MOSTEK®

MD SERIES™ MICROCOMPUTER MODULES

Operations Manual

EPROM/UART MODULE MDX-EPROM/UART

MDX-EPROM/UART

OPERATION MANUAL

Publication Number MK79604

TABLE OF CONTENTS

TABLE OF CONTENTS (contd.)

LIST OF FIGURES

LIST OF TABLES

SECTION 1
GENERAL INFORMATION

1.1 INTRODUCTION

1.1.1 The MD series and the STD BUS were designed to satisfy the need for low cost OEM Microcomputer modules. The STD BUS uses a mother board interconnect system concept and is designed to handle any MD series card type in any slot. The modules for the STD BUS are a compact 4.5 x 6.5 inches which provide for system partitioning by function (RAM,EPROM,I/O). This smaller module size makes system packaging easier while increasing MOS-LSI densities provide high functionality per module.

1.1.2 The MD series of OEM microcomputer boards and the STD BUS offer the most cost-effective system configuration available to the OEM system designer.

1.2 GENERAL DESCRIPTION

1.2.1 The MDX-EPROM/UART is one of MOSTEK's complete line of STD BUS compatible Z8Ø microcomputer modules.

1.2.2 Designed as a universal EPROM add-on module for the STD BUS, the MDX-EPROM/UART provides the system designer with sockets to contain up to 1ØK x 8 of EPROM memory (5-2716's) as shown in the Block Diagram, figure 2-1.

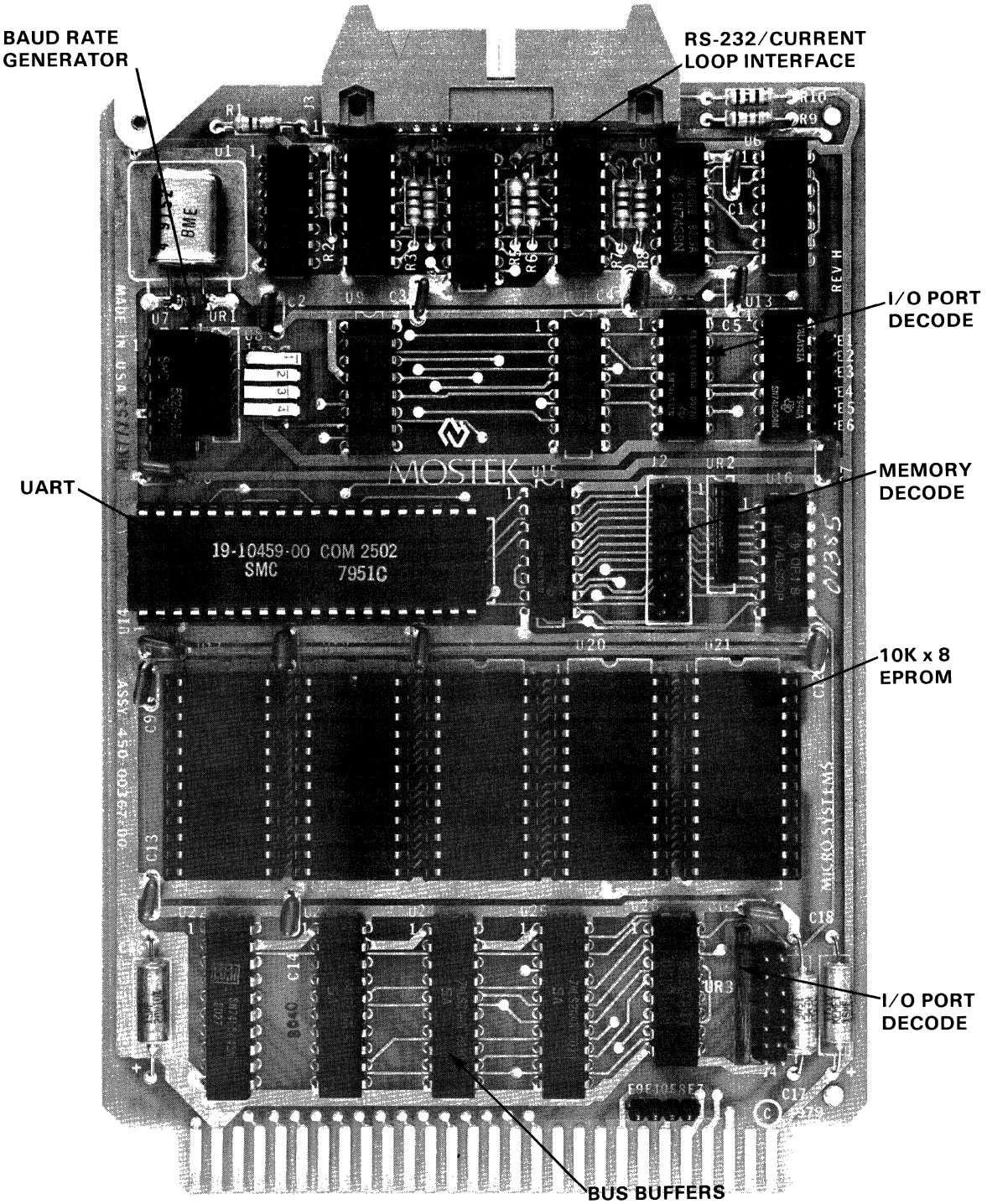
1.2.3 The EPROM memories can be positioned to start on any 2K boundary within a 16K block of memory via a strapping option provided on the MDX-EPROM/UART.

