Eight bidirectional data lines and one parity line effect information transfers between the MP Bus and the interface. The bidirectional handshake controls effect the information transfers.

The MPC interface checks parity (odd) during information transfers from the MP Bus to the interface and generates parity (odd) during information transfers to the MP Bus.

The controls contain the bidirectional three-wire handshake control lines which effect the information transfers from and to the MP Bus. Either the MP Bus or the MPC interface can initiate information transfers.

The MP Bus uses the handshake control lines to initiate and effect information transfers. The MPC interface initiates information transfers requesting service from the MP Bus. The bus then goes through a polling sequence to determine which multiplexer (of a possible 7), with the highest priority, requested the service. The information is then transferred to the MP Bus using the handshake control lines.

The clear line from the MP Bus is used to initialize the multiplexer. The clear signal is received at the MPC interface and distributed to other functional blocks.

The I/O read (IORD) and I/O write (IOWR) signals are generated by the MPU. The MPU uses the read and write signals to transfer information to and from the data bus (DBUS). The particular register in the MPC interface, which is to be read from or written to, is defined by the MPU via the address bus.

The MPC end or identity (EOI) line is bidirectional and comes from the MP Bus controller to indicate the end of a multibyte information transfer sequence. The EOI signal is generated by the MPC interface logic or received from the MP Bus controller.

When the MPC interface logic detects a parity error, an interrupt is generated.

1.8.3.2 I/O Address Decoder

The I/O address decoder receives address bits from the MPU via the MPU address bus. The address bits are decoded to generate a select signal to the MPC interface, the DMA controller, or one of the eight ports.

1.8.3.3 Interrupt Logic

The interrupt logic receives interrupts from the MPC interface and from the eight ports. The interrupt logic generates the INT signal to the MPU and places the interrupt vector on the MPU data bus. The MPU uses the vector to locate an interrupt handler which will process the particular interrupt.

1.8.3.4 Memory

The memory functional block is comprised of PROMs and RAMs.

Model 8512-2 has a memory consisting of three (2KX8) PROMs and two (2KX8) RAMs.

The PROMs contain the firmware program to control the operation of the multiplexer and are sometimes referred to as the control read only memory (CROM). The RAMs temporary information storage.

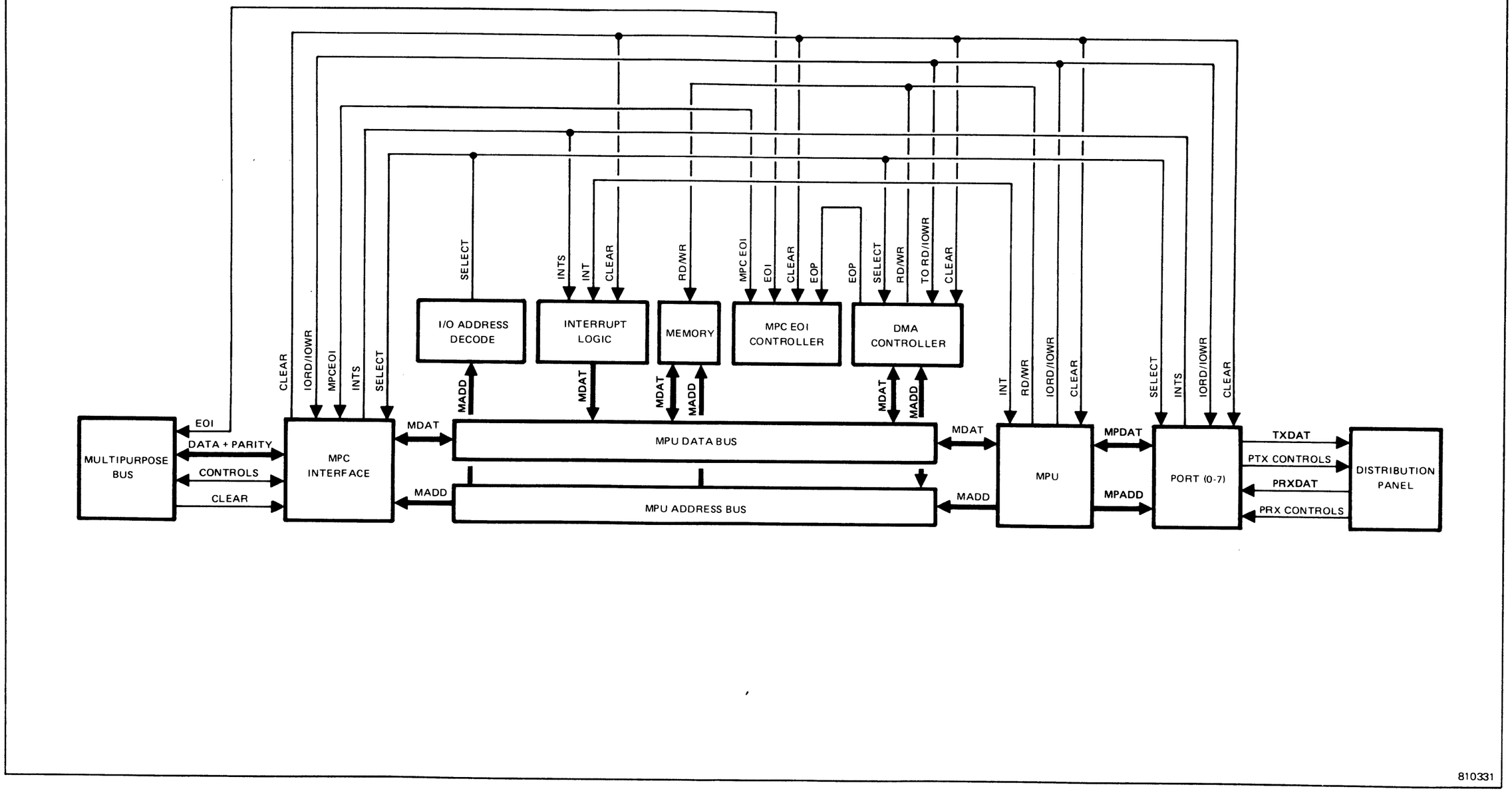


Figure 1-3. Eight-Line Asynchronous Communications Multiplexer, Block Diagram

The read/write (RD/WR) controls enable reading from and writing to a RAM location, as defined by the address on the MPU address bus. Only the read signal is used to access the PROMs.

1.8.3.5 MPC EOI Controller

The MPC interface logic enables and disables the receipt and generation of the EOI to the MP Bus. The MPCEOI signal is generated when the MPC interface receives an EOI signal. Both the EOI and the MPCEOI signals are generated when the end of process (EOP) signal is received from the DMA controller.

The end or identify signals (EOI and MPCEOI) indicate the end of a current multibyte information transfer.

1.8.3.6 DMA Controller

The DMA controller transfers information between memory and the MPC interface without MPU intervention; however, the MPU initiates the transfer operation.

The MPU selects the DMA controller using the I/O address decode logic. By using the IORD/IOWR signals, along with an address, the MPU can establish, change, or inspect internal DMA controller registers. Thus, the DMA controller is programmed to perform the information transfers between memory and the MPC interface.

The information transfers to and from memory are accomplished by the RD/WR signals. When the information transfer is complete the DMA controller issues an EOP signal.

1.8.3.7 Microprocessing Unit (MPU)

The MPU controls the operation of the Eight-Line Asynchronous Communications Multiplexer. The MPU accesses the PROM (where the firmware programs are stored) using the read (RD) signal. It also accesses the RAM (for temporary information storage) using both the RD and WR signals.

The MPU reads from or writes to the MPC interface, DMA controller, and ports using the IORD and IOWR signals.

The MPU data (MDAT) lines provide for an eight-bit bidirectional information transfer between the MPU, memory, and the board-mounted I/O device.

The MPU address (MADD) lines access memory locations, for memory read/write operations, and internal registers of the board-mounted I/O devices, for the I/O read/write operations.

The MPU also reads information from or writes information to the internal registers of the ports using the MPU port data (MPDAT) lines and the IORD/IOWR signals. The individual registers are selected by the MPU port address (MPADD) lines.

The interrupt (INT) input informs the MPU that an interrupt has occurred and the interrupt vector address is on the MDAT lines. The MPU uses the vector address to locate the interrupt handler, in memory, which will process the particular interrupt that occurred.

1.8.3.8 Port (0 through 7)

The port (0 through 7) functional block represents eight input/output ports (0 through 7). Each port transmits serial data (PTXDAT) and receives serial data (PRXDAT) from an external device (terminal or modem) through the distribution panel, using transmit and receive control signals.

The port performs a serial-to-parallel conversion on data received from the external I/O device and a parallel-to-serial conversion on data transmitted to the external I/O device.

Information transfers between the MPU and a port are accomplished using the MPDAT lines and the IORD and IOWR signals. The register to be read from or written to is selected by the MPADD lines.

The port informs the MPU that it has information to transfer by issuing an INT for that port. The MPU processes the INT, selects the port, and accepts the information on the MPDAT.

CHAPTER 2

OPERATION

2.1 Power-on/Power-off

The Eight-Line Asynchronous Communications Multiplexer receives power from the computer backplane. When power is applied to the computer system each multiplexer receives an interface clear (IFC) signal from the I/O processor (IOP). The interface clear causes the elements of the multiplexer to be initialized and the microprocessor unit (MPU) to start the program execution from location zero.

2.2 Initialization

The Eight-Line Asynchronous Communications Multiplexer is initialized during the computer system poweron sequence or when a system RESET is issued from the control panel. During initialization each multiplexer receives an IFC signal from the IOP. The IFC causes the element of the multiplexer to be initialized and the MPU to start the program execution from location zero.

2.3 Multiplexer Address

The multiplexer address (and priority) is established by jumpers. The lowest multiplexer address (address 0) has the highest priority during the polling sequence. The multiplexer address is read by the MPU after the receipt of an IFC.

Sheet 7, of the Eight-Line Asynchronous Communications Multiplexer logic diagram, contains a table showing the multiplexer address jumper configurations.

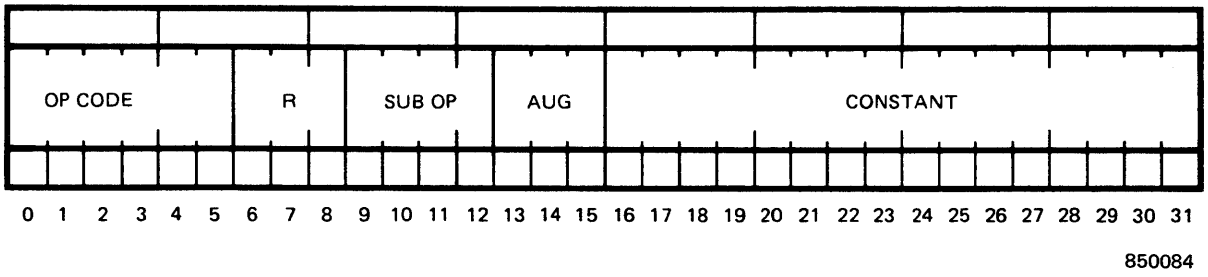
CHAPTER 3

PROGRAMMING

3.1 Instructions

3.2 Basic Instruction Format

All extended I/O instructions follow one basic format. Specific details about each instruction can be found in the IOP Reference Manual, publication number 301-000170.



Op code - All ones.

R field - The constant (bits 16 through 31) is added arithmetically to the contents of the general purpose register specified by "R". A zero in this field results in no change to the constant.

Sub op - Type of operation (defined below).

Aug code - All ones.

Constant - Eight-bit logical channel (bits 16 through 23) and eight-bit subchannel address (bits 24 through 31).

3.2.1 Sub Op Codes

0010	SIO	(Start I/O)
0011	TIO	(Test I/O)
0100	STPIO	(Stop I/O)
0101	RSCHNL	(Reset Channel)
0110	HIO	(Halt I/O)
1000	RSCTL	(Reset Controller)
1100	ECI	(Enable Channel Interrupt)
1101	DCI	(Disable Channel Interrupt)
1110	ACI	(Activate Channel Interrupt)
1111	DACI	(Deactivate Channel Interrupt)

3.2.2 Condition Codes

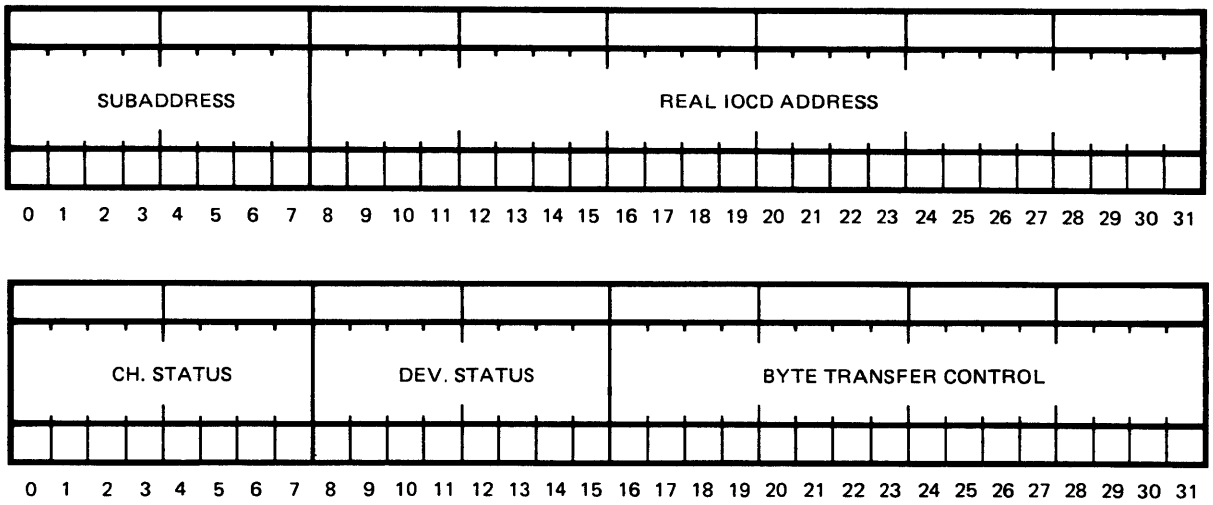
Condition codes (CC) indicate the successful or unsuccessful initiation of an I/O instruction as follows:

CC1 - CC4

0	0	0	0	Request accepted will echo status.
0	0	0	1	Channel busy.
0	0	1	0	Channel inoperable or undefined.
0	0	1	1	Subchannel busy.
0	1	0	0	Status stored.
1	0	0	0	Request accepted and queued, no echo status.

3.3 Status Doublewords

A status doubleword is returned to main memory after the completion or termination of a command operation. An eight-bit controller/device status byte is transferred from the controller to the channel and merged with an eight-bit channel status byte to form part of the second word. Specific details about each status bit as well as the rest of the status doubleword can be found in the Input/Output Processor (IOP) Reference Manual, publication number 301-000170.



850083

3.3.1 Device Status Bits

Bit		
	8	Busy
	9	Status modifier
	10	Controller end
	11	Attention
	12	Channel end
	13	Device end
	14	Unit check
	15	Unit exception

The definition of each device status bit is as follows:

1. Busy (XX8C XXXX) - This condition indicates that the device or controller cannot execute the command because it is busy executing a previously initiated operation. If a read command is received while a write with input subchannel monitoring condition exists, or if a write command is received while a read echoplex command exists, a busy condition occurs causing an interrupt to be generated.

2. Status modifier - This bit is not used by the Eight-Line Asynchronous Communications Multiplexer.

3. Controller end - This bit is not used by the Eight-Line Asynchronous Communications Multiplexer.

4. Attention (XX1C XXXX) - This condition indicates that the device has detected a device-dependent asynchronous operation. A modem ring wake-up, character ring wake-up or Delta DSR (Data Set Ready) (with no read in progress) signal causes an attention status interrupt.

5. Channel end - This condition is caused by the completion of the I/O operation involving data transfer or control information between the controller/device and the channel. This condition indicates that the subchannel has become available for use in another operation. This bit is set with one or more of the other bits.

