MOSTEK®

MD SERIES MICROCOMPUTER MODULES

Operations Manual

Z80 MICROCOMPUTER DEBUG MODULE MDX-DEBUG

Operation Manual
For
MDX - DEBUG
(MK 79611)

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT.)

LIST OF FIGURES

LIST OF TABLES

SECTION I

GENERAL INFORMATION

1-1. INTRODUCTION

1-2. The MD Series and the STD BUS were designed to satisfy the need for low cost OEM microcomputer modules. The STD BUS uses a motherboard 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 provides 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.

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

1-3. HARDWARE DESCRIPTION

1-4. The MDX-DEBUG Module has sockets for 10K bytes of masked ROM and are filled with Z80 firmware package (DDT-80/ASMB-80). This module has a STD BUS interface and is availabe in both 2.5MHz and 4.0MHz versions. Included on-board is a fully buffered asynchronous I/O port capable of 110-19,200 Baud rates. Serial Data interfaces are available for 20mA current loop (with reader step control) and RS-232. The on-board Baud Rate Generator is selectable to all common Baud rates from 110 to 19,200 Baud.

1-5. FIRMWARE DESCRIPTION

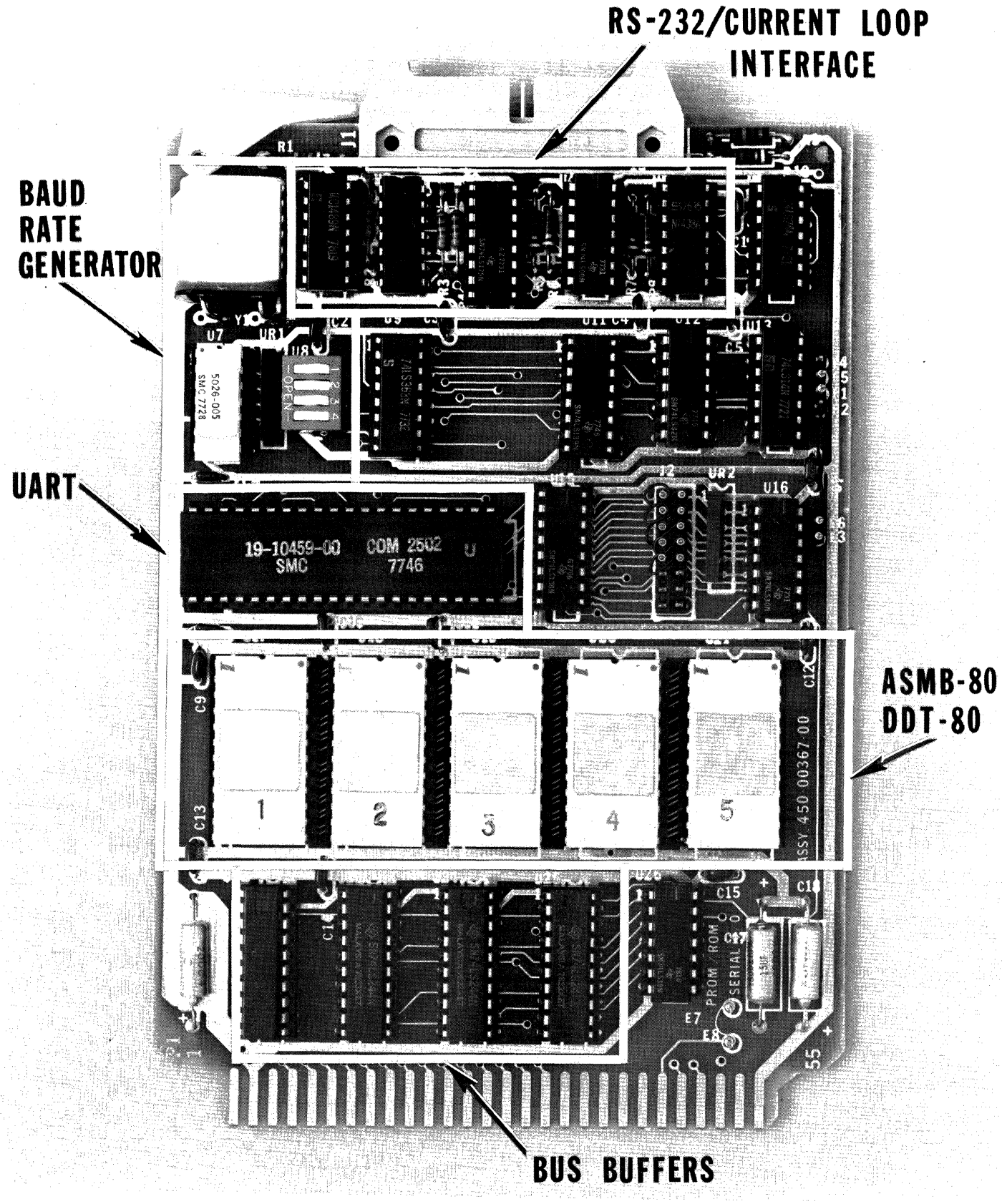
*NOTE: For detailed explanations of DDT-80 and ASMB-80 refer to the DDT-80 and ASMB-80 Operations Manual.

1-6. DEBUGGER DESCRIPTION

DDT-80 is the Operating System for the MDX-DEBUG Module. It resides in a 2K ROM (MK34000 series) resident on the MDX-DEBUG Module. It provides the necessary tools and techniques to operate the system, i.e., to efficiently and conveniently perform the tasks necessary to develop microcomputer software. DDT-80 is designed to support the user from initial design through production testing. It allows the user to display and update memory, registers, and ports, load and dump object files, set breakpoints, copy blocks of memory, and execute programs.

NOTE: Refer to Appendix E for the MDX-DEBUG System Configuration and Checkout.

FIGURE 1-1 BOARD PHOTO



1-7. DDT-80 COMMAND SUMMARY

M s - Display and/or update the contents of memory location s.

M s,f - Tabulate the contents to memory locations s through f.

P s - Display and/or update the content of I/O port s.

D s,f - Dump the contents of memory locations s through f in a format suitable to read by the L command.

L - Load, into memory, data which is in the appropriate format.

E s - Transfer control from DDT-80 to a user's program starting at location s.

H - Perform 16 bit hexadecimal addition and/or subtraction.

C s,f,d - Copy the contents of memory locations s through f to another location in memory starting at location d.

6

B s - Insert a breakpoint in the user's program (must be in RAM) at location s which transfers control back to DDT-80. This allows the user to intercept his progam at a specific point (location s) and examine memory and CPU registers to determine if this program is working correctly.

R - Display the contents of the user registers.

The s,f, and d represent start, finish, and destinations operands required for each command.

1-8. MEMORY, PORT AND REGISTER COMMANDS (M,P,R)

The M, P, and R commands provide the means for displaying the contents of specified memory locations, port addresses, or CPU

registers. The M and P commands sequentially access memory locations or ports and display their contents. The user has the option of updating the content of the memory location or port. (Note some ports are output only and their contents cannot be displayed). The M command also gives the user access to the CPU registers through an area in RAM called the Register Map (discussed in the Execute, Breakpoint section below).

The M and R commands are used to tabulate blocks of memory locations (M) or the CPU registers (R). The M command will accept two operands, the starting and ending address of the memory block to be tabulated. The R command will accept either no operand or one. If no operand is specified, the CPU registers will be displayed without a heading. If an operand is specified then a heading which labels the register contents will be displayed as well.

