UNIVERSAL TRAINER

REFERENCE MANUAL

May 10, 1993
REVISION 1.3

EMAC, inc.

EQUIPMENT MONITOR AND CONTROL

CARBONDALE, IL 62901
618-529-4525

TABLE OF CONTENTS

APPENDICES

SOFTWARE FEATURES

* Easy data entry via keypad and display or PC/dumb terminal.
* Ability to scan all 64K of memory space.
* Ability to change any RAM location.
* Ability to execute from any memory location.
* Ability to single step or run full speed with break.
* Examine and change the contents of any register.
* Separate user stack area.
* Ability to store user programs on board to RAMDISK or externally to PC.
* Ability to upload and download Intel hex files.
* Monitor Operating System provides services for I/O devices including LED display.
* I/O device Assembly Language drivers provided.

HARDWARE FEATURES

* Timers, UARTs, digital I/O ports, A/D, D/A, and DIP switch are all user accessible I/O devices.
* 1 eight bit digital input port connected to an eight position dip switch with status LED's.
* 1 eight bit digital output port with status LED's.
* 4 channel eight bit (ten bit optional) analog to digital converter.
* Analog signal conditioning card slot for optional signal conditioning of analog inputs.
* 1 RS232 serial port and 1 RS485 network port.
* 3 sixteen bit timer counters with gated inputs.
* 12 hardware interrupt sources.
* 8 character, 14 segment alphanumeric LED display with 28 key keypad.
* Status LEDs for address, data, and system status lines.
* Hardware single step with hardware breakpoint.
* Easy access to analog and digital I/O through screw terminal connectors.

INTRODUCTION

The Universal Trainer (UT) is a full featured computer providing the user easy programmability, yet access to sophisticated I/O devices. The self-contained unit consists of a single circuit board with user manual and wall mounted power supply. The unit contains a Monitor Operating System in EPROM allowing the user to single step, run full speed with breakpoints, and easily access and modify the microprocessor registers. User programs can be written virtually anywhere within the RAM space, providing ample room for even complex programs. Programs can be retained even with loss of power with optional RAMDISKs. Assembly Language Subroutine Drivers and MOS Services are provided for each I/O device to ease programming.

EMAC offers several options for the Universal Trainer including: Blank analog signal conditioning cards as well as conditioning cards for temperature, pressure, voltage, current, and more. Additional digital I/O, real time clock, EPROM programmer, networking, and IBM PC support software are also available.

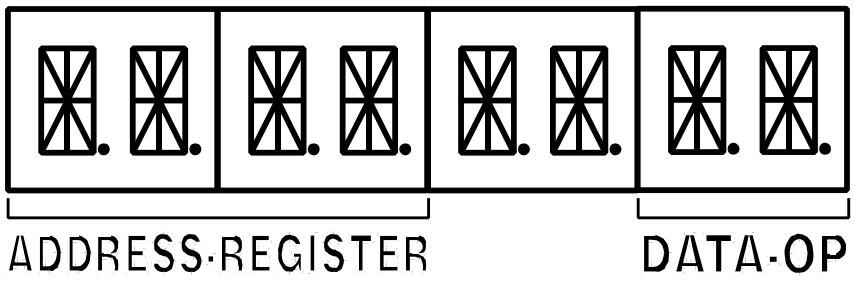
SOFTWARE REFERENCE

The Monitor Operating System (MOS) is a powerful software program that provides the user with the tools to enter and edit programs as well as run, test, and debug machine language programs. When the Universal Trainer is first turned on, interaction between the Monitor Operating System and the user is through the on-board keypad and display. The Extended Monitor Operating System (EMOS) can be invoked from this mode to allow interaction through a dumb terminal/PC. Each mode of interaction has its advantages and disadvantages.

Interaction Through Keypad and Display

When the Universal Trainer is first turned on or reset, all interaction with MOS is through the on-board keypad and display. This mode is highly functional for entering and debugging code. The major advantage of this mode is its ability to be used stand-alone (without any external hardware). This allows the UT a degree of portability not found in dedicated systems (those requiring a dumb terminal or PC). Another advantage of this input mode is cost. The unit can be purchased and used without having to purchase a dumb terminal or PC.

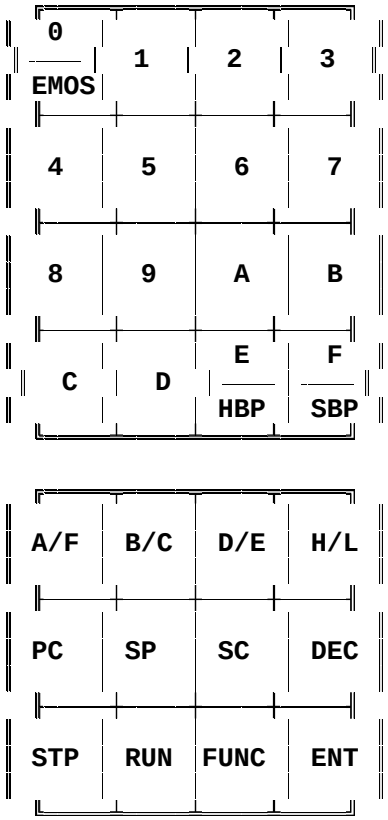
The keypad display mode uses eight, fourteen segment alphanumeric LEDs for display output. This type of display is similar to the displays used in calculators, but allows letters in addition to numbers. The ability to use letters gives the UT an advantage over other seven segment LED display based trainers. The UT is able to give descriptive prompts to the user making it easier to use.



The display at power-up reset, contains the default PC address (8F01H) in the left four characters of the display. This is what is referred to as the ADDRESS/REGISTER FIELD. In the right two characters is the contents of the displayed address. This field is known as the DATA/OP FIELD (see diagram).

The keypad display mode uses a twenty-eight key keypad for input. Each key has tactile feel (clicks when pressed) and produces an audible tone when pressed providing feedback for the user. The MOS controlled keys are in two groups of sixteen and twelve. The group of sixteen are Hexadecimal numeric keys (0 - F). The group of twelve keys are command keys. These keys allow the user to examine registers, change their contents, single step, and a variety of other functions. One of the 12 command keys is a FUNC. key, similar to a SHIFT key, providing additional functions when used with the Hex numeric keys. The UT has an additional 8 keys that are not included in the two groups mentioned above. These keys control hardware functions and are described elsewhere.

The layout of the twenty-eight key keypad for the Universal Trainer is as follows:



KEYPAD DESCRIPTION

KEY(S)	DESCRIPTION
0-F	Numeric keys. When a numeric key (0-F HEX) is pressed, the numeric value appears at the right side of the data field display. To correct an entry error just repeat, using the correct number key and the error will be overwritten.
A/F	This key displays the contents of the A register (Accumulator) and the condition Flags. The A register and the flags are displayed as four HEX digits. The two digits on the right of the display represent the contents of the A register and the two on the left represent the condition Flags. The displayed value may be changed by entering the desired HEX number using the numeric keys and then pressing the ENT key. The condition Flags are defined bit by bit as follows:

BIT 0	CARRY FLAG	(a set condition indicates a carry)
BIT 1	NOT USED	
BIT 2	PARITY FLAG	(a set condition indicates an even number of bits, odd parity)
BIT 3	NOT USED	
BIT 4	AUX. CARRY FLAG	(a set condition indicates a carry from bit 3 to bit 4)
BIT 5	NOT USED	
BIT 6	ZERO FLAG	(a set condition indicates a zero result)
BIT 7	SIGN FLAG	(a set condition indicates bit 7 of the A register is a 1)

For example if the display reads: 0044 A/F
This indicates the A register contains 0 and the ZERO and PARITY Flags are both set.

B/C This key displays the contents of the B/C register pair. The B/C register pair is displayed as four HEX digits. The two digits on the right of the display represent the contents of the B register and the two on the left represent the contents of the C register. The displayed value may be changed by entering the desired HEX number using the numeric keys and then pressing the ENT key.

D/E This key displays the contents of the D/E register pair. The D/E register pair is displayed as four HEX digits. The two digits on the right of the display represent the contents of the D register and the two on the left represent the contents of the E register. The displayed value may be changed by entering the desired HEX number using the numeric keys and then pressing the ENT key.

H/L This key displays the contents of the H/L register pair. The H/L register pair is displayed as four HEX digits. The two digits on the right of the display represent the contents of the H register and the two on the left represent the contents of the L register. The displayed value may be changed by entering the desired HEX number using the numeric keys and then pressing the ENT key.

PC This key displays the contents of the PROGRAM COUNTER. The PROGRAM COUNTER (PC) is a 16 bit register, displayed as four HEX digits. The displayed value may be changed by entering the desired HEX number using the numeric keys and then pressing the ENT key. The PC defaults to address 8F01 at power-up.

SP This key displays the contents of the STACK POINTER. The STACK POINTER (SP) is a 16 bit register, displayed as four HEX digits. The displayed value may be changed by entering the desired HEX number using the numeric keys and then pressing the ENT key. The SP defaults to address FFD4 at power-up.

SC This displays the two bytes that are at the top of the stack, or in other words, the data that would be removed from the stack if a POP instruction were executed. The left two displays show the byte at SP + 1 (stack pointer address + 1) and the two displays to the right of this represent the byte at SP. The displayed value may be changed by entering the desired hex number using the numeric keys and then pressing the ENT key. The top of the stack (SC) is initialized to zero at power-up.

DEC This key allows the user to decrement through memory, thus subtracting 1 from the PC every time the DEC key is pressed. The current PC address (4 HEX digits) and the contents of that address (2 HEX digits) are displayed. The address contents may be changed by entering the desired HEX number using the numeric keys and then pressing the ENT key.

STP This key causes the microprocessor to execute one instruction (single step) at the current PC address. Single Stepping is a valuable debugging tool that allows the user to examine registers and memory after each instruction executes. The STP key invokes a software single stepper that executes one user instruction and then returns to Monitor Operating System ready to except another user command. Unlike the Hardware Single Stepper, the Software Single Stepper requires the processor be active throughout the single stepping process. If single stepping a Service Call, the processor will execute the Service Call at full speed and stop at the instruction immediately following the Service Call (see section on single stepping for additional information).

RUN This key cause the microprocessor to execute a program at full speed starting at the current PC address. The program will continue to execute until an optional hardware or software breakpoint is encountered, a key is pressed, or until a RST 7 (0FFH) instruction is executed.

FUNC. This key selects the second function of the keys that have two functions. When this key is pressed "FUNCTION" will appear on the displays and if a key is pressed which has a second function, that function will be performed.

ENT When the UT is turned on or reset the MOS is in data entry mode. When in this mode, pressing ENT will cause the data which is shown on the right two displays to be stored into the address pointed to by the PC register (which is shown on the left four displays) and the PC register will be incremented. When registers or hard or soft breakpoints are being displayed, pressing the ENT key causes the data that is being displayed to be stored in that register or for that breakpoint to be set. If in any mode you have typed some numbers and you want to restore the original value, you can restore them if you haven't press ENT yet. Press the "Func." key twice and you will be back in data entry mode and the data, registers or breakpoints will retain their original values.