1.2.4 Included on-board the MDX-EPROM/UART is a fully buffered asynchronous I/O port with a Teletype reader step control. A full duplex UART is used to receive and transmit data at the serial port. Operation and UART options are under software control. Once the unit has been programmed, no further changes are

necessary unless there is a modification of the serial data format. Features of the UART include:

Full duplex operation
Start bit verification
Data word size variable from 5 to 8 bits
One or two stop bits may be selected
Odd, even, or no parity option
One word buffering on both transmit and receive

FIGURE 1-1 BOARD PHOTO WITH OVERLAY



1.3 STD-Z80 BUS PIN-OUT AND DESCRIPTION

BUS PIN	MNEMONIC	DESCRIPTION
1	+5V	+5Vdc system power
2	+5V	+5Vdc system power
3	GND	Ground - System signal ground and dc return
4	GND	Ground - System signal ground and dc return
5 *	-5V	-5Vdc system power
6 *	-5V	-5Vdc system power
7	D3	
8	D7	
9	D2	Data Bus (Tri-state, input/output active high). DØ-D7
10	D6	constitute an 8-bit bidirectional data bus. The data bus
11	D1	is used for data exchange with memory and I/O devices.
12	D5	
13	DØ	
14	D4	
15	D7	
16	A15	
17	A6	
18	A14	
19	A5	
20	A13	Address Bus (Tri-state, output, active high). $A_Ø$-A_{15}
21	A4	make up a 16-bit address bus. The address bus provides
22	A12	the address for memory (up to 65K bytes) data exchanges
23	A3	and for I/O device data exchanges. I/O addressing uses

24 25 26 27 28 29 30	A11 A2 A10 A1 A9 A0 A8	the lower 8 address bits to allow the user to directly select up to 256 input or 256 output ports. A_0 is the least significant address bit. During refresh time, the lower 7 bits contain a valid refresh address for dynamic memories in the system.
31	$\overline{WR}$	Memory Write (Tri-state, output, active low). $\overline{WR}$ indicates that the CPU data bus holds valid data to be stored in the address memory or I/O device.
32	$\overline{RD}$	Memory Read (Tri-state, output, active low). $\overline{RD}$ indicates that the CPU 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 CPU data bus.
33	$\overline{IORQ}$	Input/Output Request (Tri-state, output, active low). The IORQ signal indicates that the lower half of the address bus holds a valid I/O address for an I/O read or write operation. An $\overline{IORQ}$ signal is also generated with an $\overline{M1}$ signal when an interrupt is being acknowledged to indicate 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.
34	$\overline{MEMRQ}$	Memory Request (Tri-State, output, active low). The $\overline{MEMRQ}$ signal indicates that the address bus holds a valid address for a memory read or memory write operation.

35 *	$\overline{\text{IOEXP}}$	I/O expansion, not used on Mostek MDX cards.
36 *	$\overline{\text{MEMEX}}$	Memory expansion, not used on Mostek MDX cards
37 *	$\overline{\text{REFRESH}}$	REFRESH (Tri-state, output, active low). $\overline{\text{REFRESH}}$ indicates that the lower 7 bits of the address bus contain a refresh address for dynamic memories and the $\overline{\text{MEMRQ}}$ signal should be used to perform a refresh cycle for all dynamic RAMs in the system. During the refresh cycle A7 is a logic zero and the upper 8 bits of the address bus contains the I register.
38 *	$\overline{\text{DEBUG}}$	DEBUG (Input) used in conjunction with DDT-8Ø operating system and the MDX Single Step card for implementing a hardware single step. When pulled low, the $\overline{\text{DEBUG}}$ line will set a latch that will force the upper three address lines to a logic 1. To reset this latch, an I/O operation must be performed.
39 *	$\overline{\text{M1}}$	Machine Cycle One (Tri-state, output, active low), $\overline{\text{M1}}$ indicates that the current machine cycle is in the op code fetch cycle of an instruction. Note that during the execution of 2-byte op-codes, $\overline{\text{M1}}$ will be generated as each op code is fetched. These two-byte op codes always begin with a CBH, DDH, EDH, or FDH. $\overline{\text{M1}}$ also occurs with IORQ to indicate an interrupt acknowledge cycle.
4Ø	Status Ø	Not used on Mostek MDX cards.
41	$\overline{\text{BUSAK}}$	Bus Acknowledge (Output, active low). Bus acknowledge is used to indicate to the requesting device that the CPU address bus, data bus, and control bus signals have been set to their high impedance state and the external device can now control the bus.

42 *	$\overline{BUSRQ}$	Bus Request (Input, active low). The $\overline{BUSRQ}$ signal is used to request the CPU address bus, data bus, and control signal bus to go to a high impedance state so that other devices can control those buses. When $\overline{BUSRQ}$ is activated, the CPU will set these buses to a high impedance state as soon as the current CPU machine cycle is terminated, and the Bus Acknowledge ($\overline{BUSAK}$) signal is activated.
43 *	$\overline{INTAK}$	Interrupt Acknowledge (Tri-state output, active low). The $\overline{INTAK}$ signal indicates that an interrupt acknowledge cycle is in progress, and the interrupt device should place its response vector on the data bus.
44 *	$\overline{INTRQ}$	Interrupt Request (Input, active low). The Interrupt Request Signal is generated by I/O devices. A request will be honored at the end of the current instruction if the internal software controlled interrupt enable flip-flop (IFF) is enabled and if the $\overline{BUSRQ}$ signal is not active. When the CPU accepts the interrupt, an acknowledge signal ($\overline{IORQ}$ during an $\overline{M1}$) is sent out at the beginning of the next instruction cycle.
45	$\overline{WAITRQ}$	WAIT REQUEST (Input, active low). Wait request indicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter wait states for as long as this signal is active. This signal allows memory or I/O devices of any speed to be synchronized to the CPU.