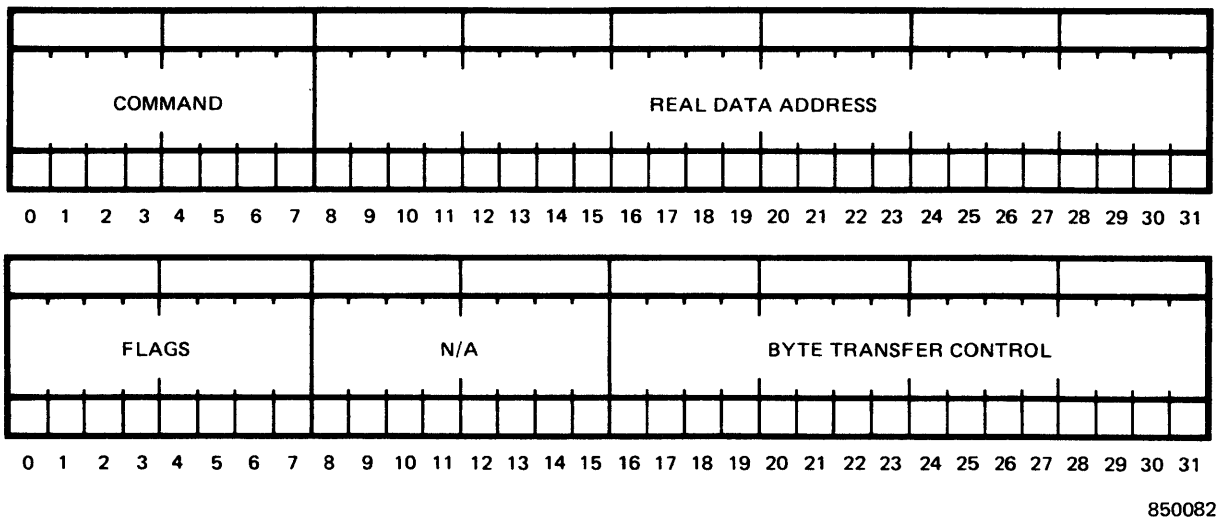
6. Device end (XX0C XXXX) - This condition is caused by the completion of an I/O operation at the device. The device end (DE) condition indicates that the device has become available for another operation. The channel end (CE) bit will always be set when this bit is set.

7. Unit check (XX0E XXXX) - This condition indicates that the controller or device has detected an unusual condition. Details about the condition can be found in the sense status (when requested). The DE and CE bits will always be set when this bit is set.

8. Unit exception (XX0D XXXX) - This bit is only set by a break interrupt signal received while a read operation is in progress. The CE and DE bits are set when this bit is set.

3.4 Input/Output Command Doublewords

All control, sense, and I/O commands are placed into doublewords and become part of an I/O Command List (IOCL) in main memory. Chaining is used to link these commands together. After receiving a start I/O (SIO) command, the Input/Output Processor (IOP) fetches these commands and passes them to the proper controller. The transfer count, address, and flags remain with the IOP while the controller receives the command bits from the I/O command doubleword (IOCD) and the subchannel address from the SIO instruction constant.



Only command bits will be covered in this technical manual while information regarding the rest of the IOCD can be found in the IOP Technical Manual, publication number 303-000170.

3.4.1 Write Commands

The write commands (bits 0 through 7) are as follows:

0 0 0 0	0 0 0 1	Write
0 0 0 0	0 1 0 1	Write with input subchannel monitoring
0 0 0 0	1 1 0 1	Write with hardware flow control only

NOTES

1. All write commands must use address hexadecimal 8 through F.
2. DSR must be true when a write command is issued or the command will be rejected with a unit check. (Exception: When a write is issued to a port that is full duplex with modem ring enabled, the DSR check is bypassed.)

3.4.1.1 Write

This command causes a block of data (1 to 65,535 bytes) to be transferred from main memory to one of eight asynchronous subchannels. Read commands are accepted on the corresponding subchannel. A CE/DE status byte is returned at the completion of the command.

3.4.1.2 Write with Input Subchannel Monitoring

This command causes a block of data to be transferred to a device while simultaneously monitoring the corresponding input subchannel for the receipt of a DC3 (X-OFF = 13 hexadecimal), DC1 (X-ON = 11 hexadecimal), or ETX (End of Text = 03 hexadecimal) character.

If a DC3 command is received, the output function is suspended until a DC1 command is received. An ETX command causes an orderly termination of the output function to occur with CE/DE status returned to the Eight-Line Asynchronous Communications Multiplexer.

This command also monitors the clear to send (CTS) line and, should CTS go negative (false), data transmission will be temporarily suspended until CTS again goes positive (true).

This command is only processed in the full-duplex mode with no read operation in progress. A read operation attempted on the input subchannel returns a busy status.

3.4.1.3 Write With Hardware Flow Control Only

This command is functionally equivalent to the write with input subchannel monitor mode command with an important exception. The command makes use of the hardware interface lines only to implement flow control. This allows full duplex operation on one port with flow control, as this command does not use the corresponding subchannel for Xon/Xoff transmission. This command monitors CTS as in the write with input subchannel monitor mode command.

3.4.2 Read Commands

The read commands (bits 0 through 7) are as follows:

	0	A	S	P	0	0	1	0	Read
	0	A	S	P	0	1	1	0	Read echoplex
*	0	A	S	P	1	0	1	0	Read with flow control
	1	A	S	P	1	1	1	0	Read with hardware flow control only

*If bit 0 is set in the command byte, DTR is used for flow control.
If bit 0 is reset in the command byte, RTS is used for flow control.

NOTES

1. A=ASCII control character detect (7-bit mode only)
   S=Special character detect
   P=Purge input buffer

2. All read commands must use addresses 0 through 7.

3. DSR must be true when a read command is issued or the command will be rejected with a unit check. (Exception: when a read is issued to a port that is full duplex with modem ring enable, the DSR check is bypassed).

3.4.2.1 Read

This command causes a block of data to be transferred to main memory for one of the eight asynchronous subchannels. If the purge modifier bit is set, any input data held in the buffer by the subchannel will be cleared. Otherwise, this 'type ahead' data is

processed before any new incoming data. The type ahead feature allows input data to be saved before a read command reaches the controller and after the termination of each read operation.

If the special character detect bit is set, the input operation is terminated whenever a previously defined eight-bit character is detected. At this point the input operation is terminated with the special character, written into memory, as the last character.

If the ASCII control character detect bit is set, the input operation is terminated whenever a control character (hexadecimal 00 through 1F) or a delete (hexadecimal 7F) instruction is detected. The last character written into memory is the ASCII control character. The 'type ahead' input buffer is used to hold input data after a read command is completed. This feature allows an uninterrupted flow of input data while waiting for a new read command to reach the controller. The ASCII control character detect applies to 7-bit mode of operation only.

Note that write commands are allowed on the corresponding subchannel with this read instruction.

3.4.2.2 Read Echoplex

This command allows a block of data to be read while all printable ASCII characters are outputted to the corresponding subchannel. This command will only be processed in the full-duplex mode with no write command in progress. After this read command is accepted, no write commands will be accepted on the corresponding subchannel.

A write attempted on the Output Subchannel will return busy status. Purge, ASCII Control Character Detect, and special character detect are selectable options.

3.4.2.3 Read With Flow Control

The read with flow control is the complement to the write with input subchannel monitor mode command. The issuance of this command causes the write subchannel to be set busy. The eight-line then monitors the input buffer. If the input buffer reaches the 70% full point (38 bytes remaining), an Xoff character (Hex 13) is sent out on the write subchannel. At the same time either DTR (if bit 0 is set in the command byte) or RTS (if bit 0 is reset) is reset. When the input buffer is nearly empty (8 bytes remaining) an Xon character (Hex 11) is sent out and the DTR or RTS line is set.

3.4.2.4 Read With Hardware Flow Control Only

This command is functionally equivalent to the read with flow control command with an important exception. The command makes use of the hardware interface lines only to implement flow control. This allows full duplex operation on one port with flow control, as this command does not use the corresponding subchannel for Xon/Xoff transmission. This command toggles DTR in the same manner as the read with flow control command.

3.4.3 Control Commands

The control commands (bits 0 through 7) are as follows:

1	1	1	1	1	1	1	1	Define ACE (UART) parameters (three bytes follow)
0	0	0	0	0	0	1	1	No operation (NOP)
0	0	0	0	1	0	1	1	Define special character (one byte follows)
0	0	0	1	0	0	1	1	Reset data terminal ready
0	0	0	1	0	1	1	1	Set data terminal ready
0	0	0	1	1	0	1	1	Reset request to send
0	0	0	1	1	1	1	1	Set request to send
0	0	1	1	0	1	1	1	Set break
0	0	0	0	0	1	0	0	Sense

NOTE

All control commands use either group of addresses (hexadecimal 0 through 7 or 8 through F).

3.4.3.1 Define ACE Parameters

This control command precedes a three-byte transfer which defines ACE parameters, wake-up ring, and half-/full-duplex operation as described by byte 1, byte 2 and byte 3.

0	.	.	.	.	.	.	7	Byte 1
-	-	-	-	-	-	0	0	Five-bit character length
-	-	-	-	-	-	0	1	Six-bit character length
-	-	-	-	-	-	1	0	Seven-bit character length
-	-	-	-	-	-	1	1	Eight-bit character length
-	-	-	-	-	0		-	One stop bit
-	-	-	-	-	1	-	-	Two stop bits or one and one half stop bits for five-character length
-	-	-	-	0	-	-	-	Parity disabled
-	-	-	-	1	-	-	-	Parity enabled
-	-	-	0	-	-	-	-	Odd parity
-	-	-	1	-	-	-	-	Even parity
-	-	0	-	-	-	-	-	Normal parity as defined
-	-	1	-	-	-	-	-	Force parity to 1 if odd, 0 if even
-	0	-	-	-	-	-	-	Disable modem ring
-	1	-	-	-	-	-	-	Enable modem ring
0	-	-	-	-	-	-	-	Full-duplex operation
1	-	-	-	-	-	-	-	Half-duplex operation

0 7	Byte 2
- - - - 0 0 0 0	50 baud
- - - - 0 0 0 1	75 baud
- - - - 0 0 1 0	110 baud
- - - - 0 0 1 1	134.5 baud
- - - - 0 1 0 0	150 baud
- - - - 0 1 0 1	300 baud
- - - - 0 1 1 0	600 baud
- - - - 0 1 1 1	1200 baud
- - - - 1 0 0 0	1800 baud
- - - - 1 0 0 1	2000 baud
- - - - 1 0 1 0	2400 baud
- - - - 1 0 1 1	3600 baud
- - - - 1 1 0 0	4800 baud
- - - - 1 1 0 1	7200 baud
- - - - 1 1 1 0	9600 baud
- - - - 1 1 1 1	19200 baud
- - - 0 - - - -	Reset diagnostic loop
- - - 1 - - - -	Set diagnostic loop
- - 0 - - - - -	Allow wake-up character detection
- - 1 - - - - -	Inhibit wake-up character detection

0 7	Byte 3
C C C C C C C C	Eight-bit wake-up character

3.4.3.2 NOP (No Operation)

The NOP command only causes a CE/DE status interrupt. No activity occurs at the controller.

3.4.3.3 Define Special Character

This control command causes a one-byte transfer to the controller which contains an eight-bit character used for input termination.

3.4.3.4 Reset Data Terminal Ready (DTR)

This control command resets the DTR signal for the addressed port. The DTR signal is automatically reset during a controller clear or power reset operation.

3.4.3.5 Set Data Terminal Ready (DTR)

This control command sets the DTR signal for the addressed port.

3.4.3.6 Reset Request to Send (RTS)

This control command resets the RTS signal for the addressed port. The RTS signal is automatically reset during device or controller clear or a power reset operation.

If the port is defined half-duplex, this signal is automatically reset at the completion of a write command.

3.4.3.7 Automatic Handling of RTS and DTR

Upon initialization of a line (define ACE parameters) the control lines request to send and data terminal ready are set or reset depending on the line parameters. If ring is enabled, the line is assumed to be remote (i.e., modem) and both signals are reset. If ring is not enabled, the line is assumed to be local (i.e., local TTY), DTR is set, and RTS is set if the line is full duplex. If the line is half duplex, RTS is reset.

3.4.3.8 Set Request to Send (RTS)

This control command sets the RTS signal for the addressed port.

If the port is defined half-duplex, this signal is automatically set at the start of each write command.

3.4.3.9 Reset Break

This control command disables the break feature, releasing the serial output line from the clamped logic 0 state.

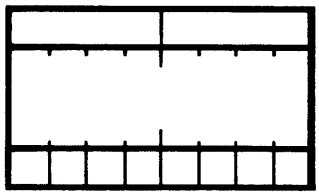
3.4.3.10 Set Break

This control command sets the serial data output line to the spacing logic 0 state until a reset break command is received.

3.4.3.11 Sense Status

This command causes eight bytes of sense information to be transferred to main memory for the addressed port. The first six bits of byte 0 are standard for all controllers.

Sense Byte 0: Device status



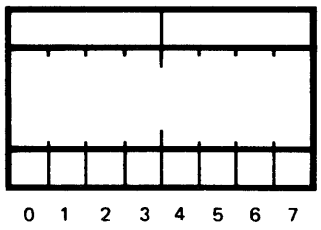
850081-1

Bit 0 Command reject (program violation)
1 Intervention required (not used)
2 Bus out check (IOP parity error)
3 Equipment check (device error)
4 Data check
5 Overrun (not used)
6 Zero
7 Zero

The definition of each device status bit is as follows:

1. Command reject (bit 0) - This bit indicates that an improper command has been received and ignored. If a port is defined as half-duplex, a command reject command occurs when the write with input subchannel monitoring or a read echoplex command is executed. Note that undefined commands also result in a program violation indication. Unit check status is returned when a program violation is encountered. If a read command is received during a write operation or a write command during a read operation and the controller is operating in the half-duplex mode, a busy condition, not a program violation, is generated.
2. Intervention required (bit 1) - This bit indicates that the device requires operator intervention (not used with the eight-line asynchronous communications Multiplexer.).
3. Bus out check (bit 2) - This bit indicates that a parity error has occurred on the I/O channel bus thus generating unit check status.
4. Equipment check (bit 3) - This bit indicates that an I/O device malfunction has occurred during the last operation. Bit 3 is set when the data set ready or carrier detect lines go low (for 150 ms) while the data terminal ready line is high. An equipment check indication also occurs if the clear to send line goes low (in the full-duplex mode) while the RTS line is high. In either situation, the I/O operation is halted, unit check status posted, and a unit check interrupt is issued.
5. Data check (bit 4) - This bit indicates that data is in error or lost. An overrun, parity, or framing error must have occurred during input at the ACE for this bit to be set. The I/O operation is halted, unit check status posted, and unit check interrupt issued.
6. Overrun (bit 5) - This bit is not used with the Eight-Line Asynchronous Communications Multiplexer.
7. Bits 6 and 7 are always zero.

Sense Byte 1: Line Status and Error Conditions



850081-2

Bit 0 ASCII control character detected interrupt
1 Special chararacter detected interrupt
2 ETX interrupt
3 Break interrupt
4 ACE framing error interrupt
5 ACE parity error interrupt
6 ACE overrun error interrupt
7 Ring character interrupt

The definition of each line status and error condition bit is as follows:

1. ASCII control character detected (bit 0) - This bit indicates that an ASCII control character (hexadecimal 00 through 1F or 7F) has been received during the last read operation. The ASCII control character detected bit must be set with the read command to cause this status condition. All buffered data, including the ASCII character, is inputted to main memory. CE/DE status is posted and a CE/DE status interrupt issued.

2. Special character detected (bit 1) - This bit indicates that the previously defined ASCII character has been received during the last read operation. The special character detect bit must have been set with the read command in order to cause this status condition. All buffered data including the special character will be inputted to main memory. CE/DE status is posted and CE/DE status interrupt is issued.

3. ETX detected (bit 2) - This bit indicates that an ETX control character has been received during the last input operation, terminating the port's output. The only command which can cause this status is the write with input subchannel monitoring to a full-duplex port. Output data is immediately terminated, CE/DE status is posted, and CE/DE status interrupt is issued.

4. Break interrupt (bit 3) - This bit indicates that the input data line has been held in the spacing (logic 0) state for longer than a full character time period. If a read operation was in progress, unit exception status is posted and a unit exception status interrupt issued. Framing error (bit 4) is also set. If no I/O operation was in progress, attention status is posted. Only one break interrupt is accepted between read operations.

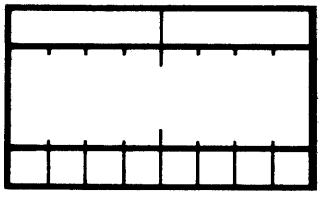
5. ACE framing error (bit 4) - This bit indicates that the incoming character did not have valid stop bit(s). If this bit is set without the break interrupt bit (bit 3), unit check status is posted and a unit check status interrupt issued.