(Below are the second functions of the dual function keys)

EMOS This starts the EMOS (Extended Monitor Operating System). This requires that a PC or dumb terminal be connected to COM1.

HBP This key displays the current Hardware Breakpoint address. The displayed value may be changed by entering the desired breakpoint address in HEX using the numeric keys and then pressing the ENT key. The act of pressing the ENT key is what arms the Hardware Breakpoint arming circuit. A hardware breakpoint can only occur when the circuit is armed. Upon the occurrence of a Breakpoint, the Hardware Breakpoint address is still maintained, however the Hardware Breakpoint arming circuit is disarmed. To rearm the hardware breakpoint for the same address simply press the HRD BRK key followed by the ENT key. NOTE: If the program execution never accesses the breakpoint address, the program can not stop at the breakpoint address. The hardware breakpoint, breaks at any address including those in EPROM and operand addresses (see Section on breakpoints for additional information).

SBP This key displays the current Software Breakpoint address. If 0000 is displayed,

then no breakpoint has been established. The displayed value may be changed by entering the desired breakpoint address in HEX using the numeric keys and then pressing the ENT key. Upon the occurrence of a Breakpoint the Software Breakpoint address is automatically reset to 0000. NOTE: If the program execution never reaches the breakpoint address or the breakpoint address is not that of an opcode, the program will not stop at the breakpoint address. Also any address specified that references addresses in EPROM (addresses below 8000 HEX) will not stop execution, even if the opcode at that address is executed (see section on breakpoints for additional information).

Interaction Through a Dumb Terminal/PC

The most productive way to use the Universal Trainer is through EMOS in which all interaction is through a dumb terminal/PC. This mode obviously allows more information to be exchanged easier than with the keypad display mode. Additional functions are available through the dumb terminal/PC mode that are not possible with the keypad display input mode.

When a dumb terminal/PC is available you may connect it to the Universal Trainer and invoke EMOS by pressing "Func." then the "EMOS" key. The following will appear on the terminal display:

EMOS Vx.xx HELP MENU

```
B --> Bring Block from RAMDISK to Memory
C --> Change register contents
D --> Dump memory contents
E --> Edit memory contents
F --> Fill memory with byte
G --> Go execute program { full speed }
H --> Hex/Decimal math {1st + 2nd, 1st - 2nd}
I --> Input from I/O port
L --> List memory contents using mnemonics
M --> Move section of memory
O --> Output to I/O port
P --> Printer ON/OFF
R --> display Register contents
S --> MOS Service call
T --> Trace program execution
W --> Write memory to RAMDISK block
< --> hex download from trainer to host
> --> hex upload to trainer from host
? --> display this help menu
```

The above Help Menu can be displayed at any time from the EMOS **-** prompt by typing a ?. Each of the EMOS commands are invoked by type a single character at the EMOS prompt. EMOS will prompt the user for any necessary command input and checks the input to be sure it is of the proper form. When an "ADDRESS" is requested, a number of up to 4 HEX digits should be entered. When a "BLOCK" is requested, a number of up to 3 DEC (decimal) digits should be entered. The "escape" key or bad input will abort the issued command and cause a "?" to be displayed. Following are descriptions of the EMOS commands.

EMOS COMMAND SUMMARY

"B" The Bring Block command allows the user to copy a block (256 bytes) or blocks of data from RAMDISK to memory.

When the Bring Block command is issued, the Monitor responds with:

DEST. ADDRESS ...

Enter the starting memory address, up to 4 HEX digits, where the RAMDISK data will be copied to.

The Monitor then responds with:

STARTING BLOCK ...

Enter the starting block number, up to 3 DEC digits, of the desired first block.
NOTE: RAMDISK slot 1 contains blocks 1 - 128, and slot 2 contains blocks 129 - 255.

The Monitor then responds with:

NUMBER OF BLOCKS ...

Enter the number of blocks, up to 3 DEC digits, that are to be copied to memory.

NOTE: If (STARTING BLOCK + NUMBER OF BLOCKS) is greater than 255, the monitor will display:

ERROR BLOCK OUT OF RANGE !

and, after a short delay, return to the system prompt.

"C" The Change command allows the user to change any of the microprocessor's registers, the top element of the stack and the current memory location pointed to by the program counter.

When the Change command is issued, the Monitor program first displays the current contents of all the registers, and prompts the user with:

SELECT REG. [F,A,B,C,D,E,H,L,T,S,P,O]..

The Monitor then expects one of the capital letters contained within the brackets to be entered. The letter "F" stands for the Processor Status Word Flags, "A" is the Accumulator and "B,C,D,E,H,L" are the 8085 General Purpose Registers of the same name. Each of these registers can contain at most 2 HEX digits (8 bits). The letter "T" stands for the "top of stack" which are the next two bytes to be removed from the stack. Though this is not an actual register, it is treated like one so you can view and/or change the value on the top of the stack. The letter "S" stands for the Stack Pointer, which is initialized at power-up to address FFD4 HEX. The letter "P" stands for the Program Counter, which is initialized at power-up to address 8F01 HEX. Each of these registers can contain at most 4 HEX digits (16 bits). The last letter "O" stands for op code, and its selection allows the user to change the contents of the memory location pointed to by the Program Counter. This location is referred to as op code since it is assumed that any location the Program Counter points to is to be executed and thus must contain a valid op code. After entering a selected letter the Monitor then responds with:

CHANGE TO..

The maximum number of HEX digits that can be entered depends on which of the letters is chosen. The value entered will then replace the current contents and the registers are redisplayed.

"D" The Dump command allows the user to view the memory contents of any given section of memory. The contents are displayed in HEX with the associated ASCII characters displayed as well.

When the Dump command is issued the Monitor responds with:

STARTING ADDRESS..

The user then enters the starting address (up to 4 HEX digits are valid) of the section of memory to be Dumped.

After entering the starting address the Monitor then asks:

NUMBER OF BYTES..

The user then enters the number of bytes to display on the screen. Up to 3 HEX digits are valid and this number, as with most numbers returned by EMOS, is a HEX value. The Dump command automatically rounds this value so as to display a full line of 16 bytes.

"E" The Edit commands allows the user to easily enter and modify code or data in successive memory locations.

When the Edit command is issued the Monitor responds with:

STARTING ADDRESS..

The user then enters the starting address of where code is to be entered or modified. Up to 4 HEX digits are valid. The Edit command will then display this address and its contents. The user then enters the new contents (up to 2 HEX digits). The next successive address is then displayed with its contents. If the user does not want to change the contents of this address, he simply presses the <ENTER> key and the next successive address and memory contents are displayed. The ability to " backtrack " is also available; typing a minus sign " - " followed by one Hex digit (1 - F) will decrement the address pointer by the amount of the digit. To quit editing and return to the monitor prompt, press the <ESC> key.

"F" The Fill command allows the user to Fill the memory contents of any given section of memory with a given byte.

When the Fill command is issued the Monitor responds with:

STARTING ADDRESS..

The user then enters the starting address of the section of memory to be Filled. Up to 4 HEX digits are valid.

The Monitor then requests:

NUMBER OF BYTES..

The user then enters the number of memory bytes to Fill. Up to 3 HEX digits are valid.

The Monitor then requests:

FILL BYTE..

The user enters the byte (up to 2 HEX digits) to fill the selected memory with.

"G" The Go command allows the user to execute a program in memory at full speed with an optional breakpoint. The program will continue to execute until the breakpoint is encountered, a key is pressed, or until a RST 7 (FFH) instruction is executed.

When the Go command is issued the Monitor responds with:

STARTING ADDRESS..

The user then enters the starting address of the program to be executed. Up to 4 HEX digits are valid. If only the <ENTER> key is pressed the program will start to execute from the current contents of the Program Counter.

The Monitor then requests:

BREAKPOINT ADDRESS..

The user then enters the address (up to 4 HEX digits) of where the user wishes the execution to stop.
Note: If the program execution never reaches the breakpoint address or the breakpoint address is not that of an op code the program will not stop at the breakpoint address. If the <ENTER> key is pressed without entering the breakpoint address, no breakpoint is set.

"H" The Hex/Dec math command provides the user with the sum and difference of two numbers. This command allows the user to declare the base of the numbers that will be entered as either Hexadecimal or Decimal.

When the Hex/Dec command is issued, the monitor responds with:

DEC OR HEX (D/H)...

After a D (for Decimal) or an H (for Hexadecimal) is entered, the monitor responds with:

1ST NUMBER...

Enter the number, up to 4 digits for HEX or 5 digits for DEC, in the chosen base. The monitor then responds with:

2ND NUMBER...

Enter the second number, up to 4 digits for HEX or 5 digits for DEC, in the chosen base. The monitor will respond with:

HEX SUM = xxxx DEC SUM = xxxx SIGNED DEC SUM = xxxx
HEX DIFF = xxxx DEC DIFF = xxxx SIGNED DEC DIFF = xxxx

where xxxx is the result of the math operation. **NOTE:** To convert a number from one base to another, choose 0 as the response to the second number prompt.

"I" The Input command allows the user to Input the contents of an I/O port. The I/O address is an 8 bit HEX value and the content is also an 8 bit value displayed in HEX.

When the Input command is issued the Monitor responds with:

I/O PORT ADDRESS..

The user then enters the I/O address of the port to be read from. Up to 2 HEX digits are valid.

After entering the I/O port address the Monitor will respond with:

CONTENTS OF THE I/O PORT IS .. xx

where xx represents the contents of the specified I/O port displayed as an 8 bit hex value.

"L" The List command allows the user to view 16 machine language instructions beginning at any address in memory. The information that will be displayed, is the memory address of the instruction, its op code and then the mnemonic. All numbers displayed are in hex.

When the List command is issued the Monitor responds with:

STARTING ADDRESS..

The user then enters the starting address (up to 4 HEX digits are valid) of the section of memory to be Listed.

The 16 machine language instructions will be listed followed by the message...

ESC TO QUIT, ANY KEY TO CONTINUE.

If ESC is pressed the command will be aborted. If any other key is pressed, another 16 lines of instructions will be listed.

"M" The Move command allows the user to Move the memory contents of any given section of memory to another memory location. The source memory contents are left intact.

When the Move command is issued the Monitor responds with:

SOURCE ADDRESS..

The user then enters the source address of the section of memory to be moved. Up to 4 HEX digits are valid. After entering the starting address the Monitor then asks:

DESTINATION ADDRESS..

The user then enters the destination address of the section of memory to Move the source contents to. After entering the destination address (up to 4 HEX digits) the Monitor then prompts for the:

NO. OF BYTES..

The user then enters the number of memory bytes to Move. Up to 3 HEX digits are valid and this number is a HEX value.

"O" The Output command allows the user to output an 8 bit HEX value to a specified I/O port address. The I/O port address is entered as an 8 bit HEX value.

When the Output command is issued the Monitor responds with:

I/O PORT ADDRESS..

The user then enters the I/O address of the port to be written to. Up to 2 HEX digits are valid.

The Monitor will respond with:

I/O DATA BYTE..

The user then enters the 8 bit HEX value to be written to the specified I/O port address.