46 *	NMIRQ	NonMaskable Interrupt Request (Input, negative edge triggered). The nonmaskable interrrupt request line has a higher priority than $\overline{\text{INTRQ}}$ and is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop. $\overline{\text{NMIRQ}}$ automatically forces the CPU to restart to location 0066_H. The program counter is automatically saved in the external stack so that the user can return to the program that was interrupted. Note that continuous WAIT cycles can prevent the current instruction from ending, and that a $\overline{\text{BUSRQ}}$ will override a $\overline{\text{NMIRQ}}$.
47	$\overline{\text{SYSRESET}}$	System Reset (Output, active low). The System Reset line indicates that a reset has been generated from either an external reset or the power on reset circuit. The system reset will occur only once per reset request and will be approximately 2 microseconds in duration. A system reset will also force the CPU program counter to zero, disable interrupts, set the I register to 00_H, set the R register to 00_H, and set Interrupt Mode 0.
48 *	$\overline{\text{PBRESET}}$	Push Button Reset (Input, active low). The push button reset will generate a debounced system reset.
49	$\overline{\text{CLOCK}}$	Processor Clock (Output, active low). Single phase system clock.
50 *	$\overline{\text{CNTRL}}$	Auxiliary Timing
51	PCO	Priority Chain Output (Output, active high). This signal is used to form a priority interrupt daisy

		chain when more than one interrupt driven device is being used. A high level on this pin indicates that no other devices of higher priority are being serviced by a CPU interrupt service routine.
52	PCI	Priority Chain In (Input, active high). This signal is used to form a priority interrupt daisy chain when more than one interrupt driven device is being used. A high level on this pin indicates that no other devices of higher priority are being serviced by a CPU interrupt service routine.
53 *	AUX GND	Auxiliary Ground (Bussed)
54 *	AUX GND	Auxiliary Ground (Bussed)
55	+12V	+12Vdc system power
56	-12V	-12Vdc system power

NOTES:

(1) The input/output designations of the STD-Z80 Bus are made with respect to the MDX-CPU card.

(2) The following signals have pull-up resistors: $\overline{WR}$, $\overline{RD}$, $\overline{IROQ}$, $\overline{MEMRQ}$, $\overline{REFRESH}$, $\overline{DEBUG}$, $\overline{M1}$, $\overline{BUSRQ}$, $\overline{INTAK}$, $\overline{INTRQ}$, $\overline{WAITRQ}$, $\overline{NMIRQ}$, $\overline{SYSRESET}$, $\overline{PBRESET}$ and $\overline{CLOCK}$.

(3) Pins indicated by * are not connected on the MDX-EPROM/UART.

SECTION 2

FUNCTIONAL DESCRIPTION

2.1 INTRODUCTION

2.1.1 The MDX-EPROM/UART shown in Figure 1-1 provides a low cost way to expand EPROM and SERIAL I/O for the MDX system. The major functions of the MDX-EPROM/UART are shown in Figure 2-1 and will be explained below.

2.2 MEMORY ARRAY

2.2.1 The memory array consists of up to 5 MK2716's (2K x 8, +5V only EPROM). The total storage of the MDX-EPROM/UART board is 10,240 bytes.

2.3 MEMORY SELECTION LOGIC

2.3.1 The memory selection logic is responsible for decoding the selected address and enabling the selected EPROM onto the data bus through the data bus buffer.