1-9. EXECUTE AND BREAKPOINT (E,B)

The E command is used to execute all programs, including aids such as the Assembler. The B command is used to set a breakpoint to exit from a program at some predetermined location for debugging purposes. At the instant of a breakpoint exit, the contents of all CPU registers are saved in a designated area of MDX-DEBUG RAM called the Register Map. In the Register Map, the register contents may be examined or modified using the M command and a predefined mnemonic (or absolute address) of the storage location for that register (example :PC,:A,...,:SP). The Register Map is also used to initialize the CPU registers whenever execution is initiated or resumed. Thus the E and B commands can be used together to initialize, execute, and examine the results of individual progam segments.

The B command gives the user the option of having all CPU

registers displayed when the breakpoint is encountered. This is done by entering a second operand to the B command. Otherwise DDT-80 defaults to displaying the PC and AF registers. When all CPU registers are diplayed, the format is the same as for the R command previously discussed.

1-10. LOAD, DUMP, AND COPY (L,D,C)

The L and D commands load and dump object files through object I/O channel in standard Intel Hex format. Checksums are used for error detection, and the addresses of questionable blocks are typed automatically while loading.

The C command will copy the contents of the memory block specified to another block of memory. There are no restrictions on the direction of the copy or on whether the blocks overlap.

1-11. HEXADECIMAL ARITHMETIC (H)

The H command is a dummy command used to allow hexadecimal addition and subtraction for expression evaluation without performing any other operation.

1-12. DDT-80 I/O CAPABILITIES

DDT-80 specifies I/O channels, designated 'Console', 'Object', and 'Source', to which any suitable devices may be assigned. The Channel Assignment Table is located in MDX RAM where it may be examined or modified using the M command. The table addresses correspond to the I/O channels and the table contents correspond to the addresses of the peripheral drivers routines. A channel which has a device assignment may have that device assignment changed using the M command. This is accomplished by merely

modifying the table contents of that channel's table address to correspond to the new peripheral driver routine. A set of peripheral driver routines is supplied and listed below. This scheme also allows the user to write a driver routine for his own peripheral, load it into memory, and easily configure that peripheral into the system.

1-13. DDT-80 I/O PERIPHERAL DRIVERS

1. A serial input driver (usually keyboard).
2. A serial output driver (usually a CRT or teletype typehead).
3. A serial input driver which sends out a reader step signal (usually a teletype reader).
4. A serial output driver which forces a delay after a carriage return (usually a Silent 700 typehead).
5. A parallel input driver (usually for high speed paper tape input).
6. A parallel output driver (usually for high speed paper tape output).
7. A parallel output driver (usually for a line printer).

1-14. TEXT EDITOR DESCRIPTION

The Text Editor permits random access editing of ASCII character strings. It can be used as a line or character oriented editor. Individual characters are typically read into memory from magnetic tape or paper tape. Each edited block can be output to magnetic tape or paper tape after editing is completed. While the primary application of the Text Editor is in editing assembly language source statements, it may be applied to any ASCII text delimited by "carriage returns".

The Editor has a macro command processing option. Up to two sets of commands may be stored and processed at any time during the editing process. All I/O is done via the DDT-80 channels. The Editor can be used with the MOSTEK ASMB-80 Assembler and Loader to edit, assemble, and load programs in memory without the need for external media for intermediate storage.

The following commands are recognized by the Text Editor:

An -	Advance record pointer n records
Bn -	Backup record pointer n records
Cn dS1dS2d-	Change string S1 to string S2 for n occurences
Dn -	Delete next n records
E -	Exchange current record with records to be inserted
I -	Insert records
Ln -	Go to line number n
Mn -	Enter command buffers (pseudo-macro)
N -	Print top, bottom, and current line number
Pn -	Punch n records from buffer
R -	Read source records into buffer
Sn dS1D -	Search for nth occurrence of string S1

1-15. ASSEMBLER DESCRIPTION

The Assembler reads Z80 source mnemonics and pseudo-ops and outputs an assembly listing and object code. The assembly listing shows address, machine code, statement number, and source statement. The object code is in industry standard hexadecimal format modified for relocatable, linkable assemblies.

The Assembler supports conditional assemblies, global symbols, relocatable programs and printed symbol table. It can assemble any length program, limited only by a symbol table. It can

assemble any length program, limited only by a symbol table size which is user selectable. Expressions involving addition and subtraction are allowed. A global symbol is categorized as "internal" if it appears as a label in the program; otherwise it is an "external" symbol. The printed symbol table shows which symbols are internal and which are external. The Assembler allows the user to select relocatable or non-relocatable assembly via the "PSECT" pseudo-op. Relocation records are placed in the object output for relocatable assemblies (The MOSTEK object format is defined below). The Assembler can be run as a single pass assembler or as a learning tool. (In this mode, global symbols and forward references are not allowed).

The following pseudo-ops are recognized by the Assembler:

ORG	-	program origin
EQU	-	equate label
DEFL	-	define label
DEFM	-	define message
DEFB	-	define byte
DEFW	-	define word
DEFS	-	define storage
END	-	end statement
NAME	-	program name definition
PSECT	-	program section definition
GLOBAL	-	global symbol definition supports the following assembler pseudo-ops
EJECT	-	eject a page of listing
TITLE	-	place heading at top of each page
LIST	-	turn listing on
NLIST	-	turn listing off

1-16. RELOCATING LINKING LOADER DESCRIPTION

The MOSTEK Relocating Linking Loader provides state-of-the-art capability for loading programs into memory by allowing loading and linking of any number of relocatable and non-relocatable object modules. Non-relocatable modules are always loaded at their starting address as defined by the ORG pseudo-op during assembly. Relocatable object modules can be positioned anywhere in memory at an offset address.

The Loader automatically links and relocates global symbols which are used to provide communication or linkage between program modules. As object programs are loaded, a table containing global symbol references and definitions is built up. At the end of each module, the loader resolves all references to global symbols which are defined by the current or a previously loaded module. It also prints on the console device the number of defined global symbols that have been referenced. The symbol table can be printed to list all global symbols and their load address. The number of object modules which can be loaded by the Loader is limited only by the amount of MDX-RAM available for the modules and the symbol table. Space for the symbol table is allocated dynamically downward in memory from either the top of memory or from a specified address entered as an operand of the load command.

The Loader prints the beginning and ending address of each module as it is loaded. The transfer address as defined by the END pseudo-op is printed for the first module loaded. The Loader execute command (E) can be used to automatically start execution at the transfer address.