6. ACE parity error interrupt (bit 5) - This bit indicates that an incoming character did not have the correct parity at the ACE. The data check bit in sense byte 0 is also set. Unit check status is posted and a unit check status interrupt issued.

7. Overrun error (bit 6) - This bit indicates that the internal 128 byte buffer has been overrun or that data has been lost at the ACE due to the controller's failure to read an incoming byte before the reception of the next byte. The data check bit in sense byte 0 is also set. Unit check status is posted and a unit check status interrupt issued.

8. Ring character interrupt (bit 7) - This bit indicates that a ring character has been received when a read operation was not in progress. Attention status is posted and an attention status interrupt issued.

Sense Byte 2: Modem Status



Bit
0 Received line signal detect status
1 Ring indicator line status
2 Data set ready line status
3 Clear to send line status
4 Delta receive line signal detect failure interrupt
5 Modem ring interrupt
6 Delta data set ready interrupt
7 Delta clear to send failure interrupt

The definition of each modem status bit is as follows:

1. Received line signal detect (bit 0) - This bit indicates the logic condition of the RLSD input signal. When this bit is set, the signal is high.

2. Ring indicator (bit 1) - This bit indicates the logic condition of the RI input signal. When this bit is set, the signal is high.

3. Data set ready (bit 2) - This bit indicates the logic condition of the DSR input signal. When this bit is set, the signal is high.

4. Clear to send (bit 3) - This bit indicates the logic condition of the CTS input signal. When this bit is set, the signal is high.

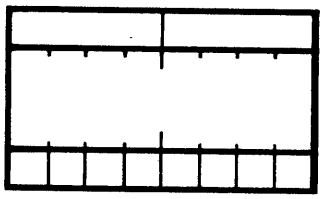
5. Delta receive line signal detect failure interrupt (bit 4) - This bit indicates that the received line signal detect failure (RLSD) input signal has gone low for at least 150 ms, while a read operation was in progress. The read operation is terminated with unit check status posted and a unit check status interrupt is issued.

6. Modem ring interrupt (bit 5) - This bit indicates that the ring indicator (RI) signal has changed (trailing edge only). If the modem ring enable bit is set during the initialization of byte 1, the controller will generate attention status. The DTR line must then be set, by software, to "answer the phone." This task is accomplished via the set DTR control command.

7. Delta data set ready interrupt (bit 6) - This bit indicates that a transition occurred on the data set ready (DSR) input signal. If a read operation was in progress it is terminated, unit check status is posted, and a unit check status interrupt issued. If a read operation was not in progress and the DTR line is high, attention status is posted and an attention status interrupt issued.

8. Delta clear to send failure interrupt (bit 7) - This bit indicates that the clear to send (CTS) input signal shifted from high to low. If a read operation was in progress (full-duplex mode) or a write operation is in progress (half-duplex mode), the read operation is terminated, unit check status is posted, and a unit check status interrupt issued.

Sense Byte 3: Modem Control/Operation Mode



850081-4

Bit
0 Half-duplex operation set
1 Modem ring enabled (1)
2 ACE parameters defined
3 Diagnostic mode set
4 Auxiliary output level 2
5 Auxiliary output level 1
6 Request to send (RTS)
7 Data terminal ready (DTR)

The definition of each modem control/operation mode bit is as follows:

1. Half-duplex operation set (bit 0) - This bit indicates that the port is designated for half-duplex operation. If this bit is not set, the port is designated for full-duplex operation.

2. Modem ring enabled (1) (bit 1) - If this bit is set the subchannel responds to a modem line ring interrupt, thereby causing an attention status interrupt to be generated. The DTR line is not set by the Eight-Line Asynchronous Communications Multiplexer. Note that it is the responsibility of the software to set or reset the DTR signal using the control commands provided.

3. ACE parameters defined (bit 2) - This bit indicates that the ACE has been initialized with a control command since it was last cleared.

4. Diagnostic mode set (bit 3) - This bit indicates that the diagnostic loop-back feature of the ACE is enabled and all data outputted must be read back in with a read command. The read command should be issued before the write command.

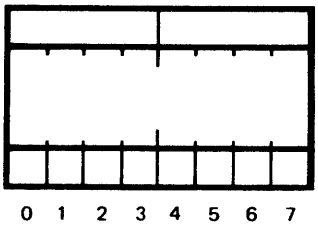
5. Auxiliary output level 2 (bit 4) - This bit indicates the logic condition of auxiliary output level 2.

6. Auxiliary output level 1 (bit 5) - This bit indicates the logic condition of auxiliary output level 1.

7. Request to send (bit 6) - This bit indicates the logic condition of the request to send (RTS) output signal. When this bit is set, the signal is high.

8. Data terminal ready (bit 7) - This bit indicates the logic condition of the data terminal ready (DTR) output signal. When this bit is set, the signal is high.

Sense Byte 4: ACE Parameters



850081-5

Bit 0 Divisor latch enable
- 0 = disabled
- 1 = enabled

1 Break set
- 0 = reset
- 1 = set

2 Forced parity
- 0 = odd parity
- 1 = even parity

3 Parity
- 0 = odd parity
- 1 = even parity

4 Parity enable
- 0 = disabled
- 1 = enabled

Bit 5 Stop bit
- 0 = 1 stop bit
- 1 = 1.5 or 2 stop bits

6-7 Character length
- 00 = 5
- 01 = 6
- 10 = 7
- 11 = 8

The definition of each ACE parameter bit is as follows:

1. Divisor latch is enabled (bit 0) - When this bit is set the ACE's divisor latch (used for loading the baud rate divisor) is accessible. Normal I/O operation cannot take place when this latch is enabled.

2. Break is set (bit 1) - When this bit is set the data output line is being held in the spacing logic 0 state.

3. Forced parity (bit 2) - If this bit is set the parity bit will always be a one for odd selected parity, or a zero for even selected parity.

4. Even and odd parity (bit 3) - When this bit is set, even parity has been selected. The sum total of all ones in the data character (with parity bit) is generated and checked for an even count. If this bit is not set, parity is generated and checked odd.

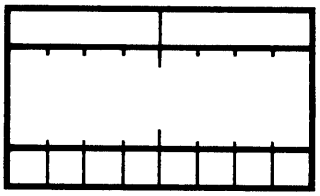
5. Parity enabled (bit 4) - When this bit is set, the proper parity is generated during an output operation, and checked during an input operation by the ACE.

6. Stop bits (bit 5) - If this bit is set, two stop bits are transmitted, with each character, and expected to be received with each character. If the character length has been defined as five bits, then only one and one-half stop bits is transmitted and received. If this bit is not set, one stop bit is transmitted and received.

7. Character length (bit 6 and 7) - These two bits indicate the character length for transmitted and received data.

   If bits 0,1 = 00: character length is five bits.
   If bits 0,1 = 01: character length is six bits.
   If bits 0,1 = 10: character length is seven bits.
   If bits 0,1 = 11: character length is eight bits.

Sense Byte 5: Baud Rate



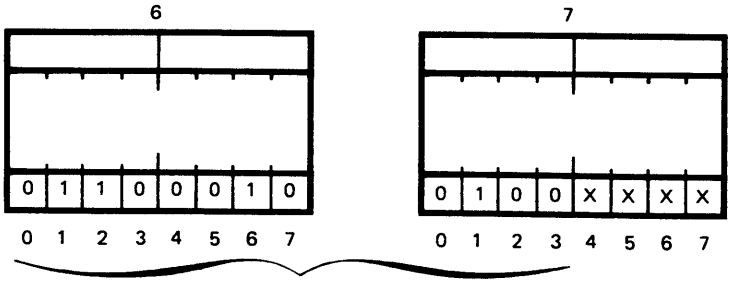
850081-6

Bit 0, 1	Zero.
Bit 2	Ring character recognition 0 = enable 1 = inhibit
Bit 3	Zero.
Bits 4, 5, 6, 7	Define baud rate as in baud rate select command.

The definition of the baud rate bits is as follows:

Bits 4, 5, 6, 7 represent the baud rate selected for the port. If the port has not been initialized, these bits will be 0 (this does not mean that the baud rate of 50 has been selected).

Sense Bytes 6 and 7: Firmware Identification, Revision Level



850081-7

3.5 Microprogramming

The Eight-Line Asynchronous Communications Multiplexer is microprogrammed in Z80 assembly code.

Each peripheral component on the multiplexer is accessed by the microprocessor via input and output instructions. Addressing the peripheral components uses the lower eight bits of the microprocessor address. In some cases, data bits are also used.

Component addressing is discussed in the following paragraphs.

3.5.1 Multimode DMA Controller

A programmable DMA controller is used to increase performance of the MPC interface. This controller contains four DMA channels: Channels 0 and 1 are available to perform high-speed MPU memory to memory transfers, channel 2 is assigned to the MPC interface, and channel 3 is not used.

The DMA controller is accessed as follows:

Address	LRD	LWR	Operation
10H*	1	0	Write bits A0 through A7 in the base and current address registers for channel 0.
			Write bits A8 through A15 in the base and current address registers for channel 0.
10H*	0	1	Read bits A0 through A7 from the current address register for channel 0.
			Read bits A8 through A15 from the current address register for channel 0.
11H*	1	0	Write bits W0 through W7 in the base and current word count registers for channel 0.
			Write bits W8 through W15 in the base and current word count registers for channel 0.
11H*	0	1	Read bits W0 through W7 in the current word count register for channel 0.
			Read bits W8 through W15 in the current word count register for channel 0.
12H*	1	0	Same as 10H but channel 1
12H*	0	1	Same as 10H but channel 1
13H*	1	0	Same as 11H but channel 1
13H*	0	1	Same as 11H but channel 1
14H*	1	0	Same as 10H but channel 2
14H*	0	1	Same as 10H but channel 2

Address	LRD	LWR	Operation
15H*	1	0	Same as 11H but channel 2
15H*	0	1	Same as 11H but channel 2
18H	1	0	Write command register
18H	0	1	Read status register
19H	1	0	Write request register
1AH	1	0	Set/reset mask register
1BH	1	0	Write mode register
1CH	1	0	Clear internal flip flop
1DH	0	1	Read temporary register
1DH	1	0	Master clear
1FM	1	0	Write mask register

* The DMA controller uses an internal flip-flop to determine whether the least-significant or most-significant byte is to be used.

3.5.2 Asynchronous Communication Element (ACE)

The eight ACEs on the Eight-Line Asynchronous Communications Multiplexer each require eight I/O addresses. The base address for each port is as follows:

20H port 0
28H port 1
30H port 2
38H port 3
40H port 4
48H port 5
50H port 6
58H port 7

The following description is for port 0. The other ports are programmed using the base address specified for that port instead of port 0's base address.

Address	LRD	LWR	DLAB	Operation
20H	0	1	0	Read receive buffer register
20H	1	0	0	Write transmitter holding register
20H	0	1	1	Read divisor latch (LS)
20H	1	0	1	Write divisor latch (LS)
21H	0	1	0	Read interrupt enable register
21H	1	0	0	Write interrupt enable register
21H	0	1	1	Read divisor latch (MS)
21H	1	0	1	Write divisor latch (MS)
22H	0	1	X	Read interrupt ID register
23H	0	1	X	Read line control register

Address	LRD	LWR	DLAB	Operation
23H	1	0	X	Write line control register
24H	0	1	X	Read modem control register
24H	1	0	X	Write modem control register
25H	0	1	X	Read line status register
25H	1	0	X	Write line status register
26H	0	1	X	Read modem status register
26H	1	0	X	Write modem status register

DLAB = Divisor latch access bit contained in line control register.

3.5.3 I/O Processor

Several control functions of the IOP interface are controlled external to the MP Bus. These functions are handled by the MPU as follows:

Address	LRD	LWR	Data Byte	Description
60H	0	1	XXXXAAAA	Read controller base address
60H	1	0	XXXXXPTE	Write IOP control

AAAA = four-bit base address
P = Reset MPC parity error interrupt
T = Reset MPC terminate interrupt
E = Enable/disable EOI logic for data transfer

The internal registers of the 8291 GPIB chip, used for the MPC interface, are addressed as follows:

Address	LRD	LWR	Operation
70H	0	1	Read data in register
70H	1	0	Write data out register
71H	0	1	Read interrupt status 1 register
71H	1	0	Write interrupt mask 1 register
72H	0	1	Read interrupt status 2 register
72H	1	0	Write interrupt mask 2 register
73H	0	1	Read serial poll status register
73H	1	0	Write serial poll mode register
74H	0	1	Read address status register
74H	1	0	Write address mode register
75H	0	1	Read command pass through register
75H	1	0	Write auxillary mode register
76H	0	1	Read address 0 register
76H	1	0	Write address 0/1 register
77H	0	1	Read address 1 register
77H	1	0	Write FOS register

CHAPTER 4

THEORY OF OPERATION

4.1 MP Bus Interface Information Transfers

The interface between the multipurpose bus (MP Bus) and the Eight-Line Asynchronous Communications Multiplexer is bidirectional. Information transfers can be initiated by the MP Bus or the Eight-Line Asynchronous Communications Multiplexer. There are two protocols on the MP Bus used for information transfers: the service request (SRQ) protocol, for transfers initiated by the Eight-Line Asynchronous Communications Multiplexer; and the MP Bus initiated protocol, for transfers initiated by the MP Bus.

4.1.1 Service Request (SRQ) Initiated Information Transfers

The multiplexer can transfer two types of information to the MP Bus using the service request protocol:

1. Data transfers
2. Status transfers

4.1.2 MP Bus Initiated Information Transfers

The MP Bus can initiate three types of information transfers to the multiplexer using the three-wire handshake protocol sequence:

1. Control signal transfers
2. Reset transfers
3. Command transfers

4.2 Service Request (SRQ) Protocol Sequence

The SRQ protocol informs the bus that the Eight-Line Asynchronous Communications Multiplexer is ready to transfer data for the port specified in the state information (the state information contains the address of the external device).

The service request protocol sequence is as follows:

1. The MPU sets the SRQ bit (bit 6) in the serial poll mode register of the talker/listener. The serial poll mode register is used to establish the state byte that the talker/listener sends to the MP Bus when it receives the serial poll enable (SPE) message. The setting of the SRQ bit causes the talker/listener to activate the SRQ signal to the MP Bus.

2. The MP Bus then initiates a parallel polling sequence to determine which 2, of a possible 16 controllers initiated the service request. The poll priority of the controllers is established by their address.

3. The MP Bus responds to the results of the parallel polling by performing a serial poll. The serial poll informs the IOP which one of two controllers, as determined in the parallel poll, requested service. If both had requested service the IOP will service the highest priority (lowest address) controller.

4. The following types of information transfers can take place, after the serial polling sequence, using the three-wire handshake protocol sequence:

   A. Data transfers
   B. Status transfers

4.2.1 Parallel Polling Sequence

The parallel polling sequence is initiated by the multipurpose bus (MP Bus) in response to a service request (SRQ) by an Eight-Line Asynchronous Communications Multiplexer. The parallel polling sequence identifies which 2, of a possible 16 controllers initiated the SRQ. The parallel polling sequence occurs when the multiplexer requires data to be transferred for the input/output command doubleword (IOCD) in progress.

The parallel polling sequence is as follows:

1. The MP Bus responds to the SRQ by activating (pulling the lines low) the attention (LATN) and the end or identity (EOI) signals. This action is recognized by all multiplexers as the start of the parallel poll.

2. The multiplexer responds by activating the data line associated with its priority (address). The priority of the multiplexers is the same as its address. See Table 4-1, Poll Priorities. (The poll priority and multiplexer address can be determined by the configuration of the address jumpers.) The multiplexer with the lowest address has the highest priority.

3. The IOP examines the MP Bus data lines for an active condition. The IOP responds to highest priority active data line by converting this information into two multiplexer addresses (an even and an odd address). These addresses are stored in the IOP for use during the serial polling sequence.

4. The ATN and the EOI signals then go inactive (high) to indicate the termination of the parallel polling sequence. The multiplexer must remove its priority from the data lines in 300 nanoseconds.

Figure 4-1, Parallel Polling Sequence Timing Diagram shows the relationship between the ATN, EOI, and data signals during a parallel polling sequence.