2.4 PORT SELECTION LOGIC

2.4.1 The port selection logic is used to allow read and write operations to the Serial I/O port (UART).

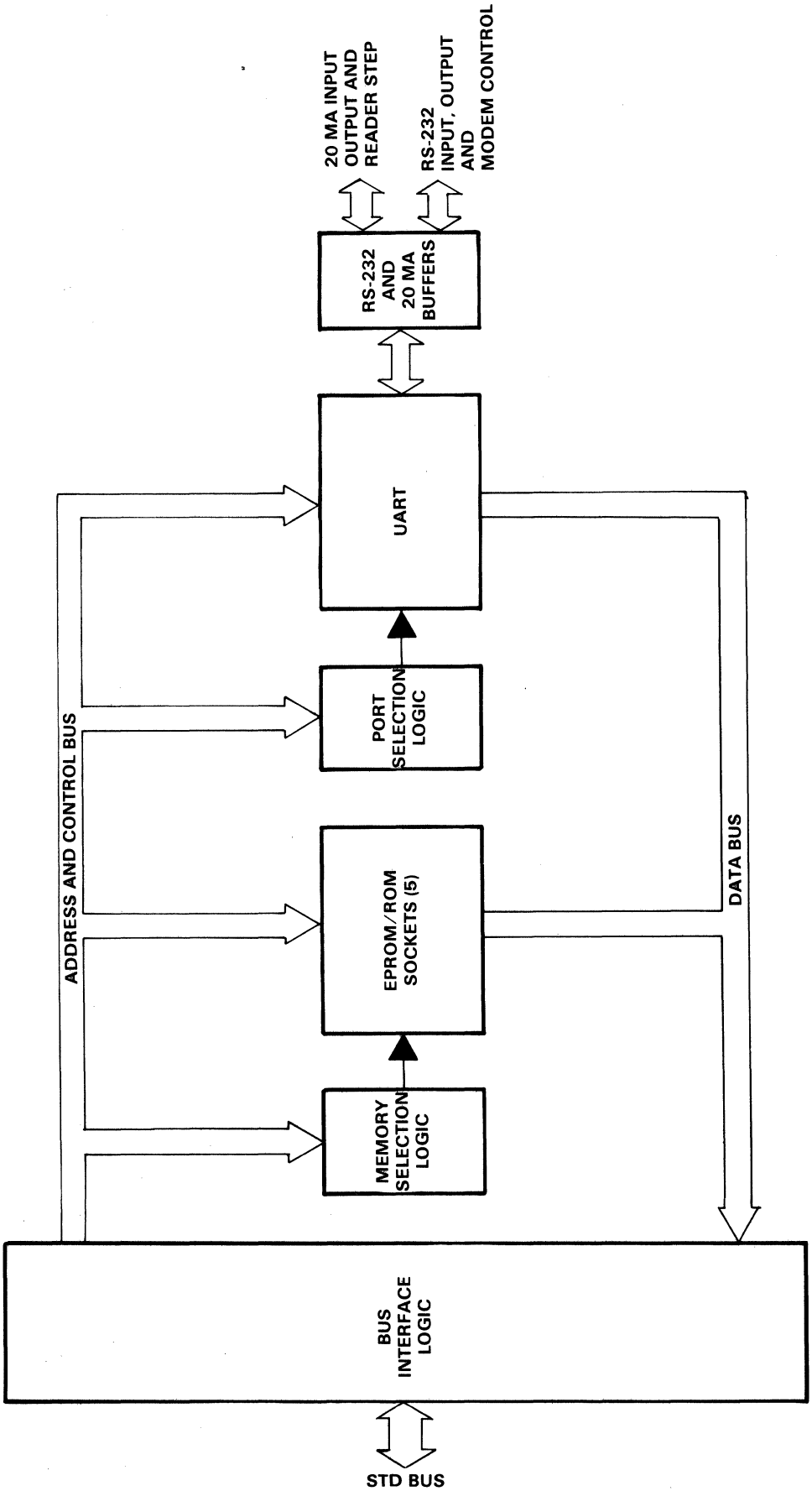
2.5 UART

2.5.1 The UART on the MDX-EPROM/UART board is used to convert parallel data from the CPU into serial data for use with a serial terminal. The UART is also used to convert serial data from the data terminal into parallel data for the CPU.

2.6 RS-232 AND 20mA BUFFERS

2.6.1 The RS-232 and 20mA buffers are used to convert TTL voltage levels used by the MDX logic to the RS-232 voltage levels and 20mA current loop for use with data communications equipment.

FIGURE 2-1 MDX-EPROM/UART BLOCK DIAGRAM



SECTION 3

UTILIZATION

3.1 INTRODUCTION

3.1.1 This section will describe the various jumper options for the MDX-EPROM/UART board.

3.2 MEMORY INTERFACE

3.3 EPROM DECODING JUMPERS

3.3.1 The MDX-EPROM/UART can be populated with up to 10K x 8 of EPROM. (5-2716's). The decoding for the EPROM's is on 2K boundaries within a 16K block. Jumper options for the EPROM decoding is shown in Table 3-1.

3.4 SERIAL INTERFACE

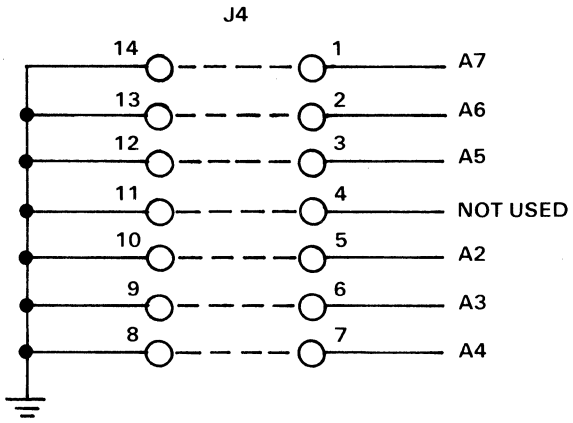
3.5 I/O PORT DECODING JUMPERS

3.5.1 The port decoding for the EPROM/UART board is jumper programmable to allow multiple EPROM/UART boards within a MDX system. The EPROM/UART board uses three read ports and three write ports to interface to the serial port. Figure 3.1 shows the format for strapping the I/O decoder.

FIGURE 3-1 I/O CHANNELS

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	I/O READ	I/O WRITE
X	X	X	X	X	X	0	0	Data from UART	Data to UART
X	X	X	X	X	X	0	1	Status from UART	Control word to UART
X	X	X	X	X	X	1	0	Read MODEM Lines	Write to MODEM LINES

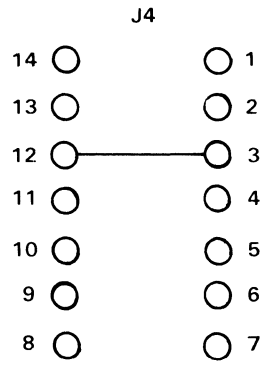
where X represents the I/O code selected on J4



STRAP = Logic Ø No STRAP = Logic 1

EXAMPLE:

STRAP J4 so that the EPROM/UART serial ports will respond to I/O ports DCH, DDH, and DEH:



This strapping will cause the I/O decoder to respond in this way.