"P" This will allow you to choose to write to the parallel printer port all subsequent characters that are displayed on the terminal.

When the Printer command is issued the Monitor responds with:

TURN PRINTER ON? (Y/N)..

If the user types "Y" then the above monitor response and all characters that are displayed to the terminal will also be written to the parallel printer port, otherwise if any other character is typed, characters will only be displayed on the terminal.

"R" The Register command displays the current contents of all accessible CPU registers, the flags, the value on the top of the stack and the op code pointed to by the Program Counter and the mnemonic of that op code.

"S" The MOS Service call allows the user to access EMOS Services without having to execute a CALL instruction.

When the Service command is issued, the monitor prompts the user for:

SERVICE NUMBER ..

For a listing of the available service calls, consult the section on EMOS services in this manual.
NOTE: The input parameters must be placed in the appropriate registers, before executing the S command. The C register is automatically loaded with the service number.

"T" The Trace command executes a single instruction and displays the register contents after each instruction.
Note: Since the Trace command is performed through software, operand fetches are not shown.

When the Trace command is issued the Monitor prompts the user for:

NUMBER OF INSTRUCTIONS..

The user enters up to a 2 HEX digit, value. This value determines the number of instructions executed. After each executed instruction the register contents are displayed. If the user does not enter the number of steps, but hits the <ENTER> key, then 1 instruction is executed.
NOTE: Instructions in EPROM cannot be traced so if a service call is traced, the service call is executed at full speed and, upon returning to the calling program, the Trace command resumes execution.

"W" The Write command allows the user to write the contents of memory to a RAMDISK block (or blocks).

When the Write command is issued the Monitor responds with:

SOURCE ADDRESS ...

Enter the starting memory address (up to 4 HEX digits) of the data to be copied.

The Monitor then responds with:

STARTING BLOCK ...

Enter the block in RAMDISK (up to 3 DEC digits) where the data is to be stored.

The Monitor then responds with:

NUMBER OF BLOCKS ...

Enter the number of blocks (up to 3 DEC digits) that will be needed to store the data.

If (STARTING BLOCK + NUMBER OF BLOCKS) is greater than 255 the error message

ERROR BLOCK OUT OF RANGE!

will be displayed, no Write action will be done, and the original prompt will return.
REMEMBER: The size of a block is 256 bytes.

">" The Hex upload command loads data from the host to the trainer in Intel HEX format.

When the upload command is issued, the Monitor responds with:

STARTING ADDRESS ...

This is the memory address in the trainer where the data is to be loaded (up to 4 HEX digits.)
The Monitor responds with:

READY TO RECEIVE, <ESC> TO ABORT

The transfer of data can be stopped by pressing <ESC>.

"<" The Hex Download command saves data from the trainer to the host in Intel HEX format.

When the Download command is issued, the Monitor responds with:

STARTING ADDRESS ...

This is the starting memory address (in the trainer) of the data to be saved (up to 4 HEX digits).

The Monitor then responds with:

LAST ADDRESS ...

This is the ending address of the data to be saved (up to 4 HEX digits).

The Monitor then responds with:

OFFSET ADDRESS ...

The offset address is normally entered with the same address used for the starting address

prompt. To begin downloading, press the <SPACE> key. **NOTE:** This command is to be used in conjunction with a communication package running on a PC.

"?" The Help command displays the help menu which contains the Monitor commands and a short description of each command. This is the same menu that is shown when EMOS is first invoked.

MOS SERVICES

The Monitor Operating System allows the user to access MOS services with a single function (subroutine) CALL to address 1000 Hex. In addition, all registers that are not used as input or output to the service are preserved. NOTE: Services that are underlined incorporate the COM1 and COM2 RS232 serial ports. In order to make use of these services a PC/Terminal device should be connected to the COM1 or COM2 serial connector. The services are as follows:

SERVICE 0	**DEMO**	Demonstration; this service routine sends a pitch of increasing frequency to the speaker while flashing the output LEDs in conjunction with the pitch.
	INPUT	REGISTER C: 0
	OUTPUT	NONE
SERVICE 1	**CONIN**	Console input; this service waits for a character from the communication port selected by register B. Note that the RST 5.5 interrupt is disabled until the service terminates.
	INPUT	REGISTER C: 1 REGISTER B: Communication port number 1 or 2.
	OUTPUT	REGISTER L: ASCII character returned from keyboard.
SERVICE 2	**CONSTAT**	Console input status; this examines the communication port selected by register B and returns a 0FFH if a character is ready, otherwise a 00H.
	INPUT	REGISTER C: 2 REGISTER B: Communication port number 1 or 2.
	OUTPUT	REGISTER L: Console status.
SERVICE 3	**CONOUT**	Console output; this service outputs a ASCII character to the communication port selected by register B.
	INPUT	REGISTER C: 3 REGISTER B: Communication port number 1 or 2. REGISTER E: ASCII character.
	OUTPUT	NONE
SERVICE 4	**PSTRING**	Print string; this service sends a string of characters to the communication port selected by register B. The string of ASCII characters starting at the address in the D/E register pair will be sent out until a "$" is encountered. The "$" is not printed and D/E is returned pointing to the character after the "$".
	INPUT	REGISTER C: 4 REGISTER B: Communication port number 1 or 2. REGISTER PAIR DE: Pointer to the start of the string.
	OUTPUT	REGISTER PAIR DE: This will point to the character after the "$".
SERVICE 5	**UPRINT**	Unsigned print; this service prints a 16 bit number in decimal, without use of sign, to the communication port selected by register B.
	INPUT	REGISTER C: 5 REGISTER B: Communication port number 1 or 2. REGISTER PAIR DE: 16 bit unsigned number to print.
	OUTPUT	NONE

SERVICE 6	**SPRINT**	Signed print; this service prints a 16 bit number in decimal with use of sign (2's complement), to the communication port selected by register B.
	INPUT	REGISTER C: 6 REGISTER B: Communication port number 1 or 2. REGISTER PAIR DE: 16 bit signed number to print.
	OUTPUT	NONE
SERVICE 7	**MULT**	Multiply; this service multiplies HL by DE and returns the result in HL and DE.
	INPUT	REGISTER C: 7 REGISTER PAIR DE: 16 bit multiplier REGISTER PAIR HL: 16 bit multiplicand
	OUTPUT	REGISTER PAIR HL: Upper 16 bits of product. REGISTER PAIR DE: Lower 16 bits of product
SERVICE 8	**DIV**	Divide; this service divides HL by DE.
	INPUT	REGISTER C: 8 REGISTER PAIR HL: 16 bit dividend. REGISTER PAIR DE: 16 bit divisor.
	OUTPUT	REGISTER PAIR HL: 16 bit quotient. REGISTER PAIR DE: 16 bit remainder.
SERVICE 9	**ADCIN**	Analog to Digital input; this service reads a 8 bit value from a selected A/D channel.
	INPUT	REGISTER C: 9 REGISTER E: Selected channel number 0, 1, 2 or 3.
	OUTPUT	REGISTER L: 8 bit analog conversion.
SERVICE A	**DIPSWIN**	DIP switch input; this service reads the current switch positions of the 8 position DIP switch.
	INPUT	REGISTER C: A
	OUTPUT	REGISTER L: DIP switch value.
SERVICE B	**PTBIN**	Port B input; this service reads the contents of the digital input port B and returns its complement in L. This is similar to service A.
	INPUT	REGISTER C: B
	OUTPUT	REGISTER L: Complement of 8 bit port B value.
SERVICE C	**PTAOUT**	Port A output; this service writes to the digital output port A.
	INPUT	REGISTER C: C REGISTER E: 8 bit value to write to port A (this port is equipped with 8 status LED's).
	OUTPUT	NONE
SERVICE D	**HEXPRINT**	Hex print; This service prints to the communication port selected by register B, the value in DE in base 16 (HEX). 4 HEX digits are printed.
	INPUT	REGISTER C: D REGISTER B: Communication port number 1 or 2. REGISTER PAIR DE: 16 bit unsigned number to print.
	OUTPUT	NONE

SERVICE E **DACOUT** Digital to Analog Converter output; This service routine outputs a 8 bit number in the E register to the Digital to Analog converter.

INPUT REGISTER C: E
REGISTER E: 8 bit value to output to DAC.

OUTPUT NONE

SERVICE 10 **PITCH** Pitch output; This service sends the 16 bit count in the DE register pair to the speaker timer (8253 timer 1). The larger the number the lower the pitch. If the DE register pair = 0, then the speaker tone is turned off.

INPUT REGISTER C: 10
REGISTER PAIR DE: 16 bit pitch value.

OUTPUT NONE

SERVICE 11 **LEDOUT** LED Display output; This service routine displays a ASCII character in the E register to the LED alphanumeric display at the character position specified in the D register.

INPUT REGISTER C: 11
REGISTER E: ASCII character to display.
REGISTER D: Character position (positions are numbered 0 to 7 from left to right).

OUTPUT NONE

SERVICE 12 **LEDHEX** LED Hexadecimal output; This service routine displays a number in the DE register pair in HEX, on the left 4 LED alphanumeric displays.

INPUT REGISTER C: 12
REGISTER PAIR DE: 16 bit number to be displayed in HEX.

OUTPUT NONE

SERVICE 13 **LEDDEC** LED Decimal output; This service displays a number in the DE register pair in decimal, left justified on the LED alphanumeric display (maximum decimal value is 9999).

INPUT REGISTER C: 13
REGISTER PAIR DE: number to be displayed in Decimal.

OUTPUT NONE

SERVICE 14 **DELAY** Delay according to the value of the HL register pair. The larger the value, the longer the delay.

INPUT REGISTER C: 14
REGISTER PAIR HL: The amount of delay.

OUTPUT NONE

SERVICE 15 **TUNE** Play the tune which is in the string pointed to by DE at the tempo specified by B. The larger the value of B the longer the duration of each tone. If a zero is encountered in the string, then the subroutine terminates and returns DE pointing to the byte after the 0.

INPUT REGISTER C: 15
REGISTER B: Length of each tone.
REGISTER PAIR DE: Pointer to a string of tones.

OUTPUT REGISTER PAIR DE: This is returned pointing to the byte after the 0 in the string.

SERVICE 16 **PRNOUT** Send the character in E to the parallel printer port.

INPUT REGISTER C: 16
REGISTER E: Character to be sent to the printer.