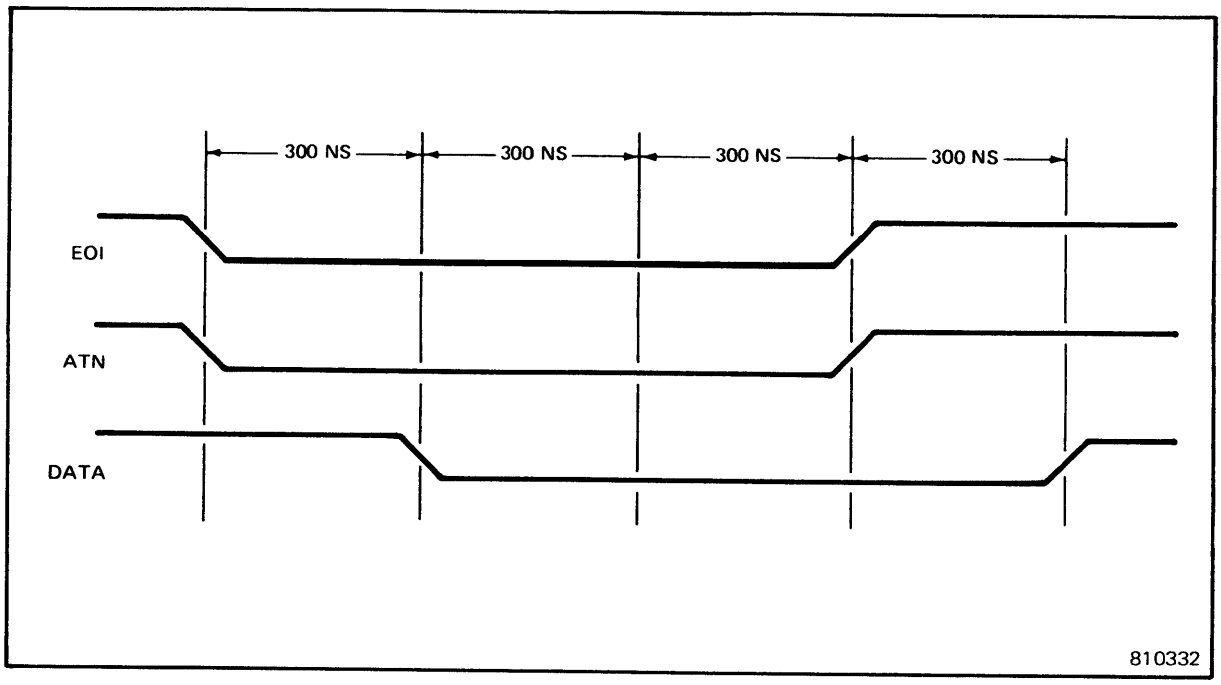


Figure 4-1. Parallel Polling Sequence Timing Diagram

4.2.2 Serial Polling Sequence

The IOP responds to a successful parallel poll by initiating a serial poll. The serial poll informs the IOP which one of the two multiplexers, as determined by the parallel poll, requested service. If both multiplexers had requested service the IOP would service the highest priority (lowest address) multiplexer. The address of the external device (input/output port) requesting service is also passed to the IOP during this operation.

The serial polling sequence is as follows:

1. The IOP places the serial poll enable (SPE) command onto the MP Bus. It then tests the MP Bus for a possible error condition. (Both the not ready for data (LNRFD) and the no data accepted (LNDAC) lines are at a logic high.) The IOP then activates the attention (LATN) line and all multiplexers enter the serial poll mode.

   The serial poll enable command format is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	0	0	1	1	0	0	0

2. The recognition of the SPE command places the multiplexers into the serial poll mode.

3. The IOP places the address of the even addressed multiplexer, as determined during the parallel polling sequence, onto the MP Bus. It then tests the MP Bus for an error condition. (Both the NRFD and the NDAC lines are at a logic high) and activates the LATN line.

4. If the even numbered multiplexer had requested service, in response to its address, it would place the contents of its data output register (a state transfer) on the MP Bus. The state transfer byte (containing information about the transfer being requested) is transferred to the IOP via the three-wire handshake protocol sequence.

Table 4-1
Poll Priorities

Priority	Address	Address Jumper Configuration	Assigned Data
		1 2 3 4	Line
0	0	- - - -	LDI 00
1	1	- - - X	LDI 00
2	2	- - X -	LDI 01
3	3	- - X X	LDI 01
4	4	- X - -	LDI 02
5	5	- X - X	LDI 02
6	6	- X X -	LDI 03
7	7	- X X X	LDI 03
8	8	X - - -	LDI 04
9	9	X - - X	LDI 04
10	A	X - X -	LDI 05
11	B	X - X X	LDI 05
12	C	X X - -	LDI 06
13	D	X X - X	LDI 06
14	E	X X X -	LDI 07
15	F	X X X X	LDI 07

Note: X = Jumper Installed
- = Jumper Removed

The state transfer format is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	S	R	B	0	A	A	A	A

Bit 0 = S (status ready) - Indicates that status is present in the data out register of the multiplexer.

1 = R (request for service) - Indicates that this multiplexer has requested service by activating the service request (LSRQ) to the IOP.

2 = B (byte transfer) - Indicates that the transfer requested should be one byte at a time. When this bit is not set it indicates that a halfword transfer is requested. Commands and status are always passed as bytes.

3 = Not used.

4-7 = A (device address) - Indicates the port, to which an external device is connected, that is requesting service.

The IOP accepts the state transfer and terminates the serial polling sequence.

If the even addressed multiplexer did not request service and no transfer is required, it responds to its address with the following state transfer:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	0	0	0	0	0	0	0

4. If the even numbered multiplexer had requested service, in response to its address, it would place the contents of its data output register (a state transfer) on the MP Bus. The state transfer byte (containing information about the transfer being requested) is transferred to the IOP via the three-wire handshake protocol sequence.

5. If the even addressed multiplexer had not requested service the IOP places the address of the odd addressed multiplexer, as determined during the parallel polling sequence onto the MP Bus. It then tests the MP Bus for an error condition (both the NRFD and the NDAC lines at a logic high) and activates the LATN line.

6. The odd addressed multiplexer responds to its address in a manner similar to that of the odd addressed multiplexer.

7. If the IOP determines that the odd addressed multiplexer did not request service (bit 6 of the state transfer not set), it recognizes that an error has occurred on the MP Bus, terminates the serial polling sequence, and processes the error.

4.2.3 Serial Polling Sequence Termination

The serial polling sequence is terminated when bit 6 (request for service) of the state transfer is recognized or when bit 6 is not set in either of the two addressed multiplexer state transfers.

The serial polling sequence termination is as follows:

1. The IOP loads the serial poll disable (SPD) command onto the MP Bus.

   The SPD command format is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	0	0	1	1	0	0	1

2. The IOP tests the MP Bus for a possible error condition (both the LNRFD and the LNDAC lines at a logic high).

3. The IOP activates the LATN line and all multiplexers exit the serial poll mode.

4.2.4 Handshake Protocol Sequence

The handshake protocol is a three-wire protocol - data valid (LDAV), not ready for data (LNRFD), and no data accepted (LNDAC) that is used for all information transfers between the MP Bus and the Eight-Line Asynchronous Communications Multiplexer.

Either the MP Bus or the Eight-Line Asynchronous Communications Multiplexer can be the source of the information transfer. The type of information transferred, using the three-wire handshake protocol, is as follows:

1. Serial polling
2. Data transfers
3. Address transfers
4. Command transfers
5. Status transfers

Figure 4-2, Handshake Protocol Timing Diagram, shows the events that occur during the sequence. The timing diagram shows two full cycles of the three-wire handshake protocol sequence. The explanation of the events is as follows:

P1 - Source initializes the data available (LDAV) to high.

P1 - Acceptor initializes the not ready for data (LNRFD) to low, and sets the not data accepted (LNDAC) low.

T1 - Source checks for error condition (both LNRFD and LNDAC high), then places data byte on LDI lines. At the same time the attention (LATN) goes low if the information being transferred is a command. The LATN remains high for data transfers.

P2 - Source delays allowing data to settle on LDI lines.

T2 - Acceptor has indicated readiness to accept first data byte, LNRFD goes high.

T3 - When the data is valid, and the source has sensed LNRFD high, LDAV is set low.

T4 - Acceptor sets LNRFD low indicating that it is no longer ready.

T5 - Acceptor sets LNDAC high indicating that it has accepted the data.

T6 - Source, having sensed that LNDAC is high, sets LDAV high. This indicates, to the acceptor, that data on the LDI lines is no longer valid. Upon completion of this step, one byte of data has been transferred. The LATN goes high if it had been set low at T1.

T7 - Acceptor, upon sensing LDAV high, sets LNDAC low in preparation for next cycle.

T8 - No significant events occur at this time.

T9 - Source checks for error condition (both LNRFD and LNDAC high) then places new data byte on LDI lines. At the same time the LATN goes low if the information being transferred is a command. The LATN remains high for data transfers.

P3 - Source delays allowing data to settle on LDI lines.

T10 - The acceptor indicates that it is ready for the next data byte by setting LNRFD high.

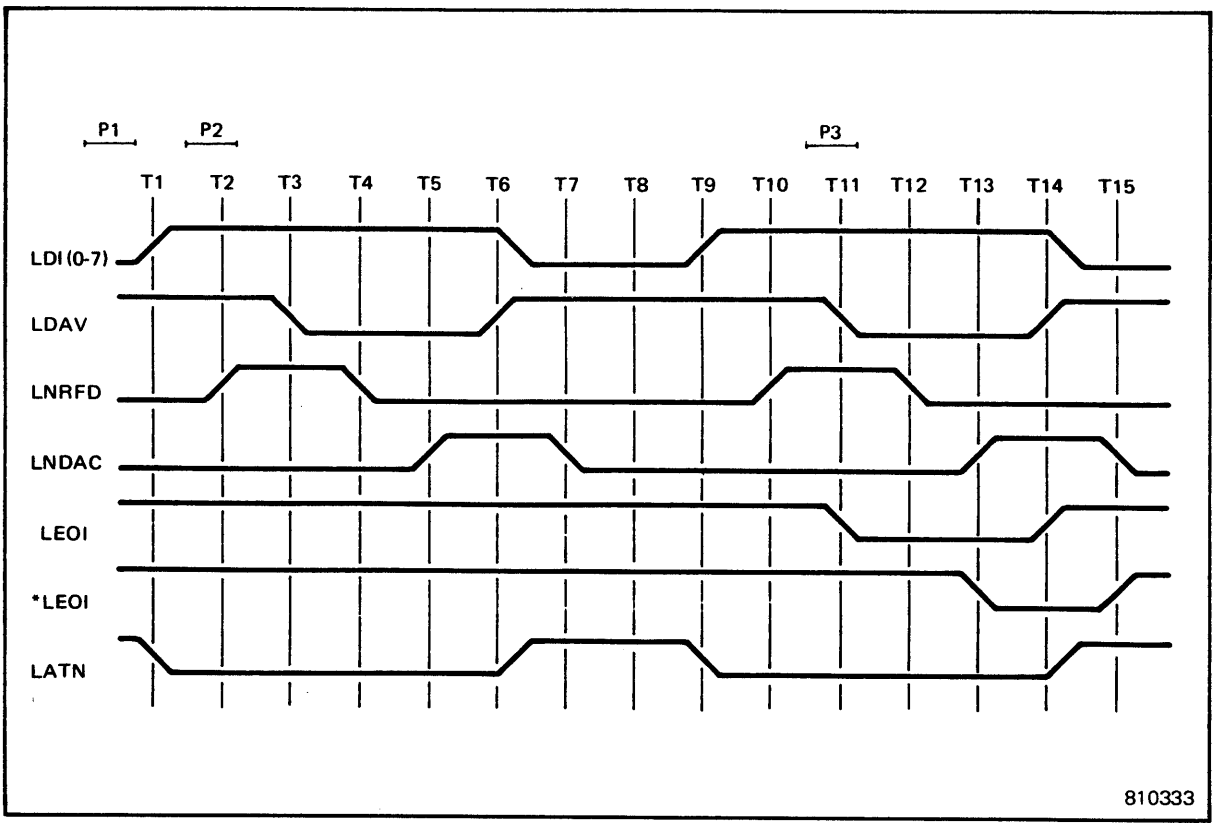


Figure 4-2. Handshake Protocol Timing Diagram

T11 - Source, upon sensing LNRFD high, sets LDAV low indicating that data on LDI lines is valid. Source indicates an end of burst transfer by setting the end or identify (LEOI) low.

T12 - Acceptor sets LNRFD low indicating that it is no longer ready.

T13 - Acceptor sets LNDAC high indicating that is has accepted the data. If the acceptor wishes to terminate the transfer it must set the LEOI line low during this cycle.

T14 - Source, having sensed that LNDAC is high, sets LDAC and LEOI high, and removes the data byte from the LDI lines. The LATN goes high if it had been set low at T9.

T15 - Acceptor, upon sensing LDAV high, sets LNDAC low and LEOI high in preparation for the next cycle.

NOTE

All three handshake lines (LDAV, LNRFD, and LNDAC) return to their initialized states.

4.3 Information Transfer Sequences

4.3.1 Status Transfer Sequence

Status is transferred to the MP Bus from an Eight-Line Asynchronous Communications Multiplexer as part of the serial polling sequence, when the IOP requests a device status report (using the extended I/O sense command), or in response to an end of transfer caused by a currently controlled device abnormally going out of service while a data input transfer is in progress.

Status transfers, in response to the serial polling sequence and an end of transfer, are initiated by an SRQ from the Eight-Line Asynchronous Communications Multiplexer.

Status transfers are also initiated in response to an IOP generated extended I/O sense command.

The SRQ initiated status transfer is as follows:

1. The multiplexer is recognized by the IOP, using the SRQ protocol sequence.

2. During the serial polling sequence, the state transfer byte will contain information indicating that status is available in the data out register of the MPC interface.

   The state transfer format is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	1	1	1	0	A	A	A	A

Bit 0 = Status ready - Indicates that status is present in the data out register of the multiplexer (MPC interface).

1 = Request for service - Indicates that this multiplexer requested service by activating the LSRQ line to the IOP.

2 = Byte transfer - Indicates that the information will be transferred to the IOP one byte at a time. Since status is always transferred one byte at a time this bit must be set for status transfers.

3 = Not used.

4-7 = A (device address) - Indicates the port, to which an external device is connected, that is requesting service.

3. Status, from the data out register of the multiplexer (MPC interface) is then transferred to the IOP using the three-wire handshake protocol sequence.

The format for the status bytes is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	B	S	C	A	E	D	U	X

Bit 0 = B (busy)
1 = S (status modifier)
2 = C (controller end)
3 = A (attention)
4 = E (channel end)
5 = D (device end)
6 = U (unit check)
7 = X (unit exception)

Once the multiplexer is recognized and an information transfer started, the multiplexer is inhibited from activating the LSRQ line until the IOP or the Eight-Line Asynchronous Communications Multiplexer terminates the information transfer with the end or identify signal (LEOI).

4.3.2 Control Signal Transfer Sequence

The control signal transfer sequence is as follows:

1. The IOP places listen address onto the MP Bus with the LATN line active. The multiplexer then accepts the listen address using the three-wire handshake protocol sequence. This action sets up the multiplexer as a receiver of information from the MP Bus.

   The format of the listen address is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	0	1	0	C	C	C	C

   Bits 0-3 = Establishes the listen mode (receive) for the selected (addressed) multiplexer.

   4-7 = C is the address bits of the selected multiplexer.

2. The IOP places one of two control signals onto the MP Bus with the LATN line active. The control signals are accepted by the multiplexer using the three-wire handshake protocol sequence. The two possible control signals are

   1. End of message (EOM) - The EOM signal indicates, to the listening multiplexer, that no more data is to be transferred to or from the addressed external device (terminal or modem).

   2. Device clear (DCR) - The DCR signal causes the addressed external device (terminal or modem) to be cleared. The Halt I/O (HIO) machine instruction causes DCR to be generated.

   The format for the EOM control signal is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	0	0	0	0	0	0	0

The format for the DCR control signal is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	0	0	0	1	1	0	0

3. The IOP places the device address onto the MP Bus with the LATN line active. The multiplexer then accepts the device address using the three-wire handshake protocol sequence.

   The format for the device address is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	1	1	0	A	A	A	A

   Bits 0-3 = Informs the multiplexer that this is a device address byte.

   4-7 = The device address (the port of the multiplexer to which an external device, terminal or modem, is connected).

4. The IOP places the unlisten address onto the MP Bus, with the LATN line active. The multiplexer accepts the unlisten address using the three-wire handshake protocol sequence.

   The format of the unlisten address is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	0	1	1	1	1	1	1

   Note that the controller address LDI (bits 4 through 7), address a nonexistent multiplexer causing all multiplexers to receive the unlisten address.