The Loader Commands are the following:

L offset - load object module at address "offset" plus program origin address

E - execute loaded program at transfer address of first module

T - print global symbol table

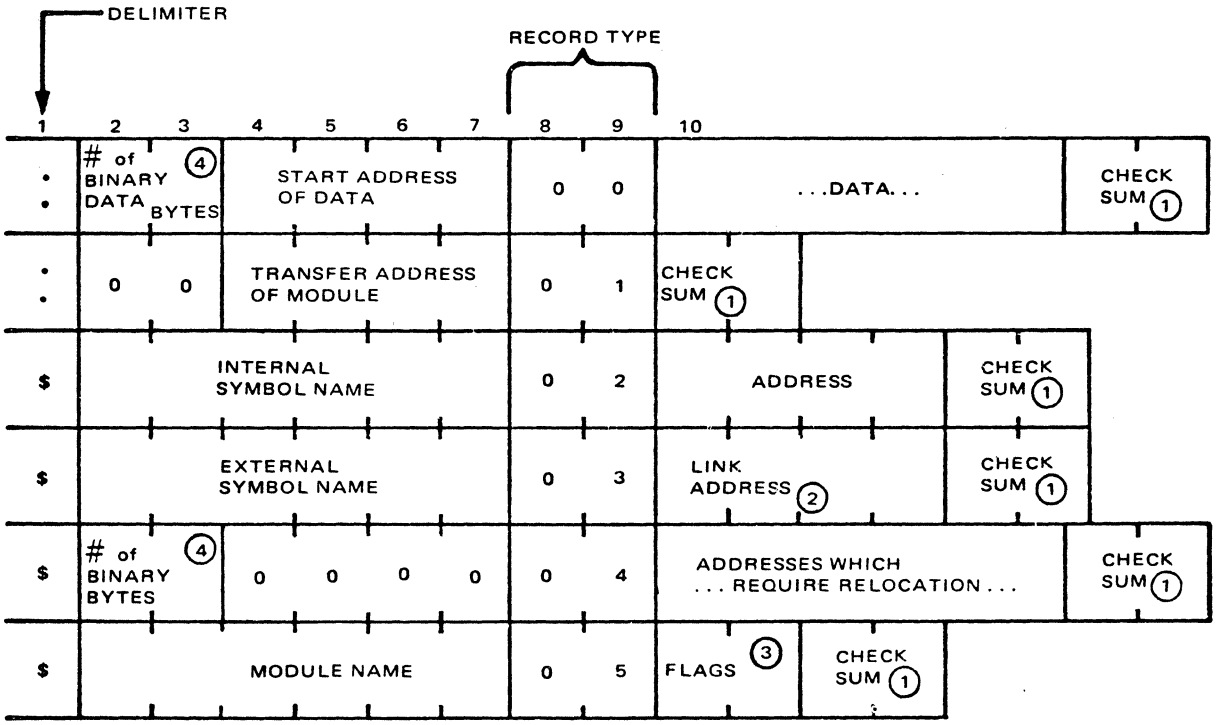
All I/O is done via the DDT-80 channels. Assemblies can be done from source statements stored in memory (by the Editor). The object output can be directed to a memory buffer rather than to an external device. Thus, assembly and loading can be done without external storage media.

1-17. MOSTEK OBJECT OUTPUT DEFINITION

Each record of an object module begins with a delimiter (colon or dollar sign) and ends with carriage return and line feed. A colon (:) is used for data records and end-of-file record. A dollar sign ($) is used for records containing relocation information and linking information. All information is in ASCII. Each record is identified by "type". The type is determined by the 8th and 9th bytes of the record which can take the following values:

00 - data
01 - end-of-file
02 - internal symbol
03 - external symbol
04 - relocation information
05 - module definition

FIGURE 1-2 OBJECT OUTPUT FORMAT



NOTES:

1. Check Sum is negative of the binary sum of all bytes except delimiter and carriage return/line feed.
2. Link Address points to last address in the data which uses the external symbol. This starts a backward link list through the data records for the external symbol. The list terminates at OFFFFH.
3. The flags are one binary byte. Bit 0 is defined as:
   0 - absolute module
   1 - relocatable module
4. Maximum of 64 ASCII bytes.

1-18. SPECIFICATIONS

1-19. ELECTRICAL SPECIFICATION

1-20. I/O TRANSFER RATE

X16 Baud Rate Clock	Baud Rate (Hz)
1,760	110
4,800	300
9,600	600
19,200	1,200
38,400	2,400
76,800	4,800
153,600	9,600
307,200	19,200

1-21. SERIAL COMMUNICATIONS CHARACTERISTICS

Asynchronous

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

1-22. SERIAL COMMUNICATIONS INTERFACE

SIGNAL	BUFFERED FOR:	
	20mA Current Loop	RS-232
Transmitted data	Output	Output
Received data	Input	Input

Data Terminal Ready (DTR)		Input
Request to Send (RTS)		Input
Carrier Detect (CDET)		Output
Clear to Send (CTS)		Output
Data Set Ready (DSR)		Output
Reader Step relay (RS)	Output (20mA)	

1-23. PARALLEL BUS INTERFACE-STD BUS COMPATIBLE

Inputs	One 74LS load Max
Bus Outputs	I_{OH}=15mA min at 2.4 Volts
	I_{OL}=24mA min at 0.4 Volts

1-24. I/O ADDRESSING

On-board Serial I/O Port
Control Port DDH
Data Port DCH
Module and Reader Step Control Port DEH

1-25. SYSTEM CLOCK

	MIN	MAX
MDX-DEBUG	1.12MHz	2.5MHz
MDX-DEBUG-4	1.25MHz	4.0MHz

1-26. POWER SUPPLY REQUIREMENT

+12 Volts ± 5% at 50 mA max.
-12 Volts ± 5% at 35 mA max.
+5 Volts ± 5% at 1.2 A max.

1-27 OPERATING TEMPERATURE

0°C to + 50°C

1-28. MECHANICAL SPECIFICATIONS

1-19. CARD DIMENSIONS

4.5 in. (11.43cm) high by 6.50 in.(16.51cm) long
0.48 in. (1.22cm) maximum profile thickness
0.062 in. (0.16cm) printed circuit board thickness

1-30. CONNECTORS

FUNCTION	CONFIGURATION	MATING CONNECTOR
STD BUS	56 pin dual read out 0.125 in. centers	Printed Circuit Viking 3VH28/1CE5 Wire Wrap Viking 3VH28/1CND5 Solder Lug Viking 3VH28/1CN5
Serial I/O	26 pin dual read out 0.100 in. grid	Flat Ribbon Ansley 609-2600M Discrete Wires Winchester PGB26A (housing) Winchester 100-70020S (contacts)

1-31. 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 gound 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). D_0-D7 constitute an 8-bit bidirectional data bus. The data bus is used for data exchange with memory and I/O devices.
10	D6	
11	D1	
12	D5	
13	D0	
14	D4	
15	A7	
16	A15	
17	A6	
18	A14	
19	A5	
20	A13	Address Bus (Tri-state, output, active

21	A4	high). A_0-A_{15} make up a 16-bit
22	A12	address bus. The address bus provides
23	A3	the address for memory (up to 65K by-
24	A11	tes) data exchanges and for I/O device
25	A2	data exchanges. I/O addressing uses
26	A10	the lower 8 address bits to allow the
27	A1	user to directly select up to 256
28	A9	input or 256 output ports. A_0 is
29	A0	the least significant address bit.
30	A8	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 $\overline{IORQ}$ 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 $\overline{M1}$ time, while I/O operations never occur during $\overline{M1}$ 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	$\overline{IOEXP}$	I/O expansion, not used on Mostek MDX cards.
36	$\overline{MEMEX}$	Memory expansion, not used on Mostek MDX cards
37	$\overline{REFRESH}$	REFRESH (Tri-state, output, active low). $\overline{REFRESH}$ indicates that the lower 7 bits of the address bus contain a refresh address for dynamic memories and the $\overline{MEMRQ}$ 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 contain the I register.
38	$\overline{DEBUG}$	$\overline{DEBUG}$ (Input) is used in conjunction with DDT-80 operating system and the MDX Single-Step card for implementing a hardware single step. When pulled low, the $\overline{DEBUG}$ 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	$\overline{M1}$	Machine Cycle One (Tri-state, output, active low) $\overline{M1}$ 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, $\overline{M1}$ will be generated as each op code is fetched. These two-byte op-codes always begin with a CBH, DDH,EDH, or FDH. $\overline{M1}$ also occurs with $\overline{IROQ}$ to indicate an interrupt acknowledge cycle.
40	Status Ø	Not used on Mostek MDX cards.
41	$\overline{BUSAK}$	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 acccepts 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 synchron-

ized to the CPU.