OUTPUT REGISTER L: This will be 0 if no errors occured. If an error occured then the bits in L will be returned indicating the following:

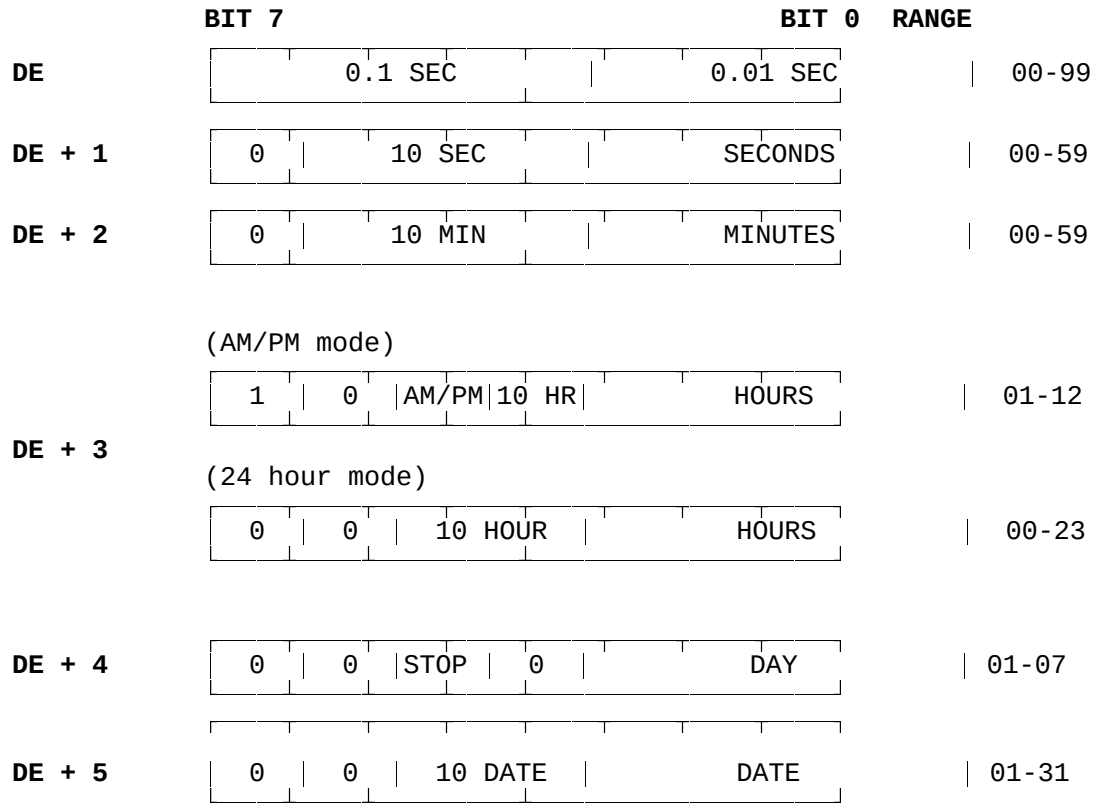
Bit 3 = Indicates a printer error in general, if it is 1.
Bit 4 = Means printer not selected, if 0.
Bit 5 = Indicates printer out of paper, if 1.
Bit 7 = Printer is busy, if 0.

SERVICE 17 **OUT422** Send the byte in E out the RS422 port.
INPUT REGISTER C: 17
REGISTER E: Character to output
OUTPUT NONE

SERVICE 18 **IN422** Get a byte from the RS422 port.
INPUT REGISTER C: 18
OUTPUT REGISTER PAIR HL: If a character has been received, H will be 1 and L will be the character. HL will be 0 if a character has not been received.

SERVICE 19 **KEYIN** Wait for a byte from the keypad and return it in L. Reading the keys starting at the top row and reading from left to right as you go down, the values returned for the first 16 keys will be 00 to 0F hex. The next 3 rows will return 14 to 1F hex.
INPUT REGISTER C: 19
OUTPUT REGISTER L: Numeric value of key pressed

SERVICE 1A **WRSCL** Write 8 bytes of memory, starting from the address in DE, to the optional real time clock. The clock provides timekeeping information in BCD including hundredths of seconds, seconds, minutes, hours, day, date, month and year information. The date at the end of the month is automatically adjusted for months with less than 31 days, including correction for leap years. The real time clock operates in either 24-hour or 12-hour format with an AM/PM indicator. The data pointed to by DE will be stored in the real time clock as follows:



DE + 6	0	0	0	10MTH	MONTH			01-12
DE + 7	10 YEAR				YEAR			00-99

If bit 7 of address DE + 3 is 0 the clock will be in 24 hour mode after WRSCLK is executed. If it is 1 then AM/PM mode is selected and bit 5 of address DE + 3 will select AM or PM (PM is selected if bit 5 is 1). When changing from AM/PM mode to 24 hour mode and vice-versa you must change the hours to match the selected mode. Once the hours are correct, the real time clock will maintain the correct hour for the selected mode.
If bit 5 of address DE + 4 is set to 1 and WRSCL is executed, the real time clock will be stopped. The clock may be restarted by resetting the bit to 0 and executing WRSCL.

INPUT REGISTER C: 18
REGISTER PAIR DE: Address of the first of 8 bytes to be written to the real time clock.
OUTPUT NONE

SERVICE 1B RDSCL Read 8 bytes of data from the optional real time clock and store them in the 8 consecutive bytes starting at the address in the DE register pair. The 8 bytes are formatted the same as the data passed to WRSCL.

INPUT REGISTER C: 19
REGISTER PAIR DE: Starting address to store the 8 bytes read from the real time clock.
OUTPUT NONE

EXAMPLE: OUTPUT THE HEX NUMBER 1234 TO THE LED DISPLAY

```
START:  MVI   C,12H      ; LEDHEX SERVICE ROUTINE
        LXI   D,1234H    ; LOAD REG PAIR D/E WITH HEX VALUE 1234
        CALL  1000H      ; CALL MOS FOR LEDHEX SERVICE
        HLT              ; HALT PROGRAM AND FREEZE DISPLAY. NORMALLY
                         ; PROGRAMS TERMINATE WITH A RST 7 INSTRUCTION (FF)
                         ; WHICH WOULD OVERWRITE DISPLAY.
```

MACHINE LANGUAGE

```
8F01  0E    ; MVI   C,12H
8F02  12
8503  11    ; LXI   D,1234H
8F04  34
8F05  12
8F06  CD    ; CALL  1000H
8F07  00
8F08  10
8F09  76    ; HLT
```

TERMINATING PROGRAMS

Each user program should end with the RST7 (FFH) software interrupt instruction. This instruction allows the Monitor Operating System to regain control of the CPU.

The Monitor Operating System also uses the RST7 instruction to implement a breakpoint when using the RUN key. If additional breakpoints are required they can be hand inserted at the appropriate spots in the user's program. It is the user's responsibility to remove these breakpoints when through.

Software Single Stepping and Breakpoints

Single Stepping and Breakpoints make debugging software and hardware easier. Users can use these tools to determine if a program is functioning correctly and if not where the problem is located. Software Single Stepping and Breakpoints are most useful when the hardware is functioning properly and a program needs debugging. These software tools provide us with easier debugging and more information then their hardware counterparts. For example registers and memory contents may examined and changed, and addresses, data, and register contents are all displayed in HEX rather than binary. The hardware status LEDs display the contents of the address, data, and control busses all in binary. The Hardware Single Stepper and Breakpoints have couple advantages of their own. First, if the hardware is not working properly the hardware single stepper can be used to locate at what address the processor goes awry and what state the bus and address lines were in at that time. The hardware single stepper also can provide us with operand fetches as well as instruction fetches and memory accesses. Second, the hardware breakpoint can break at any address, even addresses in EPROM, and is not limited to instruction fetches as is the software breakpoint.

The Software Single Step is implemented as its name states through software. When the STP key is pressed, the instruction at the current PC address is executed. This is accomplished by the Monitor Operating System (MOS) replacing the byte prior to the instruction with an EI (FB hex) instruction. The MOS program then unmasks the RST 6.5 interrupt. This interrupt should be jumpered to VCC (JP3) so as to always produce an interrupt when the 6.5 interrupt is enabled and unmasked. The 8085 microprocessor always executes one instruction after the EI instruction before enabling interrupts. That one instruction is the instruction that is to be single stepped. After the EI instruction is inserted prior to the single step instruction, the MOS program jumps to the EI instruction. After the EI instruction executes, the single step instruction is executed before the RST 6.5 interrupt is acknowledged. When the RST 6.5 interrupt is acknowledged the execution branches back to the MOS program. The MOS program then replaces the byte back at its original address overwriting the EI instruction. The current user PC address is found by POPing the return address from the interrupt off the stack. Since software single stepping requires the byte prior to the instruction to be replaced with an EI instruction, software single stepping can only take place in RAM. When software single stepping into a MOS Service CALL, which are located in EPROM, the processor will execute the service CALL full speed and break at the instruction immediately following the service CALL. Software single stepping, steps an instruction at a time, so no operand fetches or memory writes are seen.

The Software Break allows the user to run a program full speed and break (stop) the program at a user specified address. The address is entered from the keypad in conjunction with SFT BRK key. The software break is performed by the MOS program using the RST 7 software interrupt. When the user specifies a break address the Monitor Operating System replaces the opcode at this address with RST 7 (FF hex) instruction. The users program can then execute full speed until encountering the RST 7 instruction. When the users program reaches the break, the RST 7 instruction returns control to the MOS program. The MOS program then restores the opcode that was replaced by the RST 7 instruction. The MOS program only allows the use of one breakpoint, which is automatically reset to 0000 after the breakpoint has been encountered. The user can however, hand insert breakpoints by placing RST 7 instructions at strategic locations throughout the users program. Remember that any hand inserted breakpoints must also cleared by hand. When inserting a software breakpoint it is important to remember that if the program execution never reaches the breakpoint address or the breakpoint address is not that of an opcode, the program will not stop at the breakpoint address. Also any address specified that references addresses in EPROM (addresses below 8000 HEX) will not stop execution, even if the opcode at that address is executed.

Hardware Single Stepping and Breakpoints

To facilitate software/hardware inspection and trouble shooting, the trainer has a hardware single step system built in. Upon trainer power up, the Single Step circuit is disabled, and the CPU runs full speed. Should an application program need debugging, the single step system can be enabled by the HSS EN button, on PB2. The single step circuit gives a choice of stopping on all READS and all WRITES, pushbutton ALL R/W or stopping on INSTRUCTION FETCHES only, pushbutton INS FET. The ALL R/W button causes the single stepper to step at all fetches and memory writes while the INS FET button only step on instruction fetches much like the software single step. The POWER ON RESET and the CPU RESET are separate, so the CPU may be reset, without disturbing the single stepper settings, so all CPU addresses can be viewed from addresses 0000 on up. Unlike the software single stepper, where the processor is always executing MOS or user instructions, the hardware single stepper utilizes wait states in order to single step the processor. Thus the microprocessor is not executing any instructions (is idle) between hardware steps. The Single Stepper can be turned on or off at any time through the use of the HSS EN key. Note: the user should be aware if the single step enable LED is lit (hardware single stepper is enabled), as this will prevent the processor from executing.

The Hardware Breakpoint on the Universal Trainer is comprised of two 8 bit latches and an arming circuit. The Monitor Operating System loads these latches and arms the arming circuit when the HRD BRK and ENT keys are pressed. The address entered from the keypad is split and sent to the high (01 hex) and low (00 hex) address latches, after which the arming circuit (02 hex) is armed. When armed, as the trainer CPU accesses each address, whether it is an instruction or data, the addresses are compared to the latched address. When a match occurs, the comparator instantly engages (turns on) the hardware single stepper system. Simultaneously, the comparator disarms itself. With the hardware single stepper now ON, the CPU is stopped at this address, and the code can now be examined step by step by pushing the HSS button. Full speed execution can be continued by disabling the hardware single stepper. After a hardware breakpoint occurs the arming circuit is automatically disarmed. MOS however still keeps the break address in memory until another breakpoint is set or the system is reset. To rearm the previous breakpoint simply press the HRD BRK key followed by the ENT key. Note: if a hardware breakpoint is set and armed, examining memory using the ENT key or the DEC key will stop the processor if the breakpoint address is examined.