NOTE

Each of the control signals, EOM and DCR, will be passed to the MPU via the command pass through (CPT) register of the talker/listener (MPC interface). The availability of the control signals in the CPT register is indicated by the CPT status bit in interrupt status register 1 of the talker/listener.

4.3.3 Reset Transfer Sequence

The multiplexer reset sequence is as follows:

1. The IOP places the listen address for a selected multiplexer onto the MP Bus with the LATN line active. The multiplexer accepts the listen address using the three-wire handshake protocol sequence.

The format of the listen address is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	1	0	0	C	C	C	C

Bits 0-3 = Establishes the listen mode (receive) for the selected (addressed) multiplexer.

4-7 = C is the address bits of the selected multiplexer.

2. The IOP places the selected multiplexer clear (SCC) signal onto the MP Bus, with the LATN line active. The addressed multiplexer accepts the SCC using the three-wire handshake protocol sequence.

   The format of the SCC signal is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	0	1	0	0	0	0	0

3. The IOP places the unlisten address onto the MP Bus, with the LATN line active. The multiplexer accepts the unlisten address using the three-wire handshake protocol sequence.

   The format of the unlisten address is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	0	1	1	1	1	1	1

   Note that the multiplexer address LDI (bits 4 through 7) addresses a nonexistant multiplexer causing all multiplexers to receive the unlisten address.

NOTE

The selected multiplexer clear control signal is interpreted by the talker/listener (MPC Interface) and indicated to the MPU by the DEC (multiplexer clear active state) bit of interrupt status register 1 of the talker/listener.

4.3.4 Command Transfer Sequence

The command transfer sequence is as follows:

1. The IOP places the listen address onto the MP Bus, with the LATN line active. The multiplexer accepts the listen address using the handshake protocol sequence.

The format for the listen address is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	0	1	0	C	C	C	C

Bits 0-3 = Establishes the listen mode (receive) for the selected (addressed) multiplexer.

4-7 = C is the address bits of the selected multiplexer.

2. The IOP places the enable secondary command (ESC) onto the MP Bus, with the LATN line active. The selected multiplexer accepts the ESC command using the handshake protocol sequence.

   The format of the ESC command is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	0	0	0	1	1	1	1

3. The IOP places the most-significant nibble of the command byte onto the MP Bus with the LATN line active. The multiplexer accepts the nibble of the command byte using the handshake protocol sequence.
   The format of the secondary command, containing the most-significant nibble of the extended I/O command byte, is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	1	1	1	N	N	N	N

   Bits 0-3 = Indicates that this byte is a secondary command and contains a nibble of the extended I/O command byte.

   4-7 = N is a nibble (most-significant) of the extended I/O command byte.

4. The IOP places the least-significant nibble of the command byte onto the MP Bus with the LATN line active. The multiplexer accepts the nibble of the command byte using the three-wire handshake protocol sequence.

   The format of the least-significant nibble of the command byte is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	1	1	1	N	N	N	N

   Bits 0-3 = Indicates that this byte contains a nibble of the command byte.

   4-7 = N is a nibble (least-significant) of the command byte.

5. The IOP places the device address onto the MP Bus, with the LATN line active. The multiplexer accepts the device address using the three-wire handshake protocol sequence.

The format of the device address is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	1	1	0	A	A	A	A

Bits 0-3 = Informs the multiplexer that this is a device address byte.

4-7 = A is the device address (the port of the multiplexer to which an external device, terminal, or modem, is connected).

6. The IOP places the unlisten address onto the MP Bus, with the LATN line active. The multiplexer accepts the unlisten address using the handshake protocol sequence.

   The format of the unlisten address is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	0	1	1	1	1	1	1

NOTE

The ESC control signal will be passed to the MPU via the CPT register of the talker/listener (IOP interface). Both nibbles of the command byte and the device address will also be passed to the MPU via the CPT register.

4.3.5 Extended I/O Command Transfer Sequence

Each extended I/O command transfer requires an ESC transfer followed by two secondary command transfers. Each of the secondary commands contain a nibble of the extended I/O command byte.

The extended I/O commands are:

1. Sense command
2. Read command
3. Write command
4. Control command

The format of the extended I/O command nibbles is shown in Table 4-2.

The enable secondary command and the two secondary commands are transferred to the multiplexer using the command transfer sequence.

4.3.6 Data Transfer Sequence

Data transfers are bidirectional and either the MP Bus or the Eight-Line Asynchronous Communications Multiplexer can be the source for the data transfers. The data output transfers data from the MP Bus to the multiplexer while data input transfers data from the multiplexer to the MP Bus. The terms input and output are defined in reference to the MP Bus.

Table 4-2
Extended I/O Commands (Sheet 1 of 2)

Function	Nibble 1				Nibble 2				
Sense Commands									
	0	1	2	3	0	1	2	3	LDI (Bit)
Sense Status	0	0	0	0	0	1	0	0	State
Read Commands									
	0	1	2	3	0	1	2	3	LDI (Bit)
Read	0	A	S	P	0	0	1	0	State
Read echoplex	0	A	S	P	0	1	1	0	State
Bit 0 = P, purge Bit 1 = S, special character detect Bit 2 = A, ASCII control character detect									
Write Commands									
	0	1	2	3	0	1	2	3	LDI (Bit)
Write	0	0	0	0	0	0	0	1	State
Write with input subchannel (port) monitoring	0	0	0	0	0	1	0	1	State
Control Commands									
	0	1	2	3	0	1	2	3	LDI (Bit)
No operation (NOP)	0	0	0	0	0	0	1	1	State
Define UART parameters (three bytes follow)	1	1	1	1	1	1	1	1	State
Define special character (byte C follows)	0	0	0	0	1	0	1	1	State
Reset data terminal ready	0	0	0	1	0	0	1	1	State
Set data terminal ready	0	0	0	1	0	1	1	1	State
Reset request to send	0	0	0	1	1	0	1	1	State

Table 4-2
Extended I/O Commands (Sheet 2 of 2)

Function	Nibble 1				Nibble 2				
	0	1	2	3	0	1	2	3	LDI (Bit)
Set request to send	0	0	0	1	1	1	1	1	State
Reset break	0	0	1	1	0	0	1	1	State
Set break	0	0	1	1	0	1	1	1	State

4.3.6.1 Data Output (Write) Transfer Sequence

The data output transfer sequence is used to transfer data from the MP Bus to the multiplexer. The sequence is as follows:

1. The IOP places the listen address for a specific multiplexer onto the MP Bus with the LATN line low. The listen address is accepted by the multiplexer using the three-wire handshake protocol sequence. The format of the listen address is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	0	1	0	C	C	C	C

   Bits 4 through 7 = C is controller address bit.

2. After the multiplexer is addressed to listen, data is transferred from the MP Bus using the three-wire handshake protocol sequence, with the LATN line high, for each byte of data. The format for the data is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	D	D	D	D	D	D	D	D

   Bits 0 through 7 = D is data bit.

3. Each data byte is transferred to the multiplexer using the three-wire handshake protocol sequence. On the last data byte the IOP activates the EOI line to indicate the end of the current block transfer.

4. After the last byte of data is transferred to the multiplexer, the IOP places the unlisten address on the MP Bus with the LATN line low. The unlisten address is accepted by the multiplexer using the three-wire handshake protocol sequence. The format of the unlisten address is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	0	1	1	1	1	1	1

   Note that the controller address, LDI (bit) 4 through 7, address a nonexistent multiplexer causing all multiplexers to stop listening.

4.3.6.2 Data Input (Read) Transfer Sequence

The input (read) transfer sequence is used to transfer data from the multiplexer to the multipurpose bus (MP Bus). The sequence is as follows:

1. The IOP places the talk address for a specific multiplexer onto the MP Bus with the LATN line low. The multiplexer accepts the talk address using the handshake protocol sequence. The format of the talk address is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	1	0	0	C	C	C	C

   Bits 4-7 = C is multiplexer address bit.

2. After the multiplexer is addressed to talk, data is transferred to the MP Bus using the handshake protocol sequence, with the LATN line high, for each byte of data. The format for the data is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	D	D	D	D	D	D	D	D

   Bits 0-7 = D is data bit.

3. Each data byte is transferred to the MP Bus using the three-wire handshake protocol sequence. On the last data byte the multiplexer activates the EOI line to indicate the end of the current block transfer.

   The multiplexer may also activate the EOI line at any time during the data input transfer sequence to indicate the end of the transfer, when the currently controlled external device (terminal or modem) abnormally goes out of service.

4. After the last byte of data is transferred to the MP Bus, the IOP places the untalk address on the MP Bus with the LATN line low. The untalk address is accepted by the multiplexer using the three-wire handshake protocol sequence. The format for the untalk address is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	0	1	0	1	1	1	1	1

4.3.7 Status Byte

Status is transferred from a multiplexer to the MP Bus as part of the serial polling sequence, when the IOP requests a device status report (using the extended I/O sense command), or in response to an end of transfer caused by a currently controlled device abnormally going out of service while a data input transfer is in progress.

The format for the status byte is as follows:

LDI (Bit)	0	1	2	3	4	5	6	7
State	B	S	C	A	E	D	U	X

Bit 0 = B (busy)
1 = S (status modifier)
2 = C (multiplexer end)
3 = A (attention)
4 = E (channel end)
5 = D (device end)
6 = U (unit check)
7 = X (unit exception)

1. Busy (bit 0) - This bit indicates that the external device (terminal or modem) or the multiplexer cannot execute an operation because it is in the process of executing a previously initiated operation.

2. Status modifier (bit 1) - This bit is not used.

3. Multiplexer end (bit 2) - This bit is not used.

4. Attention (bit 3) - This bit is not used.

5. Channel end (bit 4) - A CE condition is caused by the completion of a portion of the I/O operation involving data or control information being transferred between the external device and the IOP. This condition indicates that the subchannel (port) has become available for another operation.

6. Device end (bit 5) - The device end indication is caused by the completion of a device I/O operation. This status condition indicates that the external device has become available for use for another operation.

7. Unit check status (bit 6) - Unit check status indicates that the external device has detected an unusual condition detailed by the information available to the sense command. Unit check status provides a summary indication of the condition identified in the sense data.

   An error condition causes the unit check status indication only when it occurs during the execution of a command or during some activity associated with an I/O operation. Errors detected during the execution of a command are reported by unit check status accompanied by a CE and/or DE command, depending on when the condition was detected. If an unusual condition was detected during the initiation of a command, the command is terminated without device end status.

   The unit check indication is caused by the following conditions:

   A. Loss of a data set ready (DSR) indication.

   B. Loss of a data carrier detected (DCD) indication.

   C. An attempt, by software, to transfer data before a DSR indication is received.

8. Unit exception (bit 7) - Unit exception indicates that a break interrupt signal was received while a read operation was in progress. The CE/DE bits are set with this condition.

4.4 Functional Description

Figure 4-3 is a detailed block diagram of the Eight-Line Asynchronous Communications Multiplexer. Figure 4-3 shows the operational blocks of the multiplexer, along with the signal flows. The sheet numbers within the operational blocks represent the sheet number of the logic diagram (130-103768) where the logic details are shown.

In most cases the signal mnemonics used on the detailed block diagram are identical to those used on the logic diagram. However, some appear as minor variations of the mnemonics used on the logic diagram. For instance, the HCLK and LCLK signals on the logic diagram are represented simply as CLK on the block diagram.

The functional blocks are as follows:

1. MPC interface
2. I/O address decode
3. Interrupt logic
4. Memory
5. MPC EOI controller
6. DMA controller
7. MPU
8. Port 0 through 7

4.4.1 MPC Interface

The MPC interface is set up by the MPU for information transfers between the MP Bus and the multiplexer. The MPC interface contains the controls necessary to effect these information transfers.

4.4.2 I/O Address Decode

The I/O address decode logic (sheet 4) uses the MPU address to select one of the ACEs (sheets 11, 12, 13 and 14) for I/O operations.

4.4.3 Interrupt Logic

This circuit interprets the interrupt, inputs into the interrupt handler and memory address, which defines the interrupt handler. The MPU uses this memory address to access the stored interrupt handler and to process the interrupt.

4.4.4 Memory

The memory contains the firmware and temporary storage required by the MPU for controlling the operation of the multiplexer.

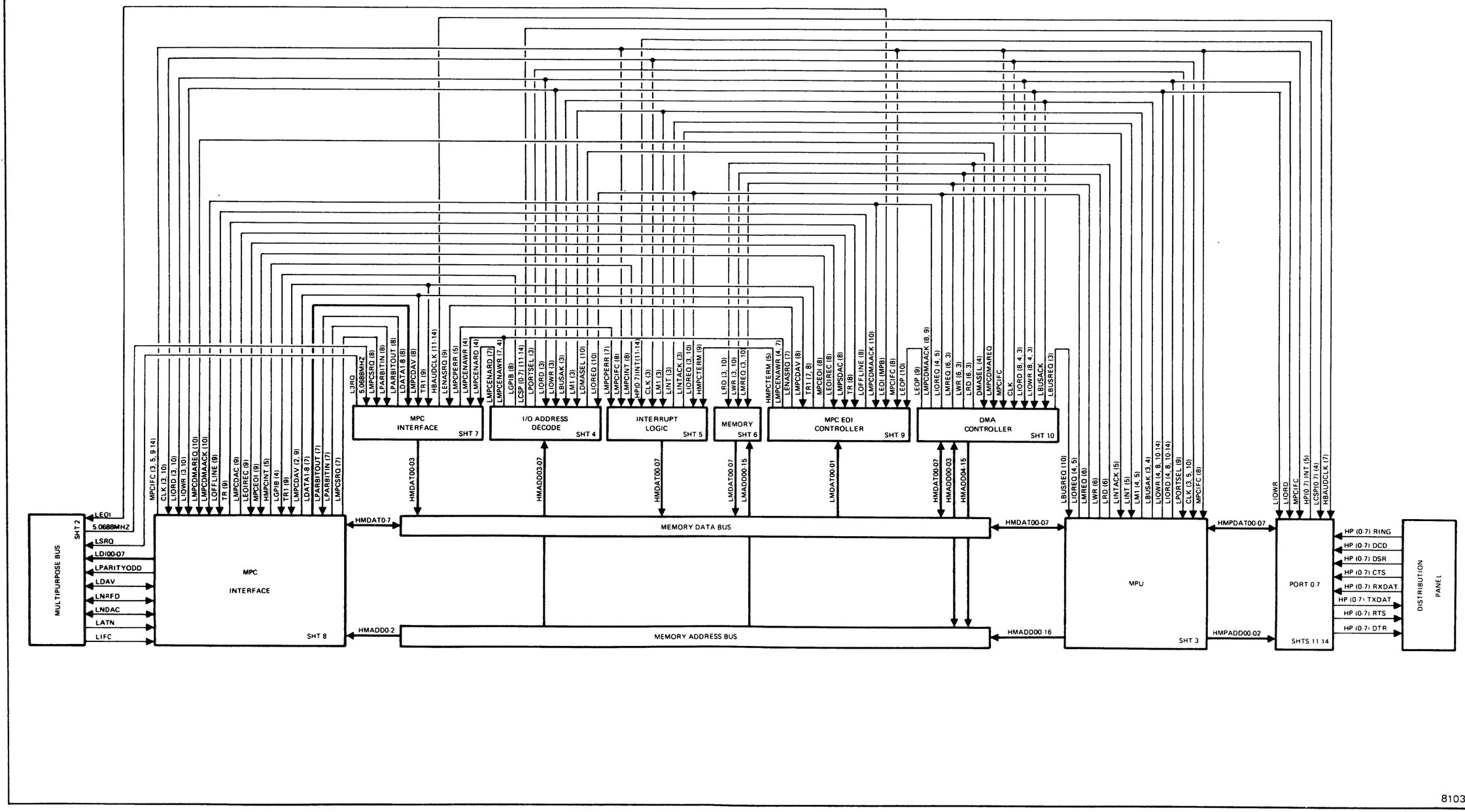


Figure 4-3. Functional Block Diagram

4.4.5 MPC EOI Controller

The MPC EOI controller enables and disables the detection and generation of the EOI signal to the MP Bus.

4.4.6 DMA Controller

The DMA controller performs information transfers between the MPC interface and memory. The MPU programs the DMA controller to accomplish the transfers. The transfers are performed without further intervention from the MPU.

4.4.7 MPU

The microprocessor unit (MPU) controls the information transfers between the multiplexer and the MP Bus, and between the multiplexer and the distribution panel. The MPU controls the information transfers via firmware programs stored in the memory.