A_7	A_6	A_5	A_4	A_3	A_2	A_1	$A_Ø$	READ/WRITE PORTS
1	1	Ø	1	1	1	Ø	Ø	DCH
1	1	Ø	1	1	1	Ø	1	DDH
1	1	Ø	1	1	1	1	Ø	DEH

Selected by J4

The strapping of J4 shown in this example is required for use with DDT-8Ø or the MDX-DEBUG board.

3.6 BAUD RATE SELECTION

3.6.1 The baud rate for the serial interface is generated by the baud rate chip U7. The baud rate is selected by DIP switch U8. Table 3.2 shows DIP switch setting versus baud rates.

3.7 PROGRAMMING THE UART

3.7.1 A full duplex UART is used to receive and transmit data at the serial port. Operation and UART options are under software control. Once the unit has been programmed, no further changes are necessary unless there is a modification of the serial format. Transmit and receive clock rates (baud clock rate) must be 16 times the desired baud rate. A programming model for the UART is shown in Figure 3-2.

3.8 SERIAL I/O CONNECTOR AND CABLE

3.8.1 All serial interface lines are brought out to a 26-pin connector J3.

3.8.2 The serial cable is constructed of the following parts:

1) 26-pin connector (Ansley No. 609-2600M)
2) 26 wire flat cable (Ansley No. 171-26)
3) 25-pin standard EIA-RS232 connector (Ansley No. 609-25P)

3.8.3 Table 3-3 shows the interconnection between the 26-pin connector and the 25-pin RS-232 connector.

3.9 RS-232 INTERFACE

3.9.1 Because the MDX-EPROM/UART was designed to communicate with RS-232 terminals (as opposed to other types of communication peripherals), the serial interface looks like a receiving modem or computer port rather than a transmitting

terminal port (such as a Silent 700). The effect of this is to scramble three pairs of signals.

3.9.2 For example:

1) Transmitted Data (RS-232) from Terminal (Pin 3) is an out direction signal at the terminal but is shown as an in-direction signal at the serial port.

   Receive Data (RS-232) at Terminal (Pin 5) is an in-direction signal at the terminal but is shown as an out-direction signal at the serial port.

2) Request To Send (Pin 7) is an out-direction signal at the terminal but is shown as an in-direction signal at the serial port.

   Clear To Send (Pin 9) is an in-direction signal at the terminal but is shown as an out-direction signal at the serial port.

3) Data Terminal Ready (Pin 14) is an out-direction signal at the terminal but is shown as an in-direction signal at the serial port.

   Data Set Ready (Pin 11) is an in-direction signal at the terminal but it is shown as an out-direction at the serial port.

3.9.3 To change the "sense" of this port i.e., to make it look like a transmitting terminal (as might be required in some OEM applications), the two signals in each pair above need to be interchanged.

3.10 TELETYPE AND READER STEP INTERFACE

3.10.1 Figure 3-3 shows how the MDX-EPROM/UART can be interfaced to an ASR-33 Teletype with and without reader step. The reader step function is controlled by lines RS+ and RS-. These lines control an optically isolated solid state relay

which controls the 115VAC teletype reader. The MDX-DEBUG board is designed to interface directly to the MOSTEK TTY cable with reader step control. The Reader Step lines can control up to 40mA of current.

NOTES:

Under no circumstances should 110VAC ever be applied to the MDX-EPROM/UART or MDX-DEBUG cards.

TABLE 3-1

MEMORY DECODING JUMPER SELECTION

	J2 JUMPERS					JUMPERS E1 - E6
DECODED ADDRESS	EPROM U17	EPROM U18	EPROM U19	EPROM U20	EPROM U21	
	J2	J2	J2	J2	J2	
0000-07FF	Pin 16 to Pin 1	Pin 15 to Pin 1	Pin 14	Pin 13 to Pin 1	Pin 12 to Pin 1	
0800-0FFF	" 2	" 2	" 2	" 2	" 2	E1 to E6
1000-17FF	" 3	" 3	" 3	" 3	" 3	and
1800-1FFF	" 4	" 4	" 4	" 4	" 4	E4 to E5
2000-27FF	" 5	" 5	" 5	" 5	" 5	
2800-2FFF	" 6	" 6	" 6	" 6	" 6	
3000-37FF	" 7	" 7	" 7	" 7	" 7	
3800-3FFF	" 8	" 8	" 8	" 8	" 8	
4000-47FF	" 1	" 1	" 1	" 1	" 1	
4800-4FFF	" 2	" 2	" 2	" 2	" 2	
5000-57FF	" 3	" 3	" 3	" 3	" 3	E1 to E6
5800-5FFF	" 4	" 4	" 4	" 4	" 4	and
6000-67FF	" 5	" 5	" 5	" 5	" 5	E3 to E5
6800-6FFF	" 6	" 6	" 6	" 6	" 6	
7000-77FF	" 7	" 7	" 7	" 7	" 7	
7800-7FFF	" 8	" 8	" 8	" 8	" 8	
8000-87FF	" 1	" 1	" 1	" 1	" 1	
8800-8FFF	" 2	" 2	" 2	" 2	" 2	
9000-97FF	" 3	" 3	" 3	" 3	" 3	E2 to E6
9800-9FFF	" 4	" 4	" 4	" 4	" 4	and
A000-A7FF	" 5	" 5	" 5	" 5	" 5	E4 to E5
A800-AFFF	" 6	" 6	" 6	" 6	" 6	
B000-B7FF	" 7	" 7	" 7	" 7	" 7	
B800-BFFF	" 8	" 8	" 8	" 8	" 8	
C000-C7FF	" 1	" 1	" 1	" 1	" 1	
C800-CFFF	" 2	" 2	" 2	" 2	" 2	
D000-D7FF	" 3	" 3	" 3	" 3	" 3	E2 to E6
D800-DFFF	" 4	" 4	" 4	" 4	" 4	and
E000-E7FF	" 5	" 5	" 5	" 5	" 5	E3 to E5
E800-EFFF	" 6	" 6	" 6	" 6	" 6	
F000-F7FF	" 7	" 7	" 7	" 7	" 7	
E800-FFFF	" 8	" 8	" 8	" 8	" 8	