The TRAP key

A program that is running without breakpoints can be terminated by pressing the TRAP key. When the key is pressed, either MOS or EMOS (depending on which monitor started the program) will regain control, allowing you to examine the values of the registers and set software or hardware breakpoints.

USING INTERRUPTS

The Universal Trainer has 13 interrupts available to the user. Five are standard on the 8085A CPU and eight are available through the 8259a Multi-level interrupt controller chip. When utilizing interrupts the Single Step feature of the Monitor System may not function correctly. It is recommended when using interrupts to run at full speed for predictable results.

The 8085A CPU has five (5) on-chip hardware restart/interrupts. These are general purpose interrupts found on all 8085 CPU chips. This listing includes the trainer subsystems to which each interrupt is wired.

1. TRAP ---------- PUSHBUTTON

The TRAP interrupt is a non-maskable interrupt and is the highest priority interrupt. It is positive edge and level sensitive. The trainer system provides a pushbutton on keypad PB1 which allows a program running at full speed to be stopped and control given back to MOS or EMOS without resetting the trainer.

2. RST7.5 ---------- PUSHBUTTON or TIMER #2

The RST7.5 interrupt is a latched positive edge sensitive restart interrupt. A low to high transition at RST7.5 input (pin 7) sets a flip-flop. An interrupt can be generated by the output of timer #2 or by pressing the INT 7.5 pushbutton.

3. RST6.5 ---------- EXPANSION/Vcc
4. RST5.5 ---------- KEYPAD

The RST6.5 and RST5.5 interrupts are level sensitive only. They respond to the interrupt pin (8 for RST6.5, 9 for RST5.5) when the pin is logic high. The pin must be held high until the CPU can get around to servicing the interrupt or it will be missed. Although the RST6.5 can be jumpered to the Expansion connector, it is normally left tied to +5V via jumper JP3, because MOS uses it for software single stepping. RST5.5 is reserved for use by the LED/KEYPAD controller, and has no jumper options.

The RST interrupts will vector to EPROM and from there they are vectored again to RAM by the MOS. The RAM vectors for RST5.5, RST 6.5 and RST7.5 start at memory addresses FFE5, FFE7 and FFE9 respectively. For example, to vector the RST 5.5 interrupt to a service routine at 90F3h, store the low byte of the address (F3h) at FFE5 and the high byte (90h) at FFE6. Remember that MOS uses RST6.5 for software single stepping, so do not change the data at FFE7 or FFE8 if you want to use single stepping.

The RST7.5, RST6.5, and RST5.5 interrupts are software maskable, which means they can be enabled/disabled by software, independently or in unison.

5. INTERRUPT (general 8080 interrupt) --- CONTROLLER

The remaining interrupt pin is the true interrupt system input pin, and it works the same as in the 8080 CPU system, which the 8085A is the offspring of. This interrupt can also be enabled and disabled by software. When a high is seen on the INTR (pin 10 of the CPU), the CPU issues an interrupt acknowledge, signal INTA*, which signals the interrupting device to place an interrupt vector address on the Data bus. In the EMAC trainer, this is accomplished with the use of the 8259A Multi-level interrupt controller chip. It has eight (8) interrupt request inputs, and upon receipt of an interrupt request, it generates an INTR to the CPU, and handles all INTA* handshaking cycles. The user must program the 8259A to accept level or edge sensitive active high interrupt requests, to resolve interrupt priority, end of interrupt type, and the interrupt vectors passed for each interrupt.

The 8259 Interrupt Controller pin assignments are as follows :

6.	**IRQ0**	**Expansion Connector ext. interrupt or Pushbutton**
7.	**IRQ1**	**Communication port 1 Receive Ready**
8.	**IRQ2**	**Com. port 1 Transmit Ready**
9.	**IRQ3**	**Com. port 2 Receive Ready**
10.	**IRQ4**	**Com. port 2 Transmit Ready**
11.	**IRQ5**	**Timer 0 (prescaler)**
12.	**IRQ6**	**Timer 2**
13.	**IRQ7**	**A/D Conversion Complete**

The trainer provides pushbuttons (PB1) to allow manual triggering of various interrupts. PB1 is a four station keypad switch, that is connected to CPU RESET, interrupt RST7.5, TRAP, and interrupt controller IRQ0. All pushbutton interrupts are debounced with an R-C network and Schmitt Trigger invertor gate. IRQ0 is shared with the expansion connector input INT0. Jumper JP25 inhibits the external input to allow the pushbutton interrupt to function independently.

EMAC TRAINER BOARD AND CIRCUIT DESCRIPTIONS

The EMAC UNIVERSAL TRAINER is an 8085 based CPU system, utilizing the INTEL 8085A microprocessor. The trainer is a fully functional industrial control board, but with certain embellishments to allow the student to easily probe the inner workings of an operational microprocessor system.

The board has been laid out to provide easy access to the DATA, ADDRESS, and CONTROL busses. The trainer features analog as well as digital I/O, and it also has many connectors to permit installation of user constructed prototype and expansion cards.

The trainer incorporates many jumpers to select options such as memory configuration, communication baud rates, etc. and also to provide simulated hardware faults. The instructor can set these fault programming jumpers to create circuit " failures " the user must find in order to hone his trouble shooting skills.

The trainer has many subsystems on it that allow the user to " get his feet wet " on many types of circuit design methods. Among them are :

1. Switching power supplies
2. Linear power supply regulators
3. Local/expansion data busses
4. Memory Mapped I/O and I/O mapped I/O
5. Analog and Digital I/O.
6. Multi-sourced interrupts.

The trainer has a number of Input/Output devices on it as well. Among them are:

1. 2 RS-232 serial ports, one of which is optional on the military version
2. 1 RS-422/485 serial port
3. Keypad and alphanumeric LED display
4. 4 channel, 8 bit analog to digital converter
5. 1 Parallel printer port
6. 3 channel Timer/Counter (with speaker on one channel)
7. Digital I/O with status LED's
8. 8 level Interrupt Controller

The trainer also includes diagnostic hardware not normally found in commercial CPU systems, but very useful for trouble shooting purposes :

1. 8 Digit LED display (Instruction address, Opcode)
2. 24 key Keypad (Commands, Data entry)
3. Bus status LED's
4. Hardware single stepper
5. Hardware Breakpoint

In addition, the trainer has connectors for bus expansion, serial and parallel communication, analog signal conditioning, and power.

POWER REQUIREMENTS

The unit operates from 12 VDC, provided by a wall mount power supply, or it can be powered from any other suitable DC power supply such as a 12v battery, etc. Power requirements are 12 VDC +/- 10% at approximately 1 Ampere. DC Power is applied to the UNIVERSAL TRAINER through a miniature jack. The plug that fits this jack should supply +DC volts at the tip and ground on the shaft of the plug. The power entering the board is fused through a 2 amp fast blow (2AG) fuse. No danger of shock exists anywhere on the circuit board.

GENERAL BOARD REQUIREMENTS

The trainer is composed of many logical sections, tied together by the CPU system, except for the power supply, which services all areas. The trainer's sub-systems will be described in the order they appear in the schematics section. It may be helpful to look at each schematic page as each description is read.

POWER SUPPLY SECTION (Schematic Sheet #1):

The trainer is powered from a D.C. supply input of 12 VDC, applied via power jack J1. A diode (D1) prevents damage to the trainer unit should the DC power input be inadvertently reversed. Fuse size F1 protects the power source and trainer should there be a major short or overload present on the trainer or peripherals. Switch S1 turns power on and off. The power supply provides 5 volts for the logic, via an LM7805 series pass linear regulator chip. As there is a large input vs. output voltage gradient (7 volts, on the average.), the 5 volt regulator will dissipate (waste) a lot of power. Therefore, in order to keep dissipation to a minimum, all logic system components are CMOS when possible. As CMOS devices use less current per device than their NMOS or TTL counterparts, the total drain from the 5 volt supply is far lower than with the other logic family parts, therefore the 5 volt regulator will have a much easier life. The regulator, even with CMOS components, has a fairly heavy load and will become hot to the touch.

An 8 volt DC supply is provided for the LED's via an LM7808 linear regulator. In this case, although the current is fairly high, the voltage drop is only 4 volts, so the dissipation of a linear regulator in this application is also tolerable. From the 8 volt supply some power is delivered to the Analog and Serial Communication systems. The Analog and Communication systems require a negative supply voltage in addition to the positive voltage. An 8 volt " INVERTING SWITCHER ", using a specialized power supply chip (TL497) and an inductor, supplies a -8 volt supply rail for the Analog and Communication systems. As the negative supply operates at less than 100 Ma. output, heat sinking is not required.

CPU SECTION (Schematic Sheet #2):

The CPU section of the trainer consists of the 80C85A (CMOS 8085) CPU chip, memory chips, address and control bus drivers, Memory and I/O decoders, and miscellaneous glue logic. The CPU is the heart of the system. It fetches instructions from the memory, (opcodes) and executes the instructions. The CPU generates almost all the timing and control of the trainer system. All CPU timing is derived from the master clock on the CPU chip, which is set at 6.144 MHZ by crystal Y1. The 6.144 MHZ oscillator signal is internally divided by two (2) to provide the internal two phase clock used by the CPU. The 8085 has a clock output pin (37) that is wired through control bus buffer U15, and this signal is used as the SYSTEM CLOCK, hereafter labeled SYSCLK. The SYSCLK frequency is 3.072 MHZ, which divides nicely down through a counter chain to the communication baud rate frequencies, which is why such an oddball CPU clock frequency was chosen. Each 8085 instruction clock period is one of these 3.072 MHZ clock periods. As defined by the 8085 CPU hardware manual, each SYSCLK period defines what is referred to as a " T " state. Therefore each T state is 1/3072000 of a second, or 325 nanoseconds per state.

The CPU is reset on power up by a simple R-C reset circuit. A RESET pushbutton is also provided. A schmitt trigger invertor is used to " square up " the reset signal, and pass it off to the CPU. The I/O decoder selects the various I/O and memory chips as the CPU addresses them. All devices on the trainer except the CPU memory are always present in the I/O Map. In some systems, however, I/O devices are placed in the

MEMORY MAP instead, to simplify decoding or to make use of the faster memory instructions. To experience memory mapped I/O, a jumper (JP8) will swap out the last 8K of EPROM memory, and place the I/O devices there (Memory Mapped I/O). The I/O devices will also remain in the I/O map, to allow the trainer's Monitor Operating System to " BOOT UP ".

Memory Map:

EPROM Space: address 0000 hex to 7FFF hex, jumper JP4 selects usage of 16k or 32k EPROMS.