4.4.8 Port 0 through 7

There are eight identical ports (0 through 7) used for connecting external devices (terminals or modems) to the multiplexer, via a distribution panel. The ports control the information transfers between the external device and the MPU, perform a parallel-to-serial conversion on data characters received from the MPU, and perform a serial-to-parallel conversion on data characters received from the external device.

4.5 Detailed Circuit Description

The Eight-Line Asynchronous Communications Multiplexer is an interrupt driven firmware controlled interface between a maximum of eight asynchronous terminals or modems and the IOP via the MP Bus.

The following discussion presents a detailed circuit description of the Eight-Line Asynchronous Communications Multiplexer, with reference to the 130-103768 logic diagrams. Additional information regarding reference designators, logic symbols, logic levels, propagation delays, and connector designation conventions may be found in the CONCEPT/32 Circuit Registration Manual, publication number 313-000670.

4.5.1 MPC Connector (Sheet 2)

Sheet 2 shows a portion of the signals which interface to the Input/Output Processor (IOP).

4.5.2 Microprocessor Unit (Sheet 3)

Sheet 3 shows the Microprocessor Unit (MPU), at F13. The MPU associated control and data lines and the clock generator are also shown. The MPU is the heart of the Eight-Line Asynchronous Communications Multiplexer and it controls the data transfer process.

In the executive idle loop the MPU tests bits and checks pointers to decide which tasks must be performed and which buffers to process.

4.5.3 Clock Generator

The clock generator is comprised of a crystal oscillator and a flip-flop at E10. The flip-flop divides the oscillator frequency in half to provide a clock (LCLK, HCLK, and HMPUCLK) to the MPU and other components of the Eight-Line Asynchronous Communications Multiplexer. The clock provides the onboard timing for the Eight-Line Asynchronous Communications Multiplexer.

4.5.4 MPU Input Signals

4.5.4.1 Interrupt Request (INT)

The INT signal at pin 16 is driven by the interrupt signal (LINT) from the interrupt logic. A request will be honored at the end of the current instruction if the internal software controlled interrupt enable flip-flop is enabled, and if the bus request (LBUSRQ) is not active. When the MPU accepts an interrupt, an acknowledge signal (IORQ, at pin 20, during M1 time) is generated at the beginning of the next instruction cycle.

4.5.4.2 Clock (CLK)

The clock, at pin 6, provides the internal timing for the MPU.

4.5.4.3 Reset

The reset signal at pin 26 is driven by the MPC interface clear (LMPCIFC) signal. It forces the program counter to zero and initializes the MPU. The initialization procedure includes

1. Disable the interrupt enable flip-flop
2. Set register I = 00
3. Set register R = 00
4. Set interrupt mode 0

During reset time, the address bus and data bus go to a high impedance state and all control output signals go to the inactive state. No refresh occurs.

4.5.4.4 Bus Request (BUSREQ)

The BUSREQ signal at pin 25 is driven by the bus request (LBUSREQ) signal from the DMA controller. The bus request signal is used to request the MPU address bus, data bus and tri-state output control signals to go to a high impedance state so that other devices can control these buses. When LBUSRQ is activated, the MPU will set these buses to a high impedance state as soon as the current MPU machine cycle is terminated.

4.5.5 MPU Output Signals

4.5.5.1 Input/Output Request (IORQ)

The IORQ signal at pin 20 is applied to the clear input of the flip-flop at E10. The IORQ signal indicates that the lower half of the address bus holds a valid I/O address for a IORD or IOWR operation. An IORQ signal is also generated with a machine cycle one (M1) signal when an interrupt is being acknowledged (indicating that an interrupt response vector can be placed on the data bus). Interrupt acknowledge operations occur during M1 time, while I/O operations never occur during M1 time.

4.5.5.2 Read (RD)

The RD signal at pin 21 is applied to an input of the buffer at F12. The RD signal indicates that the MPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the MPU data bus.

4.5.5.3 Write (WR)

The WR signal at pin 22 is applied to an input of the buffer at F12. The WR signal indicates that the MPU data bus holds valid data to be stored in the addressed memory or I/O device.

4.5.5.4 Refresh (RFSH)

The RFSH signal at pin 28 is applied to an input of the buffer at F12. The RFSH signal indicates that the lower seven bits of the address bus contain a refresh address for dynamic memories and a current MREQ signal should be used to do a refresh read to all dynamic memories. Note that A7 is a logic zero and the upper eight bits of the address bus contains the I register.

4.5.5.5 Machine Cycle One (M1)

The M1 signal at pin 27 is applied to an input of the buffer at F12. The M1 signal indicates that the current machine cycle is the op code fetch cycle of an instruction execution. Note that during execution of two-byte op codes, M1 is generated as each op code byte is fetched. These two-byte op codes always begin with CBH, DDH, EDH, or FDH. The M1 signal also occurs with IORQ to indicate an interrupt acknowledge cycle.

4.5.5.6 Memory Request (MREQ)

The MREQ signal at pin 19 is applied to an input of the buffer at F12, and to a gate at E09. The memory request signal indicates that the address bus holds a valid address for a memory read or memory write operation.

4.5.5.7 Bus Acknowledge (BUSAK)

The BUSAK signal at pin 23 is applied to the enable input of the buffer, through the inverters. The bus acknowledge is used to indicate to the requesting device that the MPU address bus, data bus and three-state control bus signals have been set to their high impedance state and the external device cannot control these signals. While the MPU is in a bus acknowledge condition, dynamic memory refresh will not occur.

4.5.5.8 Address Bits (A0 through A15)

The address bits constitute a 16-bit address bus. The address bus provides the address for memory and I/O device data exchanges. I/O addressing uses the lower eight address bits with A0 being the least-significant address bit. During refresh time, the seven address bits contain a valid refresh address.

Address bits A0 through A3 are applied to the inputs of gated buffers at E11.

The address bits from the MPU generate the memory address bits (HMADD0 through HMADD15) on the address bus.

4.5.6 Input/Output Signals (Data Bits D0 through D7)

The data bits constitute an eight-bit bidirectional data bus used for data exchanges between memory and I/O devices.

The data bits are transmitted/received on the MPU data bus (HMPDAT0 through HMPDAT7) and on the memory data bus (HMDAT0 through HMDAT7), through the transceiver at G10.

4.6 Buffers

All of the active signals from the MPU, which are applied to the octal buffers and line drivers, produce active signals at the output, if the buffers are enabled.

4.6.1 Read (RD)

The RD signal from the MPU is applied to two buffer inputs to generate two reads, read (LRD) and input/output read (LIORD) signals when the buffer is enabled. The LRD signal is used to read from memory while the LIORD signal is used to read from an I/O chip.

4.6.2 Write (WR)

The WR signal from the MPU is applied to two buffer inputs to generate two writes, write (LWR) and input/output write (LIOWR) signals when the buffer is enabled. The LWR signal is used to write into memory while the LIOWR signal is used to write to an I/O chip.

4.6.3 Input/Output Request (IOREQ)

Beside being applied to the MPU, the MPC interface clear signal is used to preset the flip-flop, at E10. The set output of the flip-flop is applied to an input of the buffer at F12.

The set output of the flip-flop will go inactive with an active I/O request from the MPU and will generate an active I/O request (LIOREQ) signal through the enabled buffer, to the interrupt logic and I/O address decoder. The LIOREQ signal will go inactive on the next positive going clock (HCLK) after the IOREQ signal from the MPU goes inactive.

4.6.4 Enable

The buffers at locations E11 and F12 are enabled when the BUSAK signal from the MPU is inactive (no I/O chip is using the bus).

4.7 Transceiver

Data (HMPDAT0 through HMPDAT7 and HMDAT0 through HMDAT7) through the transceiver is bidirectional. The direction of data flow is determined by state of the direction (DIR) line at pin 1. The transceiver is enabled by an active low signal at pin 19.

4.7.1 Direction (DIR)

Data can be transferred from the A-port to the B-port, when the DIR line is a logic high, by an active interrupt acknowledge (LINTACK) signal or an active RD signal from the MPU. Data can be transferred from the B-port to the A-port when both the LINTACK and RD lines are inactive.

4.7.2 Enable

The transceiver is enabled if the port select (LPORTSEL) and LBUSAK lines are not active, and the MREQ and I/O request lines from the MPU, are not active.

4.8 I/O Address Decode (Sheet 4)

The decoder at E08 is enabled when the MPU has control of the address bus, LBUSAK line is inactive, the MPU is not fetching an op code, LM1 is inactive, the LIOREQ line is active, and address bit 7 (HMADD07) is inactive.

The decoder at E06 is enabled when the decoder at E08 is enabled and at least one of the decoded outputs (Y2 through Y5) is active.

Memory address bits (HMADD03 through HMADD05) are applied to the inputs of the decoder at E06 and memory address bits (HMADD04 through HMADD06) are applied to the inputs of the decoder at E08. The address lines are decoded when the chips are enabled. Table 4-3, Decoded Address Bits, shows the relationship between the address bits and the signals that are generated when they are decoded.

Table 4-3
Decoded Address Bits

HMADD Bit HEX	7	6	5	4	3	Signal Generated
00	0	0	0	0	0	None
08	0	0	0	0	1	None
10	0	0	0	1	X	DMA select (DMASEL)
20	0	0	1	0	0	Chip select port 0 (LCSP0) and port select (LPORTSEL)
28	0	0	1	0	1	Chip select port 1 (LCSP1) and port select (LPORTSEL)
30	0	0	1	1	0	Chip select port 2 (LCSP2) and port select (LPORTSEL)
38	0	0	1	1	1	Chip select Port 3 (LCSP3) and port select (LPORTSEL)
40	0	1	0	0	0	Chip select port 4 (LCSP4) and port select (LPORTSEL)
48	0	1	0	0	1	Chip select port 5 (LCSP5) and port select (LPORTSEL)
50	0	1	0	1	0	Chip select port 6 (LCSP6) and port select (LPORTSEL)
58	0	1	0	1	1	Chip select port 7 (LCSP7) and port select (LPORTSEL)
60	0	1	1	0	X	MPC enable write (LMPCENAWR) if I/O write (LIOWR) is active, or MPC enable read (LMPCENARD) if I/O read (LIORD) is active
70	0	1	1	1	X	GPIB chip select (LGPIB)
80	1	X	X	X	X	None

NOTE

X = State is irrelevant to signal generated.

4.9 Interrupt Logic (Sheet 5)

The interrupt logic receives interrupts from the I/O interfaces and generates an interrupt vector address to the MPU. The MPU uses the vector address to locate and implement the appropriate interrupt handler. The interrupts are prioritized and the interrupt logic insures that the MPU processes the highest priority interrupt first.

Interrupts may come at any time during the idle loop, and will be processed immediately. Two interrupt masks exist in the 8291 and are used to selectively enable and disable MPC and DMA interrupts. Each UART contains its own interrupt mask, allowing the MPU to enable/disable transmitter, receiver, modem, and line change interrupts.

The interrupt structure implemented in the Eight-Line Asynchronous Communications Multiplexer utilizes mode 2 of the Z80 microprocessor. This operating mode allows the control program to establish a 256-byte page of memory as a vector table. Each even/odd pair of memory locations in this page can contain a vector to an interrupt handler.

The vector table page is defined by the I register in the Z80. The vector location to be used, when an interrupt is received, is determined by a byte of information returned from the interrupt logic. This byte identifies which byte is the highest requesting level. The most-significant seven bits of this byte are merged with the contents of the I register to point to the proper interrupt vector.

The interrupt vector locations are assigned as described below.

1100	LSB	MPC parity error	110B	MSB	Port 4
1101	MSB	MPC parity error	110C	LSB	Port 5
1102	LSB	Port 0	110D	MSB	Port 5
1103	MSB	Port 0	110E	LSB	Port 6
1104	LSB	Port 1	110F	MSB	Port 6
1105	MSB	Port 1	1110	LSB	Port 7
1106	LSB	Port 2	1111	MSB	Port 7
1107	MSB	Port 2	1112	LSB	MPC interface
1108	LSB	Port 3	1113	MSB	MPC interface
1109	MSB	Port 3	1114	LSB	MPC terminate
110A	LSB	Port 4	1115	MSB	MPC terminate

4.9.1 Clear

The quad-D flip-flops are initialized (cleared) by the MPC interface clear (LMPCIFC) line.

4.9.2 Address Generation

The eleven I/O interrupts are applied to the inputs of three quad-D flip-flops at G05, G03 and G06. The interrupts are clocked into the flip-flops on the trailing edge of the clock (HCLK) pulse when the MPU is not in an op code fetch cycle of an instruction execution (LM1 is inactive).

LM1, in the inactive state, also disables the buffer at G08 through the gate at C15. This action prevents the decoded interrupt address from being placed on the data lines.

The outputs of the flip-flops are applied to the inputs of two eight-line-to-three-line priority encoders at G04 and G07. The encoders ensure that only the highest order data line is encoded. The enable out (EO) line of the encoder at G04 disables the encoder at G07, if any of the inputs to the encoder at G04 are active. This action allows the priority encoders to be cascaded and still maintain the priority scheme. The GS outputs of encoders are active whenever the EI, and at least one data input line, is active.

The encoded outputs are applied directly to the inputs of a buffer at G08. The GS outputs of the encoders generate the interrupt (LINT) signal to the MPU and bit 5 (HMDAT5) of the interrupt address. The GS output of the encoder at G04 also generates bit 4 (HMDAT4) of the interrupt address.

Because the buffer inputs, which generate interrupt address bits 0, 6, and 7 (HMDAT0, 6, and 7) are tied to ground, bits 0, 6, and 7 (HMDAT0, 6 and 7) will always be low.

The MPU responds to the interrupt (LINT) signal by generating an I/O request (LIOREQ) signal, during an LM1 time to the interrupt logic. These conditions generate the interrupt acknowledge (LINTACK) signal, enables the buffer at G08, and permits the interrupted address to be impressed on the memory data bus (HMDAT0 through HMDAT7).

Table 4-4, Interrupt Priority/Address, shows the priority level and the interrupt binary address generated by each interrupt. Priority level zero is the highest priority in the priority structure.

4.10 Memory (Sheet 6)

Memory is comprised of three, 2KX8, programmable read only memory (PROM) chips and two, 2KX8, random access memory (RAM) chips. The PROMs, sometimes referred to as control read only memory (CROM), contain the firmware which controls the operation of the multiplexer.

The 2K RAMs are divided into 16 data buffers; one 128-character buffer for each port's input and one 128-character buffer for each port's output. The remaining bytes of memory are used to store status for each port as well as flags, pointers, byte counts, look-up tables, and constants.

Table 4-4
Interrupt Priority/Address

Interrupt Mnemonic	Priority Level	Interrupt Address HMDAT (Bit) 7 6 5 4 3 2 1 0	Hex
HMPCERR	0	0 0 0 0 0 0 0 0	00
HP0INT	1	0 0 0 0 0 0 1 0	02
HP1INT	2	0 0 0 0 0 1 0 0	04
HP2INT	3	0 0 0 0 0 1 1 0	06
HP3INT	4	0 0 0 0 1 0 0 0	08
HP4INT	5	0 0 0 0 1 0 1 0	0A
HP5INT	6	0 0 0 0 1 1 1 0	0C
HP6INT	7	0 0 0 0 1 1 1 0	0E
HP7INT	8	0 0 0 1 0 0 0 0	10
HMPCINT	9	0 0 0 1 0 0 1 0	12
HMPCTERM	10	0 0 0 1 0 1 0 0	14

4.10.1 Memory Chip Select (Addressing)

The individual memory chips are selected by decoding the five most-significant bits of the memory address (HMADD11 through HMADD15). Memory address bits 11 through 13 are applied to the three-line-to-eight-line decoder/demultiplexer. The decoder is enabled when memory address bits 14 and 15 (HMADD14 and HMADD15) are inactive and memory request is active.

Memory address bits 11 through 13 (HMADD11 through HMADD13) are applied to the inputs to the decoder. When enabled, it decodes the address bits and activates the appropriate output line to enable one of the five memory chips. Table 4-5 Memory Chip Select Addresses, shows the relationship between memory address bits 11 through 15 and the selected memory chip.

Memory address bits 0 through 10 (HMADD0 through HMADD10) are applied directly to the address inputs of the PROMs and the RAMs. These address bits define a location in the selected memory chip from where data will be read when the read (LRD) line is active. These bits also define a location in a RAM, when selected, to which data will be written when the write (LWR) line is active.

4.10.2 Memory Data

In the read operation, memory data bits (HMDAT0 through HMDAT7) are applied directly to the memory data bus.