NOTE: The MDX-DEBUG board is a specialized version of the MDX-EPROM/UART board. The difference between the boards is the installation of the ASMB-80/DDT-80 ROM firmware package. The MDX-DEBUG board will come from the factory with J2 jumpered for addresses C000-E7FF. However, the ROMs may be removed and EPROMs inserted for customer applications. In that case the table shown above should be used to select the appropriate jumper options.

TABLE 3-2

BAUD RATE SWITCH SELECTION

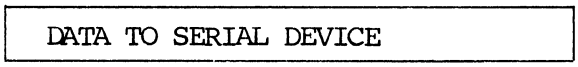
U8 SWITCH POSITION 4	3	2	1	X16 CLOCK	BAUD RATE
Ø	Ø	Ø	Ø	.8KHZ	5Ø
Ø	Ø	Ø	1	1.2	75
Ø	Ø	1	Ø	1.76	11Ø
Ø	Ø	1	1	2.152	134.5
Ø	1	Ø	Ø	2.4	15Ø
Ø	1	Ø	1	4.8	3ØØ
Ø	1	1	Ø	9.6	6ØØ
Ø	1	1	1	19.2	12ØØ
1	Ø	Ø	Ø	28.8	18ØØ
1	Ø	Ø	1	32.Ø	2ØØØ
1	Ø	1	Ø	38.4	24ØØ
1	Ø	1	1	57.6	36ØØ
1	1	Ø	Ø	76.8	48ØØ
1	1	Ø	1	115.2	72ØØ
1	1	1	Ø	153.6	96ØØ
1	1	1	1	3Ø7.2	19,2ØØ

1 = OPEN Ø = CLOSED

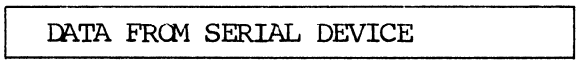
FIGURE 3-2 PROGRAMMING THE UART

1. UART DATA PORT $\underline{DC_H}$

   Write to Port DC_H

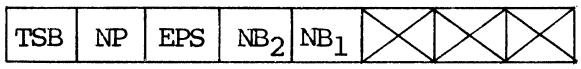


   Read from UART

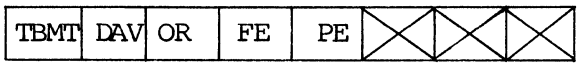


2. UART CONTROL PORT DD_H

   Write to UART

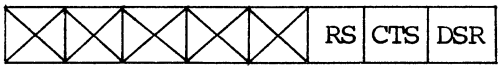


   Read from UART

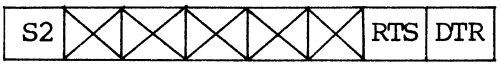


3. System Control Port $\underline{DE_H}$

   Write to Port DE_H



   Read from Port DE_H



Number Stop Bits (TSB) — This bit will select the number of stop bits, 1 or 2, to be appended immediately after the parity bit. A logic "Ø" will insert 1 stop bit and a logic "1" will insert 2 stop bits.

No Parity (NP) — A logic "1" on this lead will eliminate the parity bit from the transmitted and received character (no PE indication). The stop bit(s) will immediately follow the last data bit.

Odd/Even Parity Select (EPS) — The logic level on this pin selects the type of parity which will be appended immediately after the data bits. It also determines the parity that will be checked by the receiver. A logic "Ø" will insert odd parity and a logic "1" will insert even parity.

FIGURE 3-2 (cont)

Number of Bits/Character (NB2, NB1)

These two bits will be internally decoded to select either 5,6,7 or 8 data bits/character.

NB2	NB1	Bits/Character
0	0	5
0	1	6
1	0	7
1	1	8

Transmitter Buffer Empty (TBMT)

The transmitter buffer empty flag goes to a logic "1" when the data buffer holding register may be loaded with another character.

Data Available (DAV)

This bit goes to a logic "1" when an entire character has been received and transferred to the receiver holding register.

Over-Run (OR)

This bit goes to a logic "1" if the previously received character is not read (DAV line not reset) before the present character is transfered to the receiver holding register.