RAM Space: address 8000 hex to FFFF hex, jumper JP5 selects usage of 8K or 32K RAMS.

(addresses 6000 to 7FFF hex are sacrificed when Memory Mapped I/O is selected)

I/O Device Map:

Device	I/O address(s)	Memory address(s)
Hardware Breakpoint	**00 to 02 hex**	**6000 to 6002 hex**
Low Address	00	6000
High Address	01	6001
Arming Circuit	02	6002
8259 Interrupt controller	**10 to 11 hex**	**6010 to 6011 hex**
8253 Timer	**20 to 23 hex**	**6020 to 6023 hex**
Timer #0 (8259 IR5)	20	6020
Timer #1 (speaker)	21	6021
Timer #2 (RST 7.5)	22	6022
Timer Control	23	6023
Parallel Printer	**30 to 31 hex**	**6030 to 6031 hex**
Printer Data	30	6030
Printer Control Lines	31	6031
8255 PPI Port	**40 to 43 hex**	**6040 to 6043 hex**
Port A (LEDs)	40	6040
Port B (DIP switches)	41	6041
Port C	42	6042
PPI Control	43	6043
8279 LED Display/Keypad	**50 to 51 hex**	**6050 to 6051 hex**
Display/Keypad Data	50	6050
Display/Keypad Control	51	6051
RAMDISK	**60 to 80 hex**	**6060 to 6080 hex**
RAMDISK Low Address	60	6060
RAMDISK High Address	70	6070
RAMDISK Data	80	6080
Analog I/O	**90 to B0 hex**	**6090 to 60B0 hex**
A/D Sample & Hold	90	6090
A/D Select	A0	60A0
D/A select	B0	60B0
8251 Comport 1	**C0 to C1 hex**	**60C0 to 60C1 hex**
COM1 Serial Data	C0	60C0
COM1 Serial Control	C1	60C1
8251 Comport 2	**D0 to D1 hex**	**60D0 to 60D1 hex**
COM2 Serial Data	D0	60D0
COM2 Serial Control	D1	60D1
External Free I/O	**E0 to FF hex**	**60E0 to 60FF hex**

External Hardware Reset

PB1B (PushButton 1B) is the system reset. Fault jumper JP20 is provided to place the CPU in perpetual reset. With JP20 in the normal operation position, a reset can be generated by pushing PB1B, an external reset may be initiated by connecting a signal lead to the open pin of JP20 from a switch, relay or logic device to pull the pin to ground. If a logic device is used, it must be able to safely sink 13 ma. of current.

BUS EXPANSION PORT (Schematic Sheet #3):

The bus Expansion Port allows the user to connect additional hardware to the trainer. It consists of a 50 pin header and data bus driver. The 50 pin header has a standard EPAC series connector footprint that permits EPAC peripheral cards to directly plug onto the trainer's data, address, control, and power buses. User circuits constructed on a wire wrap board or protoboard can also be interfaced using this connector.

The bus expansion port Data Bus lines (D0-D7) are buffered with a 74HC245 bi-directional bus driver chip. This driver isolates the LOCAL BUS components (CPU, Memory, Interrupt, PPI, etc.) from the EXTERNAL BUS. Problems with externally connected peripherals are thus less likely to interfere with the bootup of the trainer's monitor program. The address and control lines of the CPU system are uni-directional (always outputs), and are already buffered at the CPU area. Dead shorts, however, on these lines will still adversely affect the trainer bootup. The Expansion Bus driver is selected by the decoder hardware at I/O addresses E0 hex through FF hex, and optionally Memory addresses 60E0 through 60FF hex. CPU signal RD NOT changes direction of the bus driver to read in data. The bus driver normally drives the Data Bus out to the connector (RD NOT inactive). No special programming is required to use the Expansion Bus, although the I/O address of any device connected to it must be E0 to FF hex. For information on the Expansion Bus pinout refer to Schematic #3.

SERIAL COMMUNICATIONS (Schematic Sheet #4):

The Universal Trainer has three (3) serial communication ports. Two of them, COM1 and COM2, are full function RS-232 serial communication ports, each with its own DB-25 connector. COM2 is optional on the military version Universal Trainer. Both channels utilize an 82C51A UART (Universal Asynchronous Receiver\Transmitter). Each channel has a hardware jumper to select the Baud Rate it operates at, from 300 to 19,200 Baud. Both channels have RS-232 level to TTL level translator chips, powered by the +8v and -8v power supplies. They both tie into the CPU data, address, and control busses. A counter chain divides the 3.072 MHZ SYSCLK down to the baud rate clocks selected by jumper selectors JP16 and JP17, which then pass the baud clocks to the two UARTs. Each of the two communication ports has DTR (Data Terminal Ready input), RTS (Ready To Send output), and optionally (jumper selectable), CTS (Clear To Send input) leads to provide "Handshaking" when required. Both COM1 and COM2 can be interrupt driven if desired, as RXRDY (Receive Ready) and TXRDY (Transit Ready) pins of the UART chips are all wired to inputs on the 8259A interrupt controller.

In normal use with the Trainer Monitor program running, the keypad and LED display are used for standard input/output. The COM1 communication port however is available for experimentation. COM2 is not populated on the military version of the Universal Trainer but can be purchased as a option.

Communication port COM0 is a specialized serial port for use in RS-422/485 networking. It consists of two (2) type 75C176 RS-422 transceiver chips. One chip is configured to always receive data from the RS-422 loop and is connected to the SID input (Serial input Data, pin 5) of the CPU. The SID input can be considered a direct line to the CPU Accumulator. The state of this line is read by the RIM (Read Interrupt Mask) instruction. The other driver is configured to transmit onto the RS-422 loop, but it can be enabled/disabled by a Port Control bit (Described later in PPI section). The transmit driver broadcasts the state of the SOD line (Serial Output Data pin 4) of The CPU. Like the SID line, SOD is a direct line to the accumulator, and is set via the SIM (Set Interrupt Mask) instruction.

Through software subroutines, the SID and SOD lines make up a software UART. This UART and the RS-

422 drivers make up a complete serial communication port. By connecting the Transmit and Receive ports together, and appropriately toggling the Transmit Control line, a two wire, bi-directional serial RS-422/485 network port is created. This network port can be used to connect and pass data between other trainers, and/or other EMAC hardware. The Baud rate for COM0 is set entirely in software. See Appendix B Assembly Language Drivers.

SCREW TERMINAL 2 AND DB25 COMMUNICATIONS PINOUTS

5 POSITION SCREW TERMINAL 2 (ST2)

TERMINAL 1	RS422/485 RECEIVE CHANNEL B (RXB)
TERMINAL 2	RS422/485 RECEIVE CHANNEL A (RXA)
TERMINAL 3	RS422/485 TRANSMIT CHANNEL B (TXB)
TERMINAL 4	RS422/485 TRANSMIT CHANNEL A (TXA)
TERMINAL 5	GROUND

COM1 AND COM2 DB25 CONNECTORS

PIN 1	CHASSIS GROUND
PIN 2	RS232 TRANSMIT (TXD)
PIN 3	RS232 RECEIVE (RXD)
PIN 5	CLEAR TO SEND (CTS)
PIN 6	DATA SET READY INPUT (DSR)
PIN 7	SYSTEM GROUND
PIN 20	DATA TERMINAL READY OUTPUT (DTR)

RAMDISK (Schematic Sheet #5)

The trainer has the ability to store data or programs in a separate memory area called a RAMDISK. A RAMDISK is comprised of Static RAM chips, plugged into a special socket that has a battery built into it. With the system power off, the battery supplies the power to keep the RAMs alive, thereby preserving data storage. The battery sockets have a CMOS power monitor chip in them that disconnects the RAMs from Power and Chip select lines during power down, placing the CMOS RAMs in micro-power, data retention mode.

During normal operation with power on, the RAMDISK is accessed through the I/O map, with software drivers loading LOW and HIGH byte addresses to latches, which provide a 16 bit linear address to the RAMDISK chips. The data stored is then read or written one byte at a time to the address pointed to by the

latches. This data is then written to or read from the active system memory. Although this is a relatively slow process, it is not much unlike the accesses made to a floppy disk drive found in larger systems. The RAMDISK is not intended for on-line fetch and execution of machine code, rather it is used to store programs or data for future reference. A user can store programs in RAMDISK at a particular address and later retrieve those stored programs back into main memory where they can then be executed. Note the RAMDISK, by virtue of I/O accessibility, does not take up system memory space, and is not as likely to get written over if a program crashes.

The Trainer has two (2) sockets that are jumper configurable to accept 8Kx8 RAMs, 32Kx8 RAMS, or 32Kx8 EPROMS for up to 64K of RAMDISK (see jumper descriptions).

ANALOG INPUT (Schematic Sheet #6):

Many microprocessor systems are assigned the task of analog data acquisition and control. Typical applications involve reading a " linear " sensor input, such as a Thermocouple voltage, a Thermistor's resistance, Photodiode voltage, etc., then rendering a digital code corresponding to that input. In most cases, the actual sensing element cannot interface directly with a microprocessor, but must be " signal conditioned " to do so. Real world sensing elements often put out millivolt sized signals, change resistance, change capacitance, or output a tiny current. In order to allow a microprocessor system to measure and respond to environmental stimuli such as temperature, pressure, amperage, voltage, light, etc., special analog circuitry is used to convert (condition) these signals into a calibrated voltage, then the voltage is measured and converted to a digital code, for presentation to the microprocessor.

A specialized type of circuit called an A/D converter (Analog to Digital Converter), hereafter referred to as an ADC, will produce a digital code output for a specific analog voltage input. Again, even the types of ADCs are as widely variant as the signals measured. The ADC used in the trainer " reads " a D.C. analog voltage in the range of one (1) to six (6) volts, and produces an eight (8) bit digital code corresponding to the input voltage. The CPU can access the ADC whenever the software needs to read a " sensor " reading. The trainer has an analog multiplexor that switches the single analog converter input to one of four (4) inputs. Each input comes from a screwdriver adjustable potentiometer that can be set as required for experimentation. The Pots can be switched out, and external voltages fed to the trainer board, via a screw terminal block provided. Additionally, an analog signal conditioning card may be plugged into the trainer at connector CN6, allowing experimentation with actual sensors.

The Conditioning Card Slot (CN6) is a special connector wired to fit standard EMAC analog conditioning cards, or the user may prototype a card for this slot. The Card slot handles four (4) sensor conditioning circuits.

The trainer's A/D Converter circuit has a Sample/Hold circuit which provides for fast signal digitization (sampling).

Using the Analog to Digital Converter

The analog to digital conversion has 8 bit resolution and the conversion time to convert a selected channel is 100 microseconds. An additional settling time of 30 microseconds is required for the sample and hold circuit. Analog inputs are 1 to 6 Vdc and each input can safely tolerate from -0.3 to +10 Vdc. Different ranges are possible by using one of the analog signal conditioning cards provided optionally from EMAC. Ranges can also be adjusted slightly by use of the calibration pots. ONLY IN SPECIAL CIRCUMSTANCES WOULD THIS BE UNDERTAKEN, AND EMAC RECOMMENDS A THOROUGH UNDERSTANDING OF THE ANALOG CIRCUITRY BEFORE ATTEMPTING TO ALTER FACTORY SETTINGS.