In the write operation, memory data bits (currently on the memory data bus) are stored in the selected memory location.

4.11 MPC Interface (Sheet 7)

The MPC interface, sheet 7, shows the generation of the service request (LSRQ) signal to the MP Bus, the Eight-Line Asynchronous Communications Multiplexer address generator, the parity generator/checker, and the baud clock generator.

Table 4-5
Memory Chip Select Addresses

HMADD (Bit) 15	14	13	12	11	Memory Chip Selected
0	0	0	0	0	PROM, F02
0	0	0	0	1	PROM, F03
0	0	0	1	0	PROM, F05
0	0	0	1	1	RAM, F06
0	0	1	0	0	RAM, F10

4.11.1 Service Request (LSRQ)

The MPC service request (LMPCSRQ) signal is applied to the MP Bus through the noninverting 75453 driver at B15 to generate the service request (LSRQ) signal. The driver is enabled by the enable service request (LENASRQ) signal.

4.11.2 Address Generator

The address generator provides a means of assigning a physical address to the Eight-Line Asynchronous Communications Multiplexer.

The multiplexer address is established by the installation of jumpers at F17. The jumpers supply ground to the inputs of the buffers. When the jumpers are removed, positive five volts is applied to the inputs by the HDADD lines. The buffer is enabled by the MPC enable read (LMPCENARD) line and the multiplexer address is applied to the memory data lines (HMDAT0 through HMDAT3).

4.11.3 Parity Generator/Checker

The nine-bit odd/even parity generator/checker receives data and odd parity from the MP Bus during the listen cycle of the talker/listener. A parity error is generated if the parity bit is not in the proper state.

During the talk cycle of the talker/listener, data is applied to the parity generator/checker and the odd parity bit is generated and sent to the MP Bus.

4.11.3.1 Parity Check

During the listen cycle of the talker/listener, the gate at C15 is enabled by the transceiver control (HTR1) line being low. The gate at G02 is enabled by the transceiver control (LTR1) line being high. The received data bits (LDATA1 through LDATA8) are applied directly to inputs A through H of the parity generator/checker. The received parity bit (LPARBITIN) signal is applied to the enabled gate at C15, and, through the gate, to the I input of the parity generator/checker.

Table 4-6, Parity Generator/Checker Truth Table, shows the relationship between the number of high inputs and the generated outputs.

Table 4-6
Parity Generator/Checker Truth Table

Number of Inputs A Through I That Are High	Outputs EP	OP
0, 2, 4, 6, 8	H	L
1, 3, 5, 7, 9	L	H

When the state of the parity bit (LPARBITIN) is correct, the odd parity (OP) output of the parity generator/checker will be a logic low. If the state of the parity bit (LPARBITIN) signal is incorrect, the OP output will be a logic high at the input to the enabled gate at G02. The flip-flop at G17 will be set when the MPC data valid (LMPCDAV) line goes inactive, causing an MPC parity error (HMPCPERR) signal to be sent to the interrupt logic.

The parity error line will remain high until cleared by the receipt of the MPC enable write (LMPCENAWR) signal.

4.11.3.2 Parity Generation

The parity generator/checker generates a parity bit when data is being transferred to the MP Bus. During the data transfer, the transceiver control (HTR1) line will be active inhibiting the gate at location C15. Therefore, the I input to the parity generator/checker will always be a logic high during transfers to the MP Bus. The even parity (EP) output of the parity generator/checker will be a logic low with an even number (0, 2, 4, 6, 8) of high data bits (LDATA1 through LDATA8). It will be a logic high with an odd number (1, 3, 5, 7) of high data bits. Thus, the parity bit output (LPARBITOUT) signal will be generated when the number of data bits is even.

4.11.4 Baud Clock Generator

An on board 5.0688 MHz oscillator (Loc. E22) is used on the 160-103768 to generate the baud rate clock. The clock frequency is divided in half to provide a 2.5344 MHz baud clock (HBAUDCLK) frequency.

4.12 MPC Interface (Sheet 8)

The MPC interface, sheet 8, shows the data, parity, and control lines which are connected to the MP Bus. The data, parity, and control signals are applied (bidirectional) to noninverting tranceivers.

The 8291 general purpose interface bus (GPIB) talker/listener is a microprocessor controlled device that handles the communications between the MP Bus and other microprocessor controlled devices. The 8291 performs information transfers, three-wire handshake protocol, talker/listener addressing procedures, device clearing and triggering, service requesting, and serial and parallel polling procedures.

After a power-up reset, controller or device clear, all pointers, counters, and flags are set to starting conditions. ACE interrupts are also disabled and the 8291 (GPIB talker/listener) is reset and loaded with the primary address and set to wait. The primary address is read from hardware jumpers on the board after each power-up or clear sequence. The eight bytes of sense status for each port are zeroed and the executive idle loop of the firmware is entered.

4.12.1 Data Lines (LDI00 through LDI07)

The data in lines (LDI00 through LDI07) are bidirectional and receive data from or transmit data to the MP Bus.

In the receive mode, data currently on the MP Bus is applied to the data lines (LDATA1 through LDATA8) through the enabled tri-state buffer (LS241) at location D14. The buffer is enabled by the external transceiver control (TR1) line at pin 1 of the 8291 GPIB talker/listener being inactive.

In the transmit mode, data currently on the data lines (LDATA1 through LDATA8) is applied to the MP Bus through the enable drivers (75453). The drivers are enabled by the external transceiver control (TR1) line, at pin 1 of the 8291 GPIB talker/listener being active.

4.12.2 Parity

The parity bit (LPARITYODD) received from the MP Bus is applied to the input of the tri-state bus buffer gate to generate the parity bit in (LPARBITIN) signal when the gate is enabled. The gate is enabled by the external transceiver control (TR1) line, at pin 1 of the 8291 GPIB talker/listener being inactive.

The parity bit out (LPARBITOUT) signal, which is generated by the parity generator on sheet 7, is applied to the input of the enabled driver at location A16. The driver is enabled by the external transceiver control (TR1) line, at pin 1 of the 8291 GPIB talker/listener being active.

4.12.3 MP Bus Controls

Not ready for data (LNRFD), no data accepted (LNDAC), and data valid (LDAV) are bidirectional control lines which are functionally similar to the parity receive and transmit control lines.

Interface clear (LIFC) and LATN line are input signals which are applied to the enabled buffers at location A12.

4.12.4 Interface Clear (LIFC)

The MP Bus can issue an LIFC signal to the Eight-Line Asynchronous Communications Multiplexer. The clear signal, which is applied to pin 24 (IFC) of the talker/listener, initializes the talker/listener. The LIFC signal also generates the MPC interface clear signals (LMPCIFC and HMPCIFC) which are used to initialize other components of the Eight-Line Asynchronous Communications Multiplexer.

4.13 GPIB Talker/Listener

The GPIB talker/listener is under Z80 software control. The IOP dictates when the GPIB should be a talker or a listener. Signals are placed onto the GPIB via the bus drivers. Signals coming off the GPIB are sent to the tri-state bus drivers.

4.13.1 Data Bus Port (D0 through D7)

The data bus port (D0 through D7) is a bidirectional port which receives and transmits data on the memory data bus (HMDAT0 through HMDAT7).

4.13.2 GPIB Data Port (DI01 through DI08)

The eight-bit GPIB data port (DI01 through DI08) is used for bidirectional data byte transfer between the talker/listener and the MPC bus. The data port transmits data (LDATA1 through LDATA8) to the MPC bus through the enabled drivers and receives data (LDATA1 through LDATA8) through the enabled buffer at location D14.

4.13.3 Handshake Controls

The three handshake control lines are bidirectional and receive from or transmit to the MP Bus. These signals include DAV, NRFD, and NDAC. All of these signals are active low.

4.13.3.1 Data Valid (DAV)

The DAV signal is transmitted from the MPC to the talker/listener to indicate that valid data is currently on the MP Bus and ready for transmission to the talker/listener. When data is to be transmitted from the talker/listener to the MP Bus the DAV is transmitted to the Bus to indicate that valid data is present on the MP Bus and ready for transmission.

4.13.3.2 Not Ready for Data (NRFD)

The NRFD signal is transmitted to the MP Bus or to the talker/listener to indicate that it is ready (or not ready) to accept the data which is currently on the MP Bus.

4.13.3.3 Not Data Accepted (NDAC)

The NDAC signal is transmitted to the MP Bus or to the talker listener to indicate that it has accepted (or not accepted) the data which is currently on the MP Bus data Bus.

4.13.4 Command Controls

The command controls consist of the ATN, IFC, SRQ, remote enable (REN) and EOI signals.

4.13.4.1 Attention (ATN)

The ATN signal is applied to the talker/listener to indicate the type of information that is present at the GPIB data port (DIO lines). A logic low indicates that a command (control) byte is present on the data bus and a logic high indicates the presence of data.

4.13.4.2 Interface Clear (IFC)

The IFC signal is applied to the talker/listener from the MP Bus. The IFC signal places the talker/listener in an initialized state.

4.13.4.3 Service Request (SRQ)

The SRQ signal is transmitted to the MP Bus to indicate that the talker/listener needs attention and requests an interruption of the current sequence of events on the MP Bus.

4.13.4.4 End or Identify (EOI)

The EOI signal is bidirectional and indicates to the MP Bus or to the talker/listener the end of a multiple byte transfer sequence or, in conjunction with ATN, initiates the parallel polling sequence.

4.13.5 DMA Controls

Controls between the talker/listener and the DMA consist of DMA request (DREQ) and DMA acknowledge (DACK) signals.

4.13.5.1 DMA Request (DREQ)

The DREQ signal is generated in response to MP Bus data being accepted by the talker/listener, and in response to the data output register being empty. The data output register is used to transfer information to the MP Bus when the talker/listener is addressed to talk. The data input register is used to transfer information from the MP Bus to the microprocessor or to memory when the talker/listener is addressed to listen. The DREQ line is reset when the talker/listener receives the DACK.

The DREQ signal generates the MPC to DMA request (LMPCDMAREQ) signal when the gate at D15 is enabled by the LEOIREQ signal.

The DACK signal resets the DREQ.

4.13.5.2 DMA Acknowledge (DACK)

The MPC to LMPCDMAACK signal is applied to the DACK input of the talker/listener. The acknowledge signal is generated by the DMA in response to a DREQ. It sets up the talker/listener such that the RD and WR signals, sent by the DMA, refer to the data in and data out registers of the talker/listener.

4.13.6 Transceiver Controls

The transceiver control signals (TR1 and TR2) control the transmit and receive functions of the bidirectional signals.

4.13.6.1 Transceiver Control 1 (TR1)

The TR1 line is set high to transmit information (LDI00 through LDI07) and the LDAV signal to the MP Bus. It also receives the LNRFD and the LNDAC signals from the MP Bus. The TR1 line is set low to receive information (LDI00 through LDI07) and the LDAV signal from the MP Bus and to transmit the LNRFD and the LNDAC signals to the MP Bus.

4.13.6.2 Transceiver Control 2 (TR2)

The TR2 line is set high to indicate an output signal on the EOI line. The TR2 line is set low to indicate an expected input signal on the EOI line during a parallel poll sequence.

4.13.7 MPU Controls

The RD/WR controls enable the bidirectional communications signals between the MPU and the 8291 GPIB talker/listener. These controls include the register select inputs (RS0 through RS2), the chip select (CS), the RD and WR strobes, and INT lines.

4.13.7.1 Register Select Inputs (RS0 through RS2)

Memory address bits 0, 1, and 2 (HMADD0, 1, and 2) are applied to the register select inputs (RS0, RS1, and RS2). The MPU uses these address bits to select the register (0 through 7), internal to the talker/listener, which will be written into or read from with the execution of a write (LIOWR) or a read (LIORD) instruction.

4.13.7.2 Read Strobe (RD)

The read strobe is driven by the LIORD line and, when low, the contents of the selected register (0 through 7) are read by the MPU.

4.13.7.3 Write Strobe (WR)

The write strobe is driven by the LIOWR line and when low, the MPU writes data into the selected register (0 through 7).

4.13.7.4 Chip Select (CS)

The CS line is driven by the LGPIB select signal and, when low, enables the reading from and the writing into the register (0 through 7) selected by the RS0, RS1, and RS2 lines.

4.13.7.5 Interrupt Request (INT)

The INT signal generates the HMPCINT signal to the interrupt logic. The interrupt is set high for a request and cleared when the appropriate register is accessed by the MPU.

4.13.8 Miscellaneous Controls

4.13.8.1 Clock (CLK)

The clock (HCLK) is applied to pin 3 (CLK). The clock is used only for timing the internal delay generator on the GPIB.

4.13.8.2 Trigger Output (TRIG)

The TRIG signal is not used.

4.13.8.3 Reset

The RESET input signal is not used.

4.14 MPC EOI Controller (Sheet 9)

The MPC EOI controller signal enables and disables the receipt and generation of the EOI signal. In addition, the MPC EOI circuitry provides a means of taking the Eight-Line Asynchronous Communications Multiplexer off line.

4.14.1 Clear

The MPC EOI circuitry is initialized by the LMPCIFC signal which places the flip-flops at E17 in a clear state. The output of pin 5 clears the flip-flop (pin 13) at E18 and disables the gate (pin 1) at G02. The output at pin 8 of the flip-flop at E17 inhibits the generation of the EOI signal.

4.14.2 Off-line Select

Switch S1 provides a means of electrically enabling or disabling the Eight-Line Asynchronous Communications Multiplexer.

4.14.3 Transmit EOI

The Eight-Line Asynchronous Communications Multiplexer can terminate data transfers to the MP Bus by setting the LEOI line to the MP Bus. This action indicates that the last byte has been transferred.

4.15 DMA Controller (Sheet 10)

The DMA controller operates in two major cycles: the idle and the active. The DMA controller enters the idle cycle when there is no valid DMA request pending (the MPC to DMA request, HMPCDMAREQ, is a logic low). While in the idle state, the DMA controller is inactive; however, it may be in the process of being programmed by the MPU. In the active state the DMA controller can perform information transfers between memory and the MPC and between memory locations.

4.15.1 Idle Cycle

If during the idle cycle (HMPCDMAREQ is inactive) the DMA select (LDMASEL) line goes to a logic low, the DMA controller enters a program condition whereby the MPU can establish, change, or inspect the registers internal to the DMA controller. During this operation the MPU places the register address on memory address lines (HMADD00 through HMADD03). The MPU then issues an LIORD or an LIOWR signal to read from or write to the internal register selected by the address line. The MPU either receives data bits or sends data bits, depending on the operation, on the memory data bus (HMDAT00 through HMDAT07).

4.15.2 Active Cycle

At the time the DMA controller receives an MPC to DMA request (HMPCDMAREQ), the MPU receives an interrupt and an interrupt vector address. The MPU then programs the DMA controller to handle the information transfer and, through software control, enables the request line (REQ2) at pin 17. The request line must be held active until the acknowledge becomes active in order to be recognized. The DMA controller then enters the active cycle and processes the request.

The DMA controller acknowledges the MPC request by issuing a MPC to DACK signal (LMPCDMAACK) and issues a LBUSREQ to the MPU. When the MPU relinquishes the bus, a HBUSAK signal is returned to the DMA controller. The BUSAK signal informs the DMA controller that transfers may begin and enables the buffer at location E11.

The single byte transfer mode is established in the DMA controller and will make one transfer. The word count will be decremented and the address decremented or incremented following each transfer. If the MPC to DMA controller request line is held active throughout the single byte transfer, the bus request will go inactive and relinquish the bus to the MPU. The BUSREQ will again go active and upon receipt of a new BUSAK signal, and another single byte transfer will be performed. The BUSREQ/BUSAK signals will toggle in this manner until all information has been transferred.

When the word count of the DMA controller goes to a minus one, the EOP signal at pin 36 will be generated and transfer will be terminated. The EOP signal causes an EOI to the MP Bus. If the MP Bus has more information to be transferred to memory, the sequence is started over.

4.15.3 Write Transfers

Write transfers move information from MPC interface (sheets 7 and 8) to the RAM (sheet 6).

During the program condition, the MPU gives the DMA controller a starting address and a byte count for the transfer. After the MPU has relinquished the bus to the DMA, the DMA controller activates the memory write (MMW) line at pin 4 and the IOR line at pin 1. The MMR signal generates the LWR and the LMREQ signals to memory. The IOR signal generates the LIOREQ signal to interrupt logic and the I/O address decoders, through enabled buffer at E11.