Framing Error (FE)

This bit goes to a logic "1" if the received character has no valid stop bit.

Parity Error (PE)

This bit goes to a logic "1" if the received character parity does not agree with the selected parity.

Reader Step (RS)

A logic 1 on this bit will activate the reader step current loop driver.

FIGURE 3-2 (cont)

Clear to Send (CTS)	A logic 1 on this bit will set the CTS output to a +V RS-232 level.
Data Set Ready (DSR)	A logic 1 on this bit will set the the DSR output to a +V RS-232 level
Request to Send (RTS)	This bit goes to a logic 1 when the RS-232 RTS is in its active state.
Data Terminal Ready (DTR)	This bit goes to a logical 1 when the RS-232 DTR is in its active state.
Serial IN (SI)	This port line inputs the serial data stream from the EIA or teletype terminal that is required by DDT-80.

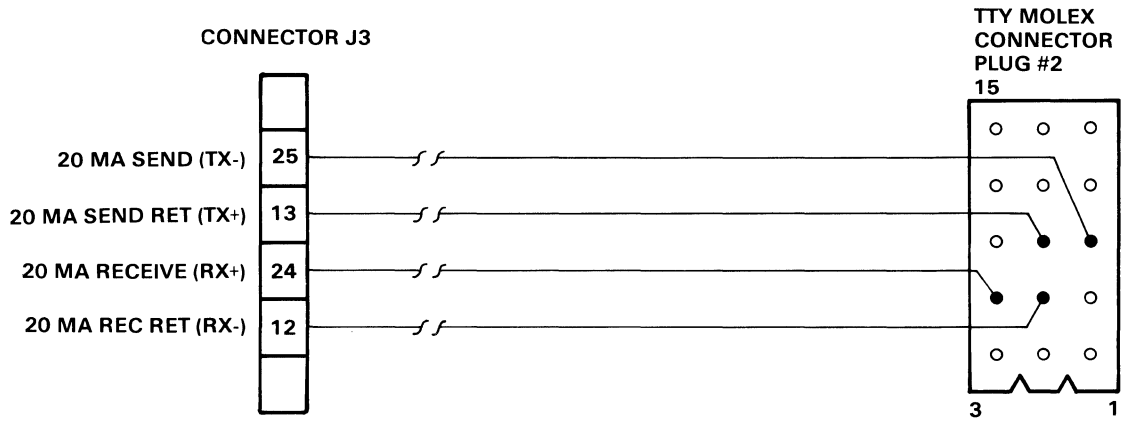
NOTE: DDT-80 was written to automatically calculate and generate the baud rate for the UART using a CTC. However, the MDX-EPROM/UART and MDX-DEBUG cards do not use this feature. The baud rate for the MDX-EPROM/UART and MDX-DEBUG cards is generated by a switch selectable baud rate generator.