46 $\overline{\text{NMIRQ}}$ Non-Maskable Interrupt Request (Input, negative edge triggered). The non maskable 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 to zero, disable the 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

1-32. STD-Z80 ELECTICAL BUS SPECIFICATIONS

Bus Receivers

Logical Low: 0.8V max at -0.36 mA
Logical High: 2.0V min at 20 microamperes

Bus Drivers

Logical Low: 0.5V at 24 mA
Logical High: 2.4V at -15 mA
Off State Output Current (tri-state) ±100 microamperes

Recommended Bus Drivers and Receivers

Bus Drivers - 74LS240, 74LS241, 74LS373, 74LS374, 74LS244

Bus Receivers - 74LS240, 74LS241, 74LS244

Bus Transceivers - 74LS245, 74LS242, 74LS243

NOTES:

(1) The input/output designations of the STD-Z80 BUS are made with respect to the MDX-CPU1 card.

(2) The following signals have pull-up resisters: $\overline{WR}$, $\overline{RD}$, $\overline{IORQ}$, $\overline{MEMRQ}$, $\overline{REFRESH}$, $\overline{DEBUG}$, $\overline{M1}$, $\overline{BUSRQ}$, $\overline{INTAK}$, $\overline{INTRQ}$, $\overline{WAITRQ}$, $\overline{NMIRQ}$, $\overline{SYSRESET}$, $\overline{PBRESET}$ and $\overline{CLOCK}$. The value of the pull-up resistors are 1K Ohm except for $\overline{WAITRQ}$, which is 500 Ohm and $\overline{PBRESET}$ while is 10K Ohm.

2-1. INTRODUCTION

2-2. The MDX-DEBUG is specialized version of the MDX-PROM/UART. The difference between the MDX-DEBUG and the MDX-PROM/UART is that the MDX-DEBUG has the ASMB-80/DDT-80 ROM firmware. If desired, the ROM firmware may be removed and EPROMs inserted for custom applications.

NOTE: Refer to the ASMB-80/DDT-80 Operations Manual for instructions for using ASMB-80/DDT-80.

2-3. MEMORY ARRAY. The memory array consists of up to five MK2716's (2K x 8, +5V only EPROM/ROM). The total storage of the MDX-DEBUG board is 10,240 bytes.

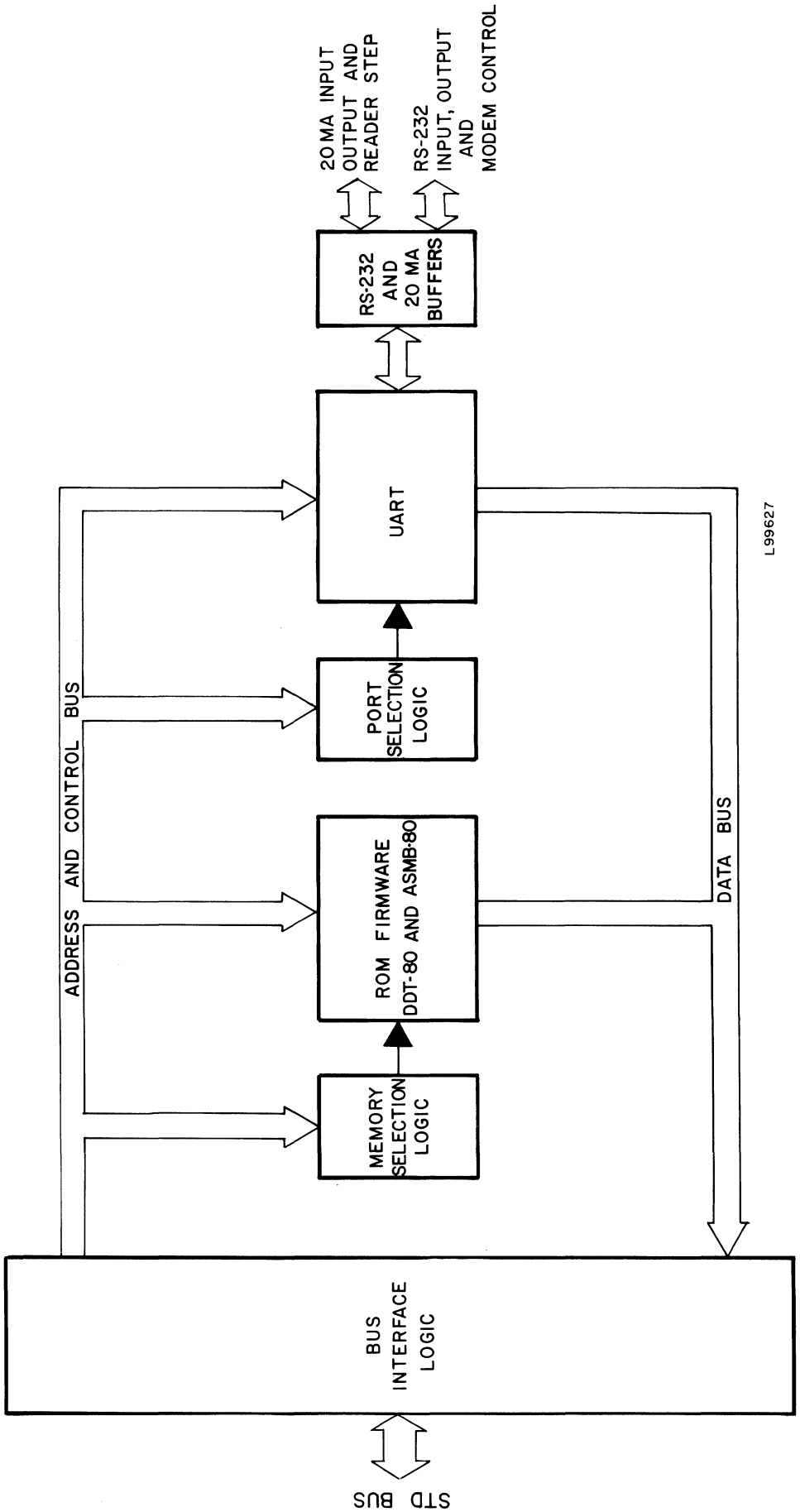
2-4. MEMORY SELECTION LOGIC . The memory selection logic is responsible for decoding the selected address and enabling the selected EPROM/ROM onto the data bus through the data bus buffer.

2-5. PORT SELECTION LOGIC. The port selection logic is used to allow read and write operations to the Serial I/O port (UART).

2-6. UART. The UART on the MDX-DEBUG 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.

2-7. RS-232 AND 20mA BUFFERS. 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 equipments.

FIGURE 2-1 MDX-DEBUG BLOCK DIAGRAM



3-1. INTRODUCTION

3-2. The MDX-DEBUG is a specialized version of the MDX-PROM/UART. The main difference between the boards is the installation of the MDX-DEBUG ASMB-80/DDT-80 ROM firmware package. The MDX-DEBUG comes ready to use i.e.; all memory decoding jumpers are installed. The ASMB-80/DDT-80 firmware package comes jumpered at C000-E7FF. If desired, the firmware ROMs may be removed and EPROMs inserted. In that case, read section 3-4 for installing memory decoding jumpers.