4.15.4 Read Transfers

The read transfer is initiated by the MP Bus and transfers information from the RAM (sheet 6) to the MPC interface. During the program condition the MPU sets up the DMA controller for the read transfer. The MPU supplies a memory starting address, the location in memory from where the read transfer will be started and a word count of the transfer to be completed.

The read transfers are accomplished in a manner similar to the write transfers, with the DMA activating the MMR line at pin 3 and the IOWR line at pin 2.

The MMR signal generates the LRD and the LMREQ signals to memory. The IOWR signal generates the LIOREQ signal to interrupt logic and the address decoder, through the enabled buffer at E11.

4.15.5 Addressing

4.15.5.1 Program Condition

During the program condition the MPU can establish, alter, or inspect the individual registers within the DMA controller. The address of the specific register is placed on the address 0 (A0) through the address 3 (A3) lines at pin 32 through 35.

4.15.5.2 Read/Write Transfers

During the read and write transfers, the DMA controller generates a 16-bit memory address (HMADD00 through HMADD15). The eight lower order memory address bits (HMADD00 through HMADD07) are applied directly to the memory address bus from the DMA controller.

4.15.6 Address Generation

The DMA controller multiplexes the eight higher order address bits on the data lines. The eight higher order address bits are applied to a latch at G09 via the data bus lines (DB0 through DB7). The latch is enabled, output enable (OE), by the address enable (AEN) line at the DMA controller. The DMA controller then generates an address strobe (ADSTB) signal which gates (G) the address bits into the latch. The eight higher order address bits (HMADD08 through HMADD15) are present on the address bus as long as the latch OE input is active.

4.15.7 Clear (Reset)

The DMA controller is reset by the MPC interface clear (HMPCIFC) line going active. The reset at pin 13 clears the DMA controller internal command, status, request and temporary registers. It also clears the first/last flip-flop and the mask register and places the DMA controller in the idle cycle.

4.15.8 Clock

The clock (HCLK) signal times the internal operation of the DMA controller and its rate of information transfers. The HCLK signal is applied to pin 12 (CLK) of the DMA controller.

4.16 Port 0 (Sheets 11 through 14)

The interface between the Eight-Line Asynchronous Communications Multiplexer and the distribution panels is shown on sheets 11 through 14 of the logic diagram (130-103768). Each of the eight ports operate in a similar manner. The following discussion will address only the port 0 interface shown on sheet 11.

The port 0 interface on sheet 11 shows the port controls and the port data lines. The port performs a serial-to-parallel conversion on data characters received from the distribution panel and parallel-to-serial conversion on data characters received from the MPU. The asynchronous communications element (ACE) controls the data communications and the bidirectional data conversions.

After a power-up sequence, all ports are blocked from receiving or transmitting data. The data terminal ready (DTR) and request to send (RTS) signals are set false.

Clear to send (CTS), data set ready (DSR), ring and break signals are ignored. The only interrupts accepted are those from the MPC.

Definition control commands must be sent in order to activate the eight ports. Once a port has received its parameters and baud rate, the DTR line will be set true.

When a write command is sent to an addressed port, the RTS signal will be set true for full- or half-duplex operation. When the DSR and CTS signals are received true, data will proceed to be transmitted. Should either of these signals go false, transmission will be halted and sense status updated. Data will be taken from the port 0 output buffer and outputted through that port. If an EOM signal is received from the MP Bus while the buffer is being filled, then the output will be complete when the buffer empties.

When a read command is sent to an addressed port, while a write is in progress during half-duplex operation, a busy condition will be sent to the MP Bus. If no commands are in progress, data will then be buffered into the input buffer for that port.

Data will be transferred to the MP Bus when a buffer fills or if a terminating character is detected. The type ahead feature allows input to continue and be held until a read is received. Data will proceed to main memory until an EOM signal is received from the MP Bus.

4.16.1 Asynchronous Communications Element

The ACE is a UART type device which performs the function of controlling the data between the distribution panel and the MPU, a parallel-to-serial conversion on data characters received from the MPU, and a serial-to-parallel conversion on data characters received from the distribution panel.

The MPU can read the complete status of the ACE at any time during the functional operation. Status information from the ACE includes the type and condition of the transfer operation being performed as well as any error conditions.

In addition, the ACE includes a programmable baud rate generator for driving transmit and receiver logic.

4.16.2 Input Signals

4.16.2.1 Chip Select (CS2)

The chip select port 0 through 7 (LCSP0 through LCSP7) signal is applied to the CS2 line at pin 14 and enables communications between the ACE and the MPU.

4.16.2.2 Data Input Strobe (DISTR)

The DISTR signal, at pin 21 is driven by the LIORD line. When the LIORD line is active, while the chip is selected, the MPU can read status or data from a selected register.

4.16.2.3 Data Output Strobe (DOSTR)

The DOSTR at pin 18 is driven by the LIOWR line. When the LIOWR is active, while the chip is selected, the MPU can write data or control words into a selected register.

4.16.2.4 Register Select (A0-A2)

The register select lines are driven by the MPU address bits (HMPADD0, HMPADD1, and HMPADD2). These three inputs are used during the read or write operations to select an internal register. The state of the most-significant bit (divisor latch bit DLAB) of the internal line control register affects the selection of certain registers (see Table 4-7, ACE Register Addresses). The DLAB line is set low when the master reset (MR) input is active. The DLAB line must be set high by the system software to access the baud rate generator.

4.16.2.5 External Clock (XTAL1)

The baud clock frequency of 2.5433 MHz is applied to the XTAL1 line. This clock is divided by the specified divisor in the baud generator latches to establish the baud rate for transmitted and received data.

4.16.2.6 Master Reset (MR)

The MR line is driven by the interface clear (HMPCIFC) signal and clears all the internal registers (except the receiver buffer register, the transmitter holding register) and the control logic. The MR signal also sets the serial output (SOUT) line at pin 11 to a logic high, the interrupt (INTRPT) line at pin 30 to a logic low, and the DTR line at pin 33 and RTS line at pin 32 to a logic high.

4.16.2.7 Receiver Clock (RCLK)

The RCLK signal at pin 9 is the baud rate clock for the receiver section of the chip and is driven by the baud rate output (BOUT) line at pin 15.

4.16.2.8 Serial Input (SIN)

The port 0 received data (HPORXDAT) signal, from the distribution panel, is applied to the SIN at pin 10.

4.16.2.9 Clear to Send (CTS)

The CTS line is driven by the port 0 clear to send (HP0CTS) signal from the distribution panel. The CTS signal indicates that the peripheral device connected to the distribution panel at port 0 is in a condition to send data.

4.16.2.10 Data Set Ready (DSR)

The DSR line is driven by the port 0 data set ready (HP0DSR) signal and indicates that the data set connected to the distribution panel port 0 is ready to establish the communications link and transfer data with the ACE.

4.16.2.11 Received Line Signal Detect (RLSD)

The RLSD signal at pin 38 is driven low by the port 0 HP0DCD signal and indicates that the data carrier has been detected by the device connected to port 0 of the distribution panel.

4.16.2.12 Ring Indicator (RI)

The RI line is driven by the port 0 ring (HP0RING) signal and indicates that a telephone ringing signal has been received by the external device connected to port 0 of the distribution panel.

4.16.3 Output Signals

4.16.3.1 Data Terminal Ready (DTR)

The DTR signal at pin 33 generates the port 0 data terminal ready (HP0DTR) signal and informs the external device connect to port 0 of the distribution panel that the ACE is ready to communicate.

Table 4-7
ACE Register Addresses

DLAB	A2	A1	A0	Register AC
0	0	0	0	Receiver buffer (read). Transmitter holding register (write)
0	0	0	1	Interrupt enable
X	0	1	0	Interrupt identification (read only)
X	0	1	1	Line control
X	1	0	0	MODEM control
X	1	0	1	Line status
X	1	1	0	MODEM status
X	1	1	1	None
1	0	0	0	Divisor latch (least-significant byte)
1	0	0	1	Divisor latch (most-significant byte)

4.16.3.2 Baud Out (BAUDOUT)

The BAUDOUT signal is the clock from the ACE transmitter section and is applied to the RCLK input. The clock rate is equal to 2.5344 MHz divided by the specified divisor in the baud generator divisor latches.

4.16.3.3 Interrupt (ITRPT)

The ITRPT signal at pin 30 is used to generate the port 0 interrupt (HP0INT) signal to the interrupt logic. The interrupt is active whenever one of the following interrupt sources is active: receiver error flag; received data available; transmitting holding register empty; and modem status. The interrupt is set low with a MR signal.

4.16.3.4 Serial Output (SOUT)

The SOUT signal at pin 11 transmits serial data (HP0TXDAT) to the external device connected to port 0 of the distribution panel. The SOUT line is set to the marking state (logic high) with a MR signal.

4.16.3.5 Request to Send (RTS)

The RTS at pin 32, when low, informs the external device that the ACE is ready to transmit data. The RTS signal is set high upon a master reset operation.

4.16.4 Input/Output Signals

4.16.4.1 Data Bus (DATA0 through DATA7)

The data bus is comprised of eight three state bidirectional data lines. The bus provides bidirectional communications between the ACE and the MPU. Data, control words, and status information are transferred via the data bus.

4.17 Buffers/Drivers

All of the incoming signals from the distribution panel port 0 are applied to buffers at A01 and A03.

The output signals to the distribution panel port 0 are applied through drivers at C01.

4.18 Distribution Panels

The Eight-Line Asynchronous Communications Multiplexer is equipped with one of two distribution panels which interfaces with external I/O devices to provide the RS-232 or the 20-milliampere current loop protocols. These distribution panels convert the signal levels and pinouts between the Eight-Line Asynchronous Communications Multiplexer and the external I/O device for the appropriate protocol.

4.18.1 RS-232C Distribution Panel, Model 8580

The 140-103053, Schematic, RS-232 Distribution Panel shows the signal level and pinout translation required for the RS-232 protocol. The signals from the Eight-Line Asynchronous Communications Multiplexer to the external I/O device do require a translation of signal level. However the signals from the I/O device are applied to a voltage divider comprised of a 2.2K ohm and a 3.3K ohm resistor. See Appendix A for Signal Definitions.

4.18.2 20-Milliampere Current Loop Distribution Panel, Model 8582

The 130-103442, Logic Diagram, MPC-ELA Current Loop shows the pinout and the logic elements necessary to translate the signal levels to the 20-milliampere current loop protocol. Signal level translation is accomplished by a photo transistor optoisolator.

APPENDIX A

Signal definitions for the RS-232 distribution panel connected to the Model 8512-2 Eight Line Asynchronous Communications Controller.

Pin 1 - Frame Ground
Connected to earth ground through the frame of the system.

Pin 2 - Transmit Data - TXD - Output from the 8-line.
Serial output data from the 8-line.

Pin 3 - Receive Data - RXD - Input into the 8-line.
Serial data input into the 8-line.

Pin 4 - Request to Send - RTS - Output from the 8-line.

When the port is defined as half-duplex (bit 0 of byte 1 in the ACE parameter definition data = 1) or if modem ring (bit 1 of byte 1 in the ACE parameter definition data = 1) is enabled, then RTS is reset at initialization time. If the port is full-duplex without modem ring enabled, then RTS is set at initialization.

In the half-duplex mode, when a write is issued, the 8-line checks clear-to-send for a low. If CTS is not low the 8-line resets RTS to insure that RTS is not holding CTS high. The 8-line will wait for CTS to go low. If CTS is low or when it goes low, then RTS is then set and then CTS is checked again, this time for a high. If CTS is high or when it goes high, then data transmission starts. When data transmission is complete, RTS is reset.

In the full-duplex mode without modem ring enabled, RTS remains set unless reset by software or by a reset (controller, channel or system reset). In the full-duplex mode with modem ring enabled, RTS is reset until a write is issued. When a write is issued, the data-carrier-detect line is checked for a high. If DCD is not high the 8-line waits for it to go high. If DCD is high or when it goes high, then RTS is set and CTS is checked for a high. If CTS is not high the 8-line waits for it to go high. If CTS is high or when it goes high, then data transmission starts. Once RTS is set it will remain set unless reset by software or a reset (controller, channel or system reset). This sequence was set up this way to insure that data is not transmitted before the modem link is complete.

When the read-with-flow-control command (command byte = 0XXX1010) is issued and the most significant bit of the command byte is reset, then RTS is used to implement the hardware flow control portion of the command (this command also uses the TX port to send out the Xon and Xoff characters). If the input buffer reaches the 70% full point (38 bytes remaining), an Xoff character (Hex 13) is sent out on the write subchannel. At the same time the RTS is reset. When the input buffer is nearly empty (8 bytes remaining), an Xon character (Hex 11) is sent out and the RTS line is set.

Pin 5 - Clear-To-Send - CTS - Input to 8-Line

When a write is issued in the half-duplex mode, CTS is checked first. If CTS is high, RTS is reset to insure that it is not holding CTS high. The 8-line will wait for CTS to go low. If CTS is low or when it goes low, RTS is set and CTS is checked again, this time for a high. If CTS is low the 8-line waits for it to go high. When CTS goes high, data transmission starts. Once data transmission starts, CTS must remain high until RTS goes low or a CTS drop error will be reported.

In the full-duplex mode, when a write is issued, CTS is checked for a high before data transmission starts. If CTS is low, data transmission is not started. The 8-line will wait for CTS to go high. If CTS is high or when it goes high, data transmission is started. Once data transmission starts, CTS must remain high as long as the write is active or a CTS drop error will be reported.

When the write-with-input-subchannel-monitor (command byte = 00000101) or the write-with-hardware-flow-control-only (command byte = 00001101) commands are issued, CTS is used to implement the hardware portion of flow control. (The write-with-input-subchannel-monitor command also looks for Xon or Xoff characters from the receive port.) In these modes of operation CTS is used to control the transmission of data on a character by character basis. If CTS is high, data transmission is allowed uninhibited. If CTS goes low, data transmission is suspended until CTS goes high again. CTS drop errors are not reported while either one of the write-with-flow-control commands are active.

Pin 6 - Data Set Ready - DSR - Input to 8-line

DSR must be true when a write or a read is issued or the command will be rejected. An exception to this is when the port is defined as full-duplex with modem ring enabled. Then a write or a read may be issued without DSR being checked first. This is to allow communication with auto-dial modems.

The read subchannel will always interrupt on a delta DSR and report status. When there is no read in progress a negative delta DSR will report attention, channel end, device end and unit check. A positive delta DSR will report attention, channel end and device end. When a read is in progress a delta DSR in either direction will terminate the read and produce channel end, device end and unit check status.

If a write is active and a delta DSR occurs in either direction, the write is terminated and a channel end, device end and unit check status is returned along with whatever status is returned for the read subchannel.

NOTE When DTR is reset, delta DSRs will not cause interrupts. The exception to this is when the read-with-flow-control commands that use DTR for hardware flow control are used. In this case delta DSRs cause interrupts regardless of the condition of DTR.

Pin 7 - Signal Ground - SG

This pin is connected to logic ground through the 8-line board.

Pin 8 - Data Carrier Detect or Received Line Signal Detect - DCD or RLSD - Input into 8-Line

In half-duplex a DCD drop error is reported only if data is received after the drop occurred during the same read.

In full-duplex a DCD drop is an immediate error.

NOTE When a write is issued in full-duplex with modem ring enabled, DCD is checked first. If DCD is not high, the 8-line waits for it to go high. If DCD is high or when it goes high, RTS is set and CTS is checked for a high. If CTS is high, or when it goes high, data transmission starts. This insures that data is not transmitted prematurely before the modem link is completed.

Pin 20 - Data Terminal Ready - DTR- Output from 8-1

When modem ring is enabled, DTR is reset at initialization. If modem ring is disabled, DTR is set at initialization. It is the responsibility of software to control DTR after initialization.

When the read-with-flow-control command is issued with the most significant bit set (command byte = 1XXX1010) or the read-with-hardware-flow-control-only command (command byte = 1XXX1110) is issued, DTR is used to implement the hardware portion of flow control. (The read-with-flow-control command also uses the TX port to send out the Xon and Xoff characters.) If the input buffer reaches the 70% full point (38 bytes remaining), an Xoff character (Hex 13) is sent out on the write subchannel. At the same time the DTR line is reset. When the input buffer is nearly empty (8 bytes remaining), an Xon character (Hex 11) is sent out and the DTR line is set.

Pin 22 - Modem Ring - MR - Input into the 8-Line

When modem ring is enabled and there is not a read in progress, a positive pulse on this line will cause an interrupt with attention, channel end and device end status to be returned.

INDEX