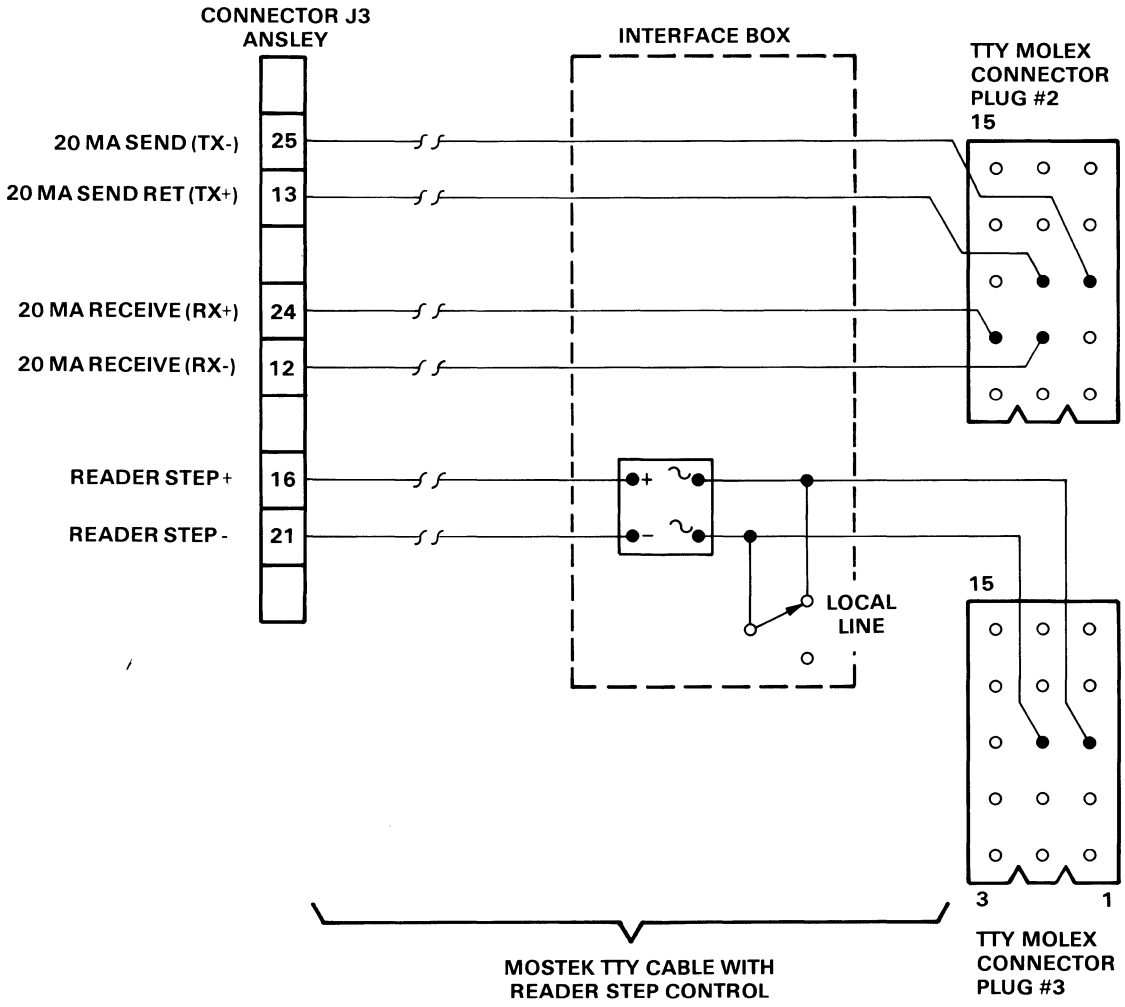
TABLE 3-3

SERIAL PORT TO RS-232 CONNECTOR

SIGNAL NAME	J3 CONNECTOR Pin Number	RS-232 CONNECTOR Pin Number
Chassis GND	1	1
	14	14
Transmitted data	2	2
(RS-232) from terminal	15	15
Receive data (RS-232)	3	3
at terminal		
Reader step +	16	16
Request to send	4	4
	17	17
Clear to send	5	5
	18	18
Data set ready	6	6
	19	19
GND	7	7
Data terminal ready	20	20
Carrier detect	8	8
Reader step -	21	21
	9	9
	22	22
	10	10
	23	23
	11	11
20mA Receive (RX+)	24	24
20mA Receive RET(RX-)	12	12
20mA Send (TX-)	25	25
20mA Send RET (TX+)	13	13
	26	Wire remove from 26 wire flat cable

FIGURE 3-3 TTY CONNECTION WITHOUT READER STEP





APPENDIX A

SPECIFICATIONS

SPECIFICATIONS

A.1 ELECTRICAL SPECIFICATIONS

WORD SIZE

8 bits for PROM

5 to 8 bits for Serial I/O.

MEMORY ADDRESSING

ROM/EPROM

2K blocks jumper selectable for any 2K boundary within a given 16K boundary of Z80 memory map.

MEMORY CAPACITY

10K bytes of 2716 memory.

(2716's not included)

MEMORY SPEED REQUIRED

MEMORY	ACCESS TIME
2716*	450ns

*Single 5 Volt type required

I/O TRANSFER RATE

X16 BAUD RATE CLOCK	BAUD RATE (Hz)
1760	110
4800	300
9600	600
19200	1200
38400	2400
76800	4800
153600	9600
307200	19200

SERIAL COMMUNICATIONS CHARACTERISTICS

Asynchronous

- Full duplex operation
- Start bit verification
- Data word size variable from 5 to 8 bits
- One or two stop bits
- Odd, even, or no parity
- One word buffering on both transmit and receive.

I/O ADDRESSING

On-board fully programmable

SYSTEM CLOCK

	MIN.	MAX.
MDX-EPROM/UART	250 KHz	2.5 MHz
MDX-EPROM/UART-4	250 KHz	4.0 MHz

POWER SUPPLY REQUIREMENTS

+12 Volts $\pm$ 5% at 50 mA max.

-12 Volts $\pm$ 5% at 35 mA max.

+5 Volts $\pm$ 5% at 1.2 A max.

OPERATING TEMPERATURE RANGE

0^{o} to $+60^{o}C$

A.2 MECHANICAL SPECIFICATIONS

CARD DIMENSIONS

4.5 in. (11.43cm) high by 6.50 in. (16.51 cm) long

0.48 in. (1.22 cm) maximum profile thickness

0.062 in. (0.16 cm) printed circuit board thickness

CONNECTORS

FUNCTION	CONFIGURATION	MATING CONNECTOR
STD BUS	56 pin dual Ø.125 in. centers	Printed Circuit Viking 3VH28/1CE5 Wire Wrap 3VH28/1CND5 Solder Lug Viking 3VH28/1CN5
Serial I/O	26 pin dual Ø.1ØØ in. grid	Flat Ribbon Ansley 6Ø9-26ØØM Discrete Wires Winchester PGB26A (housing) Winchester 1ØØ- 7ØØ2ØS (contacts)

A.3 STD-Z8Ø ELECTRICAL BUS SPECIFICATIONS

Bus Receivers
One 74LS load max.

Bus Drivers
Logical Low: Ø.5V max at 24mA
Logical High: 2.4V min at -2.6mA

Recommended Bus Drivers and Receivers

Bus Drivers - 74LS24Ø, 74LSL241, 74LS373, 74LS374, 74LS244

Bus Receivers - 74LS24Ø, 74LS241, 74LS244

Bus Transceivers - 74LS245, 74LS242, 74LS243

MOSTEK®

1215 W. Crosby Rd. • Carrollton, Texas 75006 • 214/323-6000
In Europe, Contact: MOSTEK Brussels
150 Chaussee de la Hulpe, B1170, Belgium;
Telephone: 660.69.24

PRINTED IN USA February 1981
Publication No. MK 79604