An analog input of 1 Vdc or less will produce a 00H when converted into a digital code. From 1 Vdc and up, each additional 20 mv will increase the hex count to the CPU by about 1 count to a maximum of HEX count of 0FFH.

Each analog input is fed through an analog multiplexor, a Sample and Hold Circuit, and then into the Analog to Digital Converter. Port C lines C0 and C1 select the desired channel. Once the channel has been selected wait 30 microseconds for the voltage to settle before closing the door on the sample and hold. Outputting to I/O address 90H puts the sample and hold in the hold mode (closes the door). Immediately after holding, output to I/O address 0A0H to start the ADC. Bit 4 of port C, buffers the status ADC's End Of Conversion line. When this line goes low the conversion is finished. This line can be read by the CPU in the same manner as Port B. To read the ADC, input from I/O address 0A0H. Easy access is provided to the ADC through the use of MOS Service Calls.

12 POSITION SCREW TERMINAL 1 (ST1)

TERMINAL 1	ANALOG CHANNEL 0 INPUT
TERMINAL 2	ANALOG CHANNEL 0 COMPENSATING INPUT
TERMINAL 3	ANALOG GROUND
TERMINAL 4	ANALOG CHANNEL 1 INPUT
TERMINAL 5	ANALOG CHANNEL 1 COMPENSATING INPUT
TERMINAL 6	ANALOG CHANNEL 2 INPUT
TERMINAL 7	ANALOG CHANNEL 2 COMPENSATING INPUT
TERMINAL 8	ANALOG GROUND
TERMINAL 9	ANALOG CHANNEL 3 INPUT
TERMINAL 10	ANALOG CHANNEL 3 COMPENSATING INPUT
TERMINAL 11	EXTERNAL +VOLTS
TERMINAL 12	EXTERNAL -VOLTS

LED DISPLAY AND KEYPAD (Schematic Sheet #8):

The trainer has an eight (8) character, fourteen (14) segment per character alphanumeric LED display, for visual output of microprocessor information. There is also a twenty-four (24) key keypad for the entry of data, addresses, and operands. The Keypad and Display unit are driven by an 8279 keyboard/display driver I.C. The 8279 accepts multiplexed keypad inputs, and stores them in an internal memory. It handles all matrix and de-bounce functions. It has a polled and/or interrupt driven hardware interface to the CPU bus. The trainer monitor program in EPROM uses the interrupt driven system for keypad inputs. The 8279 also contains a display memory, that is output to a scanned matrix of LED's that make up the multiplexed LED display, which is refreshed at approximately 100 KHZ.

When a key is pressed the trainer monitor program reads the keypad input through the 8279, and the monitor program then sends data up to the display. This interactive system controlled by the monitor operating system allows the user alter the CPU registers, single-step programs, etc. For a complete description of each key see the Software Section. User programs can easily access the display through the use of MOS Service Calls. If inputting from the keypad from within a program, it is essential that interrupts be disabled (DI instruction) and then reenabled before the program terminates. As another alternative the RST 5.5 interrupt can be masked off (SIM instruction) but again must unmasked before the program terminates.

PARALLEL PRINTER PORT (Schematic Sheet #9):

Included with the EMAC Universal Trainer is a PC compatible style parallel printer port, which allows the user to make hardcopies of programs/data. The port has a female DB-25 connector that directly plugs into PC compatible printers using a standard IBM PC type printer cable. This printer port is essentially a parallel digital I/O port, with special open collector drivers and connector pinouts to conform to Centronix parallel printer standards. See table below for pinouts of this connector. Note a name followed by a "*" denotes active LOW line.

CONNECTOR PIN #	FUNCTION	PRINTER PIN #
1	STROBE*	1
2	D0	2
3	D1	3
4	D2	4
5	D3	5
6	D4	6
7	D5	7
8	D6	8
9	D7	9
10	ACKNOWLEDGE*	10
11	BUSY	11
12	PAPER EMPTY (PE)	12
13	SELECT	13
14	AUTOFEED*	14
15	ERROR*	32
16	INITIALIZE*	31
17	SELECT IN*	36
18-25	GROUND	16,19-30,33

Most parallel printers except standard ASCII codes for each character. Special characters are available on some printers allowing them to print lines and graphs. See the user manual that accompanied your printer for further information.

The Parallel Printer Port is a dedicated parallel digital I/O port with connector configuration and hardware drivers to support PC compatible printer cables and printers. The port normally is an output port with status/handshake lines. The printer port can be configured to an input port by software to allow the polling of hardware " keys " used by certain software programs. Although not needed by the trainer's Monitor Operating System, this capability is included for experimental purposes. The Printer port consists of two octal (8 bit) latches, one for the data output to the printer, and one for control lines to the printer. The data latch handles printer data bits 0 through 7. It is trainer I/O address 30 hex (MMIO 6030 hex). An output data instruction to this address places the CPU data output onto the Parallel Printer Port connector's data pins.

An octal bus receiver/driver at this same address (30 Hex) allows the CPU to read this output data if a Read data instruction is performed at this address. The control latch (address 31 Hex) latches the control strobes to the printer. Their bit status cannot be directly read, so it is advised to keep a " copy " of the control bits written to this port in memory. The printer control bits are as follows: Note a name followed by a "*" denotes active LOW line.

BIT0	STROBE*	(writes the data to printer)
BIT1	AUTOFEED*	(causes printer to double line feed)
BIT2	INITIALIZE*	(causes a local printer RESET)
BIT3	SELECT*	(" chip selects " the printer)
BIT4	SPARE	
BIT5	SPARE	
BIT6	SPARE	
BIT7	READEN	(allows printer port to read in data)

Upon Trainer Hardware RESET, the control port bits are all reset low. As the printer control lines are active low, and these lines pass through inverting drivers (see schematic), the printer port comes up in the INACTIVE state, with the data direction heading OUT. In order to use the printer, the control line named SELECT must have a one written to it. (08 H to addr. 31 H) Data to the printer may then be written to the data latch (30 H), and a STROBE pulse must now be written to the control latch. Bit 0 must be set high in addition to keeping a high at SELECT. After the strobe is set high it should be immediately set low (strobed), while still keeping the printer selected, This is accomplished by writing another 08 hex to the control port.

At this point, the printer should input the written character, and as this may take time to do so, the status of the printer should be checked by reading the input handshake lines at I/O address 31 hex. This is the same address as the printer control outputs, and is the printer status port. Bit assignments are as follows :

BIT0	Always 0	(spare)
BIT1	Always 0	(spare)
BIT2	Always 0	(spare)
BIT3	Printer ERROR	lets computer know a problem exists
BIT4	SELECT(ed)	confirms printer is ON LINE
BIT5	PAPER END	lets computer know of paper out
BIT6	ACKNOWLEDGE	confirms printer ready for data
BIT7	BUSY	goes away when printer operation in progress is complete

Either acknowledge or busy can be used to know when to initiate another data write cycle as described above. Basically, the printer must be selected, selection confirmed, data placed at data port, strobed, and the busy or acknowledge state checked, and re-looped until the character to be printed is finished. The signal ACKNOWLEDGE only refers to the status of the printer's data input buffer. It only acknowledges that it has received a character. The BUSY line behaves similarly, in that it acknowledges the buffer ready condition, but it also includes the printer's overall readiness to accept data. (I.E. the BUSY line will be asserted even if the printer is out of paper or unselected, etc.) The SELECT OUTPUT lead acts as a " chip select " for the printer, and no data or control transactions should be considered valid unless it is active. The SELECT INPUT lead is used to confirm the printer is connected, alive, and ON LINE. The INITIALIZE lead acts as a RESET to adjust the printer, and it should be strobed high, then back low. Signals ERROR and PAPER END are useful, but not absolutely essential signals, as the printer will not issue an ACKNOWLEDGE if these " problem " signals are active. (I.E. they are diagnostics). Signals ACKNOWLEDGE and/or BUSY are used during the data handshake for printing. Signal AUTOFEED is rarely used, except for special form feeding. Signal STROBE is used to send the data to the printer.

Lastly, should the Printer Port need to be configured for INPUT, such as when a hardware key is used, set control port bit 7 (READEN) high. (write an 80 H to address 31 H). This will disable the driver output of the DATA latch (address 30 H), and allow the port to be " read only ".

Bus Status LED's (Schematic Sheet #10):

In addition to the LED/Keypad display previously mentioned, the trainer has status LED's on the major busses and control lines of the CPU system. These LEDs are directly driven by the bus lines through open collector darlington drivers. The LEDs are on when the bus lines are at a logic high, or when the control lines are active. Some LEDs reflect a condition rather than a line status, so these are decoded status LEDs. The descriptions of the LED's are as follows:

Address LED's A15 through A8 and A7 through A0

The associated LED will light as each address line is active high. LED's are grouped in groups of four (4) LEDs to facilitate visual HEX conversion. During normal full speed operation of the trainer, these LEDs will be turning on off very quickly making the address very difficult to decipher. In order to see the address lines change the trainer must be HARDWARE Single Stepped. When Hardware Single Stepped the user will get an appreciation of the internal operation of the microprocessor.

Data LED's D7 through D0

The associated LED will light as each data bus line is active high, these lines, like the address bus, are also grouped in fours for easy HEX visualization. During normal full speed operation of the trainer, these LEDs will be turning on off very quickly making the data very difficult to decipher. In order to see the data lines change the trainer must be HARDWARE Single Stepped. When Hardware Single Stepped the user will get an appreciation of the internal operation of the microprocessor.

Status LED's RD,WR,I/O and MEM

The RD (Read) and WR (Write) lines are active low and so there associated LEDs light when these control lines are active at a logic low. RD and WR indicate whether the processor is reading information or writing information from an I/O device or memory. The I/O line is also active low and its LED is lit when in its active low state. When this LED is lit the microprocessor is accessing an I/O device. When the I/O line is high and its LED is off the MEM LED is lit indicating the microprocessor is accessing memory. The I/O and the MEM LEDs can be used in conjunction with the RD and WR LEDs to determine if the microprocessor is reading or writing to an I/O device or to memory. During normal full speed operation of the trainer, these LEDs will be turning on off very quickly making the control status very difficult to decipher. In order to see the control lines change the trainer must be HARDWARE Single Stepped. When Hardware Single Stepped the user will get an appreciation of the internal operation of the microprocessor.

Status LED IF

When this LED is lit, the CPU is performing an instruction fetch (IF). When CPU status output lines S0 and S1 are both high, while RD is active, an instruction fetch is occurring and decoding circuitry will enable the IF LED.

Status LED's RDY and WAIT

The RDY and WAIT LEDs light to indicate the operational status of the CPU. The CPU does not require wait states at the low clock speed it operates, but the hardware single step circuit puts the CPU in wait condition between steps. When the microprocessor is in a wait state the address, data, and control busses are all active but frozen. If the CPU is in normal operation, the RDY (Ready) LED will be lit. When the Single Stepper stops the CPU mid cycle, the WAIT LED will be lit indicating the microprocessor is waiting to continue.