NOTE: For instructions on how to use ASMB-80 and DT-80, refer to the ASMB-80/DDT-80 Operations Manual.

3-3. MEMORY INTERFACE

3-4. EPROM/ROM DECODING JUMPERS

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

3-6. SERIAL INTERFACE

3-7. BAUD RATE SELECTION

3-8. 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 U8 setting versus baud rates.

3-9. PROGRAMMING THE UART

3-10. 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 to 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 3-1.

3-11. SERIAL I/O CONNECTOR AND CABLE

3-12. All serial interface lines are brought out to a 26-pin connector J3.

3-13. The serial cable is done in the following way:

1) 26-pin connector is Ansley No. 609-2601M
2) 26 wire flat cable is Ansley No. 171-26
3) 26-pin standard EIA-RS232 connector is Ansley No. 609-259

3-14. Table 3-3. shows the interconnection between the 26-pin connector and the 25-pin RS-232 connector.

3-15. RS-232 INTERFACE

3-16. Because the MDX-DEBUG was designed to talk 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.

3-17. For example:

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

TABLE 3-1 EPROM/ROM DECODING JUMPER OPTIONS

	J2 JUMPERS					JUMPERS E1 - E6
DECODED ADDRESS	EPROM U17	EPROM U18	EPROM U19	EPROM U20	EPROM U21	
	J2	J2	J2	J2	J2	
0-7FF	Pin 16 to Pin 1	Pin 15 to Pin 1	Pin 14 to Pin 1	Pin 13 to Pin 1	Pin 12 to Pin 1	
800-FFF	" 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	E3 to E5
A800-AFFF	" 6	" 6	" 6	" 6	" 6	
B000-B7FF	" 7	" 7	" 7	" 7	" 7	

TABLE 3-1 (Contd)

						JUMPERS E1 - E6
DECODED ADDRESS	EPROM U17	EPROM U18	EPROM U19	EPROM U20	EPROM U21	
	J2	J2	J2	J2	J2	
B800-BFFF	Pin 16 to Pin 8	Pin 15 to Pin 8	Pin 14 to Pin 8	Pin 13 to Pin 8	Pin 12 to Pin 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-PROM/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

U8 SWITCH

POSITION 4	3	2	1	X16 CLOCK	BAUD RATE
0	0	0	0	.8KHZ	50
0	0	0	1	1.2	75
0	0	1	0	1.76	110
0	0	1	1	2.1523	134.5
0	1	0	0	2.4	150
0	1	0	1	4.8	300
0	1	1	0	9.6	600
0	1	1	1	19.2	1200
1	0	0	0	28.8	1800
1	0	0	1	32.081	2000
1	0	1	0	38.4	2400
1	0	1	1	57.6	3600
1	1	0	0	76.8	4800
1	1	0	1	115.2	7200
1	1	1	0	153.6	9600
1	1	1	1	316.8	19,200

1 = OPEN 0 = CLOSED

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

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.

3-18. 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.

3-19. TELETYPE AND READER STEP INTERFACE

3-20. Figure 3-2 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 RST 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.

Figure 3-1 PROGRAMMING THE UART

2. UART DATA PORT DC_H

Write to Port DC_H	DATA TO SERIAL DEVICE
Read from UART	DATA FROM SERIAL DEVICE

3. UART CONTROL PORT DD_H

Write to PORT DD_H	TSB	NP	EPS	NB_2	NB_1			
Read from PORT DD_H	TBmT	DAV	OR	FE	PE			

4. System Control Point DE_H

Write to Port DE_H						RS	CTS	DSR
Read from Port DE_H	SI					RTS	DTR	

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 "0" 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 — 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

"0" will insert odd parity and a logic "1" will insert even parity.

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	NB2	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.

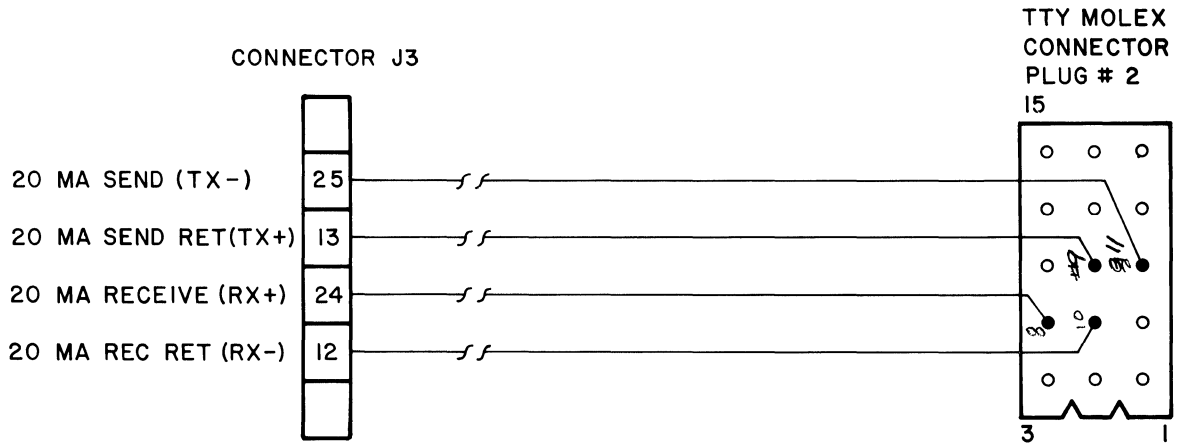
Framing Error (FE)