Status LED HALT

A Halt state is not to be confused with a Wait state. The Halt condition occurs as a result of a software instruction, but Wait is a hardware induced state. When a HALT instruction is executed, the CPU goes to " Sleep " , and most of the outputs go TRI-STATE. (The trainer's address drivers do not tri-state in the halt condition, however, so the busses will still arbitrarily be driven by the floating driver inputs). The only way to continue processing from a HALT state is through an unmasked hardware interrupt occurring with interrupts enabled. Normally pressing one of the keypad keys will accomplish this.

Note: In a WAIT state the CPU is fully active, but suspended in mid cycle, waiting for the device that placed a low on the READY line to release the wait state. As the CPU recognizes READY inactive slightly before T4 (wait cycle), the CPU will stop during a RD or WR active. But the HALT state, being an altogether different type of CPU action, has its own LED. The CPU status lines S0 and S1 are asserted low together to indicate the HALT state, and the decoder for this condition lights the LED through a driver. As mentioned above, once the CPU is in a HALT state, only a previously enabled interrupt or a system reset can reawaken the CPU.

TIMER / COUNTER AND SPEAKER (Schematic Sheet #11)

The trainer has three counter/timers for general use and experimentation purposes. The trainer utilizes the INTEL 8253 chip, which has three (3) 16 bit counters. Each counter is individually programmable into many counting and timing modes.

Counters 0 and 1 have 307.2 KHz applied to their clock inputs, and their enable pins (G0 and G1) are tied to Vcc (the counters will not operate otherwise). The counters can be polled at any time to determine their conditions. Counter 0's output is connected to 8259 interrupt input pin IR5. Counter 1's output drives the UT's sub-miniature speaker and it may be programmed to output many different tones.

Counter 2's output drives a digital I/O port input line (Port C - Bit 5), the 8085's RST 7.5 line, and the 8259 interrupt input pin IR6, to provide polled and/or interrupt driven access to the output. In addition, JP29 provide access to Counter/Timer 2's GATE input and JP30 allows you to access the clock input as well as jumper it to the system clock of 3.072 MHz or to ground. Access to the counter input allows this Counter/Timer to count or time external events. For additional information see the Jumper Description section.

PARALLEL PERIPHERAL INTERFACE (PPI) I/O PORTS (Schematic Sheet #12):

The trainer has two general purpose digital I/O ports with status LEDs for visual indication of inputs and outputs. Intel's 8255 PPI chip has three (3) 8 bit parallel I/O ports. The Monitor Operating System initializes the PPI, and the user should NOT change this configuration. In the trainer's normal configuration, Port A is configured as a digital output port, and drives LED's through an LED driver. Port B is configured as a input port, and is connected to a DIP switch with status LEDs.

Port C is used for general purpose control of several I/O devices. Bits 0 through 3 are programmed as outputs. Bits 0 and 1 of port C are decoded to specify which of the four Analog to Digital Converter channels is active. Bits 0 and 1 also are connected to the clock and the data input of the serial EEPROM (optional on the military version). Bit 2 controls the transmit enable line of the RS422/485 serial port. Bit 3 selects the serial EEPROM and is used in conjunction Bits 0 and 1. Bits 4 through 7 are programmed as inputs. Bit 4 is connected to the Analog to Digital Converter. When this input goes low the ADC has completed its conversion and is ready to be read. Bit 5 is connected to the output of Timer 2 of the 8253 IC. When this input goes high Timer 2 count has reached zero. Bit 6 is connected to the output of the serial EEPROM. Bit 7 is not used. The parallel I/O ports A and B have a 20 pin connector (CN5) which permits external access to these parallel I/O ports. Signals are asserted active low, and may also be loopbacked for testing purposes. When connecting external digital I/O, be aware the hardware interface is active low for logic true signals (Negative Logic). The digital input DIP switch S2 should be turned off at all eight (8) stations, otherwise it will short to ground the input signal lines, causing erroneous responses or possibly even damage to the external device and the trainer.

The Parallel Digital I/O port, as described above, may be connected to external devices via connector CN5. The A port of the 82C55 PPI chip is connected to an open collector Darlington driver I.C. with eight (8) output channels. A logic 1 written to each port line will turn on the associated driver, grounding it, which will light the associated LED. Each output is rated to "sink" 500 mA. singly, or together with no more than a total package dissipation of 1.5 Amps. Each output is pulled up to +5 volts via a resistor pack, for TTL/MOS interface. Be aware that the saturation voltage of the darlington outputs is .9 volts, which may not satisfy all TTL gate inputs. However most NMOS and CMOS inputs will not care about this slight level mismatch. These outputs are primarily intended to drive relays, light bulbs, etc. When driving relays the maximum coil voltage should be 12 volts, and be sure to have flyback diodes correctly in place! The high value of the trainer's pullup resistor will not interfere with the relay drop-out current, and the LED's will be reversed biased when outputs are off (If > 5 volts is used for relay common). The driver I.C. type ULN2803A, has flyback diodes built into a common bus, provided at pin 10 of the package. The trainer does not connect to this pin, in order to not limit the value of the external supply. The user may connect to this pin should he wish to make use of the built in flyback diodes.

The Digital Inputs of the trainer are connected from connector CN5 directly to PPI port B. These eight (8) inputs are also pulled up to +5 volt via a resistor pack, they all have LED's, and all are active LOW. A DIP switch, S2 is used for manual digital input entry. The DIP switch switches the input lines to ground as each switch station is turned ON. Software reads PPI port B in, and inverts the lows to highs, for easy reading of the port input in logical convention. These inputs should NEVER be driven above +5 volts, (may cause LATCHUP) nor below trainer digital GROUND. These inputs are primarily intended to read switch or relay contact dry circuit closures, utilizing the trainer's built in pullup resistors. Also, when the digital inputs are connected to external hardware via CN5, be sure to switch appropriate S2 stations OFF, or else false inputs will occur.

JUMPER DESCRIPTIONS

The trainer has an assortment of jumper options, which are used to configure various options or in some cases to cause programmable " failures ". Some jumpers have an inactive position, or " parking spot " to help keep the jumpers from getting lost when not in use at that jumper position. Most jumpers' functions are plainly marked, those that are only labeled " A " or " B ", or " 1 ", " 2 ", etc. are explained here. A list of the jumpers and their descriptions as found on the Revision 1 board is as follows:

JP1 This is a failure configuration jumper. For normal Trainer operation, this jumper should be placed in position " A ". Removal of this jumper, or placing it in its " parking spot ", position " B ", causes the -8 volt inverting switcher to shut down, creating -8 volt supply " failure ". This will only affect the analog and communication sections.

JP2 This jumper configures the Bus Expansion Connector to operate with either EPAC 1000 series expansion peripherals, or 3000 type peripherals. The jumper is clearly marked on the board as to which position is which. This jumpers ties connector CN1 pin 25 to either CPU signal ALE, or to ground. EPAC expansion cards monitor this pin, and behave accordingly. This jumper is always in one mode or the other, so it does not require a " parking spot ".

JP3 This jumper ties the RST6.5 input of the 8085 CPU either to +5v, or to the expansion connector for external use. Note that there is an R-C network to ground, the R-C network filters out noise spikes from adjacent leads when a long ribbon cable is used on the BUS EXPANSION PORT. This jumper also has no particular parking spot.

NOTE: When the trainer is using the Monitor Operating System (MOS) EPROM, the normal configuration of JP3 is placed in the +5v position, to allow software single stepping.

JP4 This jumper selects the type of EPROM chip used by the Trainer. When in the 16 K position, EPROM type 27C128's are useable, and in the 32 K position, EPROM type 27C256 is selected. The trainer monitor version 1 uses 16 K (27C128), future revisions may use the 32 K option.

JP5 This jumper selects between 8 K or 32 K static RAMs, types 6264LP or 62256LP.

JP6 This jumper selects MEMORY MAPPED I/O (in addition to regular I/O), or I/O mapped I/O only.

JP7 This jumper should be placed across pins 1 and 2 for normal trainer operation. Removing it or placing it at parking spot pins 2 and 3, places a logic HIGH on the HOLD input to the 8085 CPU. The CPU will suspend normal operation, and TRI-STATE it's outputs. It will then activate it's HOLD ACKNOWLEDGE pin, and the trainer bus buffers will also go TRI-

STATE. This will leave the CPU DATA, ADDRESS, and CONTROL busses floating. The purpose of this jumper is to allow connection of other bus masters to the trainer, such as a DMA controller. JP7 is used as an input to the system in this mode (response to a DMA request), or this jumper can be used as one of the programmable " failures ".

JP8-JP13 These jumpers select the type of Memory chips used in the RAMDISK memory slots. They choose power, address, and strobe functions as per the type of memory required. The RAMDISK sockets U31 and U32 are 28 pin sockets, which will support the use of 8 K or 32 K STATIC RAMS, RAMDISKS, or 32 K EPROMS. All positions are marked " A " or " B ". Following is a table relating type of memory chip to jumper settings for the trainer's RAMDISK.

JP8, JP9, and JP10 are for RAMDISK socket 0, and JP11, JP12, and JP13 are for RAMDISK socket 1.

USAGE	TYPE	JP8/11	JP9/12	JP10/13
8 K SRAM	6264LP	A	A	A
32 K SRAM	62256LP	B	A	B
32 K EPROM	27C256	A	B	B

JP14 This jumper is normally in position " A " in Trainer operation, but can be moved to " B " to " WRITE PROTECT " the RAM ICs in the RAMDISK sockets. Positioning this jumper to " B " inhibits the CPU write strobe from reaching sockets U31 and U32. When in position " A ", WRITE is enabled, and LED D39 is lit indicating this condition.

JP15 These jumpers select whether an external CTS (Clear To Send) is recognized at the COM 1 communication port connector. The 8251 UART will not transmit data (internal hardware lockout) unless CTS is active. As a result, these jumpers double as programmable fault jumpers. As the UART must be transmit enabled, jumper JP15 provides a pullup resistor to +8 volts, (labeled +V) which through the RS-232 interface chip, will assert CTS active low at the 8251, permitting transmission of data. With systems or experiments requiring the use of CTS, the jumper is placed in the CTS position, which connects RS-232 CTS line to the connector. In normal operation, JP15 is left in the +V (TIE-UP) position. JP15 is for Com. channel COM1, at connector CN3.

JP16 This jumper permit selection of the Communication Baud rates desired at COM1. Placement of a jumper at one of the seven (7) positions for each jumper header select baud rates from 300 to 19,200.

JP17 This jumper permit selection of the Communication Baud rates desired at COM2. Placement of a jumper at one of the seven (7) positions for each jumper header select baud rates from 300 to 19,200. COM2 is optional on the Military Version of the Universal Trainer.

JP18 These jumpers select whether an external CTS (Clear To Send) is recognized at the COM 2 communication port connector. The 8251 UART will not transmit data (internal hardware lockout) unless CTS is active. As a result, these jumpers double as programmable fault jumpers. As the UART must be transmit enabled, jumper JP18 provides a pullup resistor to +8 volts, (labeled +V) which through the