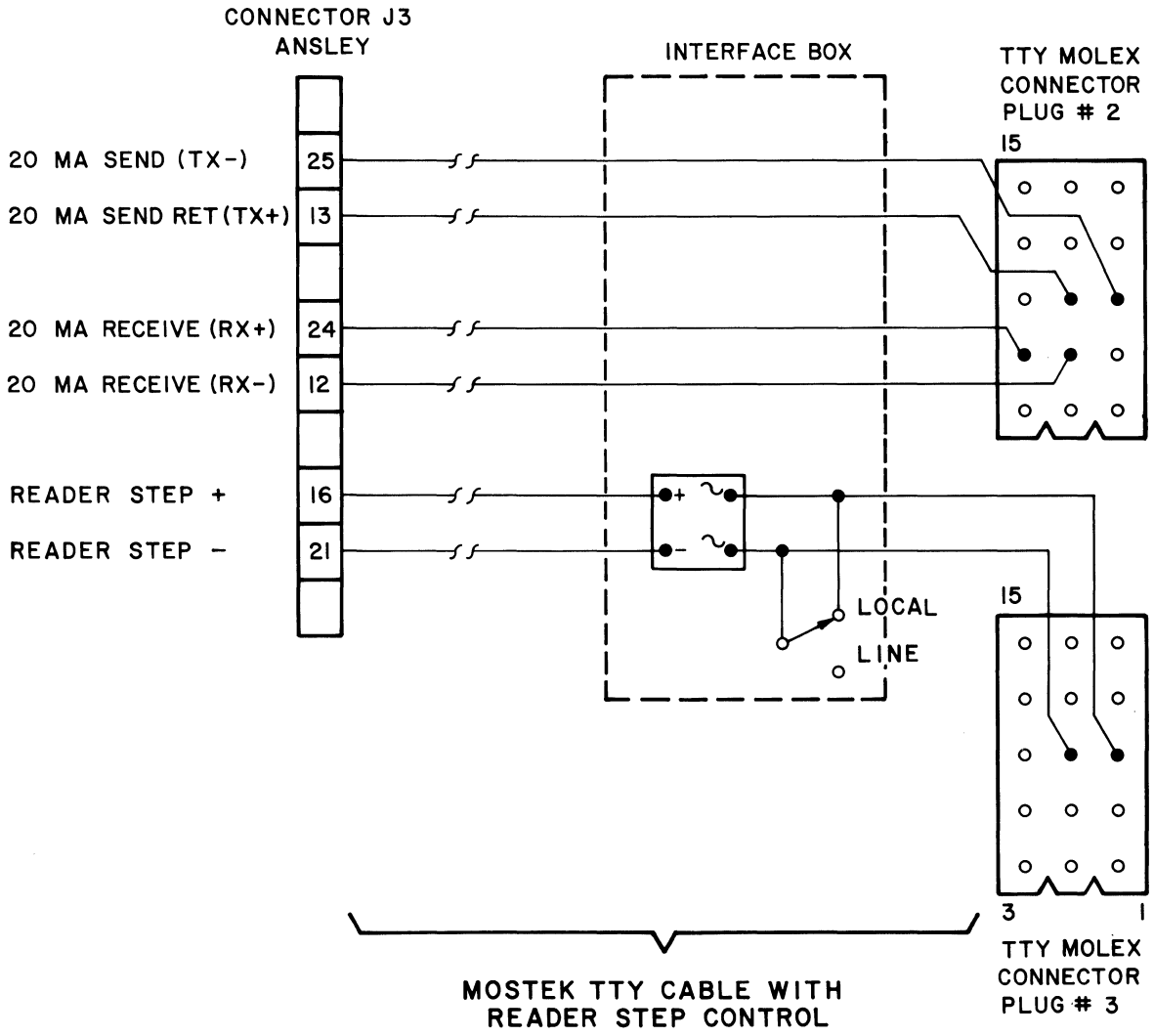
This bit goes to a logic "1" if

	the received character 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.
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.
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.

FIGURE 3-2 TTY CONNECTION WITHOUT READER STEP





APPENDIX A

MDX-DEBUG SYSTEM CONFIGURATION AND CHECKOUT

The purpose of this section is to provide the MDX user with the system configuration, list of minimum equipment, and checkout procedure for using the MDX-DEBUG board.

For checking jumper options, the user should consult the appropriate MDX user's manual.

I. MINIMUM EQUIPMENT

- a. MDX-CPU1
- b. MDX-DRAM 8/16/32
- c. MDX-DEBUG
- d. Serial ASCII terminal such as ASR-33, Silent 700, Hazeltine 1500 or ADB-3 CRT with cable
- e. +5VDC power supply @ 2.5A
- f. +12VDC power supply @ 350mA
- g. -12VDC power supply @ 150mA
- h. Card cage with motherboard or wire-wrapped backplane
- i. Reset switch (SPST push button normally open)

II. MDX-CPU1 CONFIGURATION

- a. 256 x 8 scratchpad memory enabled
- b. On board EPROM memory must not conflict with external RAM and must not be located at C000-E7FF or FF00-FFFF
- c. Restart to E000 enabled
- d. If the MDX-SST single step card is to be used, connect a jumper from E1 to E2.

III. MDX-DRAM 8/16/32 CONFIGURATION

- a. MDX-DRAM board configured for starting address of 0000

b. RAM memory must not extend beyond 3FFF.

IV. MDX-DEBUG CONFIGURATION

a. ASMB-80 and DDT ROMs installed
b. Memory decoding enabled at C000-E7FF
c. ASCII terminal connected via cable to MDX-DEBUG
d. Set the baud rate generator to match the terminals baud rate.

V. POWER SUPPLY CONFIGURATION

a. Connect +12V power supply to backplane Pin 55 CON-7
b. Connect -12V power supply to backplane Pin 56 CON-6
c. Connect +5V power supply to backplane Pins 1 and 2 - CON-1
d. Connect power supply grounds to backplane Pins 3 and 4 - CON-3

VI. RESET BUTTON

a. Connect a SPST N.O. push button switch to backplane Pin 48 to ground

VII. SYSTEM CHECKOUT

a. Install the boards in the card cage. Be sure boards are oriented properly, or permanent damage could result.
b. Install the serial terminal cable
c. Power up the system
d. Push and release the reset button
e. Type a character on the terminal keyboard
f. The DDT-80 should respond with a "."
   If the system does not respond with a "." , power down the system and recheck the following:
   1. Power supplies and connections
   2. Backplane connections
   3. Jumper options and swithces
   4. Proper card insertion
   5. Terminal cable

Power up the system again, and repeat the procedure. If the system fails to respond with a ".", call Mostek for further assistance.

g. If the system responds with a ".", then the system is now executing DDT-80, and commands may be entered.

The following terminology is used for the example below:

	indicates carriage return
	indicates up carat
.	indicates DDT-80 proompt character
_	All entered responses are underlined
XX	indicates unknown

Enter the following commands:

```
                              COMMENTS
.M 0                          MODIFY MEMORY LOCATION 0
0000 XX FF_                   LOAD FF
0000 FF 00_                   LOAD 00
0000 00 .                     REENTER COMMAND MODE
.E  :ED                       EXECUTE EDITOR
                              EDITOR PROMPT
PRESS RESET BUTTON AND TYPE CHARACTER ON TERMINAL
    . E  :AS
    OPTIONS?
PRESS RESET BUTTON AND TYPE CHARACTER ON TERMINAL
                              DDT-80 MONITOR PROMPT
```

If the system responds as shown above, the MDX-DEBUG system is ready for operation. Refer to the ASMB-80/DDT-80 operations manual for information on how to use the text editor assembler, linking loader, and DDT-80 operating system.

APPENDIX B

FACTORY REPAIR SERVICE.

In the event that difficulty is encountered with this unit, it may be returned directly to MOSTEK for repair. This service will be provided free of charge if the unit is returned within 90 days of purchase. However, units which have been modified or abused in any way either will not be accepted for service or will be repaired at the owner's expense.

When returning the circuit board, place it inside the conductive plastic bag in which it was delivered in order to protect the MOS devices from electrostatic discharge during shipment. The circuit board must NEVER be placed in contact with styrofoam materials. ENCLOSE a letter containing the following information with the returned circuit board:

Name, address, and phone number of purchaser
Date and place of purchase
Brief description of the difficulty

Mail a copy of this letter SEPARATELY to:

MOSTEK Corporation
Microcomputer Service Manager
1215 West Crosby Road
Carrollton, Texas 75006

Securely package and mail the circuit board, prepaid and insured to:

MOSTEK Corporation
Microcomputer Service Department
1215 West Crosby Road
Carrollton, Texas 75006

LIMITED WARRANTY

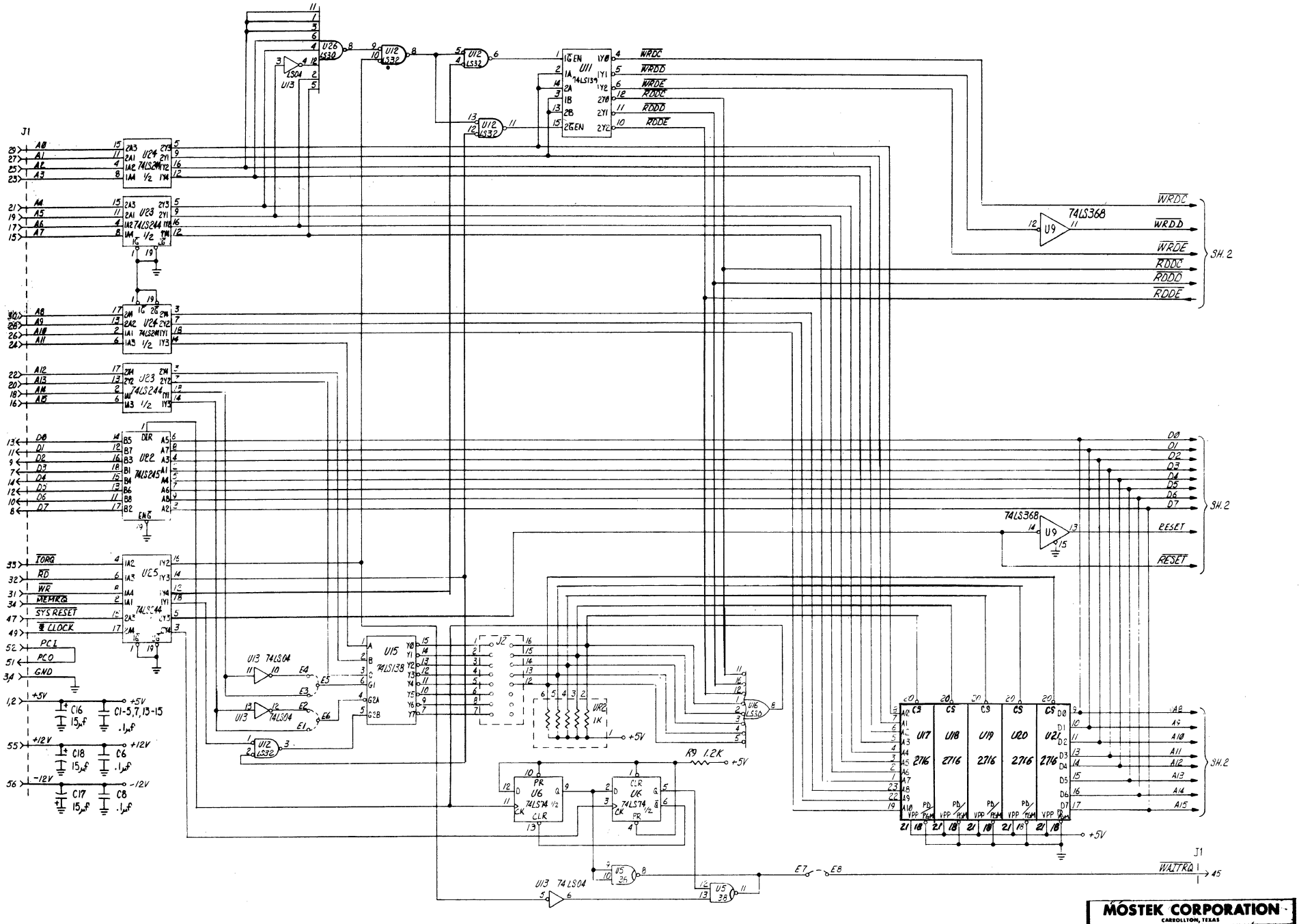
MOSTEK warrants this product against defective materials and workmanship for a period of 90 days.

This warranty does not apply to any product that has been subjected to misuse, accident, improper installation, improper application, or improper operations, nor does it apply to any product that has been repaired or altered by other than MOSTEK personnel.

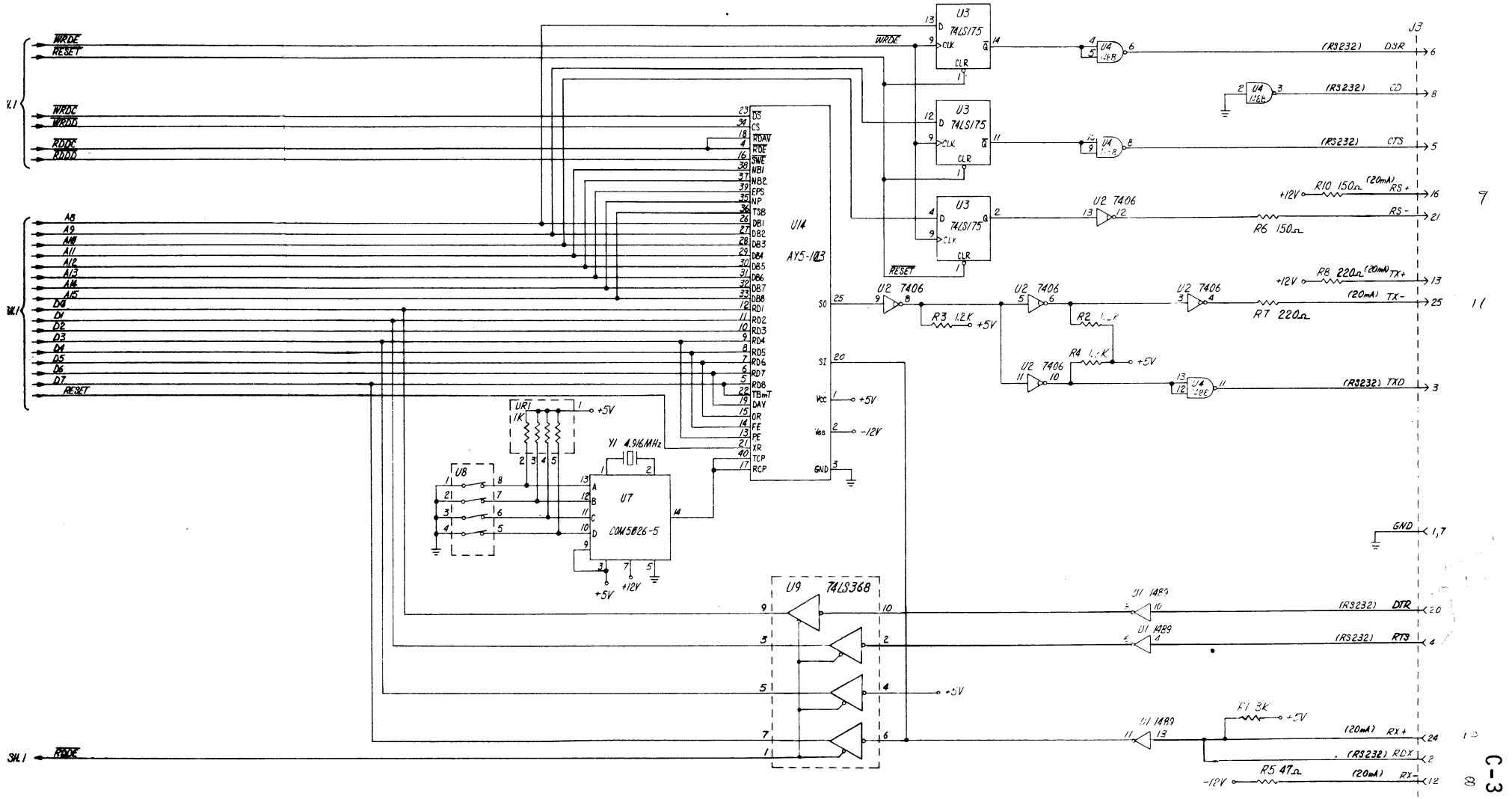
There are no warranties which extend beyond those herein specifically given.

APPENDIX C

SCHEMATIC DIAGRAMS



MOSTEK CORPORATION	CARROLLTON, TEXAS
TITLE	SCHEMATIC – P.C.B., PROM / ROM SERIAL I/O
DWG	REV

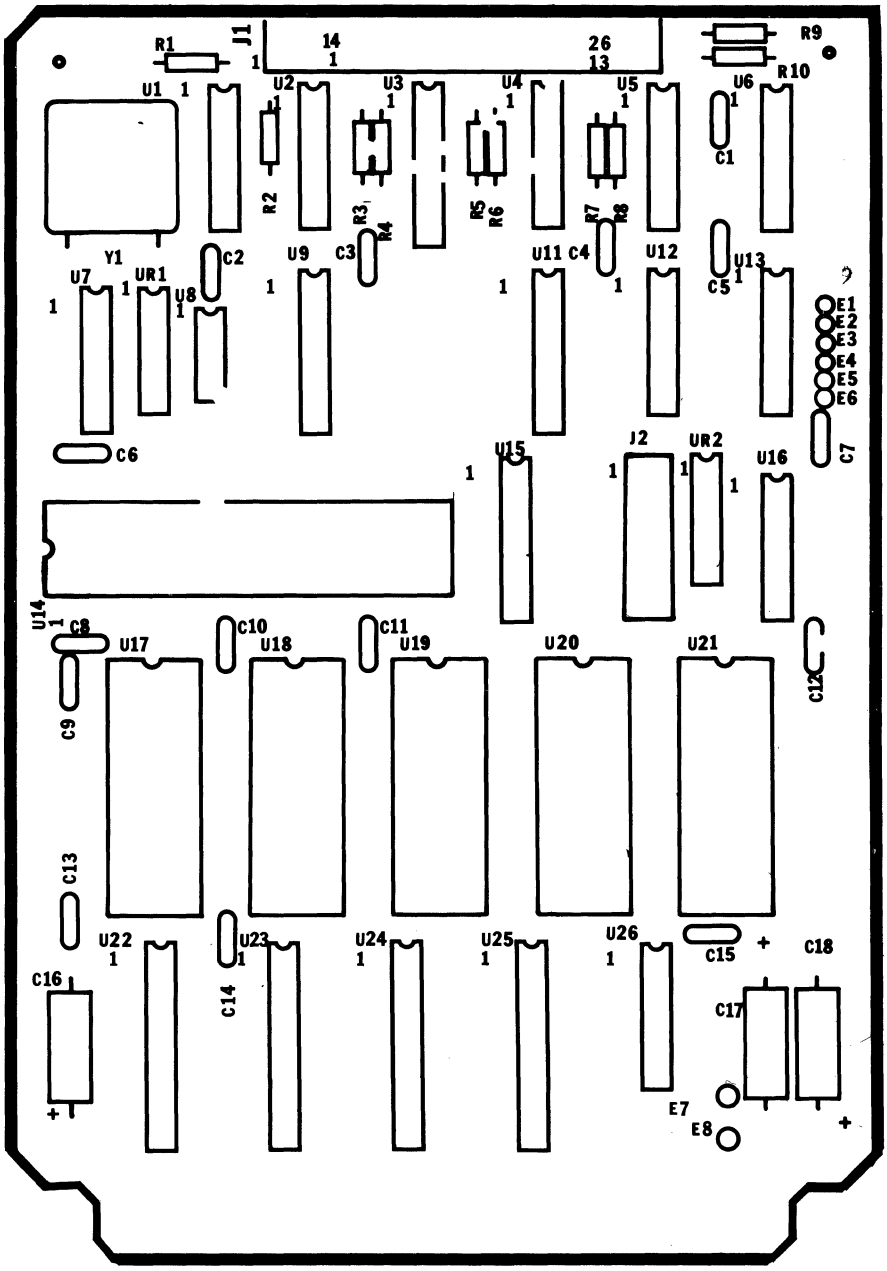


MOSTEK CORPORATION	CARROLLTON, TEXAS
TITLE	SCHEMATIC - P.C.B. PROM/ROM SERIAL I/O
DWG NO	450-0036B-00
REV	C
SHEET	2/2

C-3

APPENDIX D

PARTS LIST


R1
J1
14
1
26
13
R9
R10
U1
U2
U3
U4
U5
U6
C1
R2
R3
R4
R5
R6
R7
R8
Y1
C2
U9
C3
U11
C4
U12
U13
C5
U7
UR1
U8
E1
E2
E3
E4
E5
E6
C6
C7
U15
J2
UR2
U16
U14
C8
C10
C11
U17
U18
U19
U20
U21
C12
C9
C13
U22
U23
U24
U25
U26
C15
C18
C16
C14
C17
E7
E8

PART NO.	QTY	DESCRIPTION	REFERENCE DESIGNATOR	USED ON
4610111	1	FAB 450-00366-00 REV C	A MD PROM/ROM	77753
0000000		SCH 450-00368-00 REV C	A MD PROM/ROM	77753
0000000		ASSY 450-00367-00 REV C	A MD PROM/ROM	77753
4150111	15	CAPACITOR .1UF	C1-15	77753
4150140	3	CAPACITOR 15 UF	C16-18	77753
4280007	8	WIRE WRAP TERMINAL	E1-8	77753
4210132	1	HEADER 26 PIN R/A	J1	77753
4210143	1	HEADER 16 PIN	J2	77753
4470084	1	RESISTOR 3K	R1	77753
4470053	1	RESISTOR 150 OHM	R10	77753
4470075	4	RESISTOR 1.2K	R2,3,4,9	77753
4470041	1	RESISTOR 47 OHM	R5	77753
4470057	3	RESISTOR 220 OHM	R6,7,8	77753
4313258	1	IC 1489	U1	77753
4313567	1	IC 74LS139	U11	77753
4313411	1	IC 74LS32	U12	77753
4313288	1	IC 74LS04	U13	77753
4313181	1	IC AY-5-1013A	U14	77753
4313295	1	IC 74LS138	U15	77753
4313410	2	IC 74LS30	U16,26	77753
4313008	1	IC 7406	U2	77753
4313508	1	IC 74LS245	U22	77753
4313507	3	IC 74LS244	U23,24,25	77753
4313306	1	IC 74LS175	U3	77753
4313564	1	IC 1488	U4	77753
4313078	1	IC 7438	U5	77753
4313413	1	IC 74LS74	U6	77753
4313419	1	IC 5026-005	U7	77753
4640006	1	SWITCH DIP 4 POS	U8	77753
4313566	1	IC 74LS368	U9	77753
4470178	2	RESISTOR SIP 6 PIN 1K	UR1,2	77753
4620019	1	SOCKET 40 PIN	X14	77753
4620018	5	SOCKET 24 PIN	X17-21	77753
4230016	1	CRYSTAL 4.916 MHZ	Y1	77753
4140003	1	EJECTOR	Z	77753
5025266	1	TRAVELER WHIP	Z:NOTE IN HOUSE USE ONLY	77753
5013204	1	BOX SHIPPING	Z:SHIPPED NOT ASSEMBLED	77753
5013004	2	BAG ANTISTATIC	Z:SHIPPED NOT ASSEMBLED	77753

MOSTEK®
Z80·F8 Covering the full spectrum of
3870 microcomputer applications.

1215 W. Crosby Rd. • Carrollton, Texas 75006 • 214/242-0444
In Europe, Contact: MOSTEK Brussels
150 Chaussee de la Hulpe, B1170, Belgium;
Telephone: (32) 02/660-2568/4713

PRINTED IN USA November 1978
Publication No. MK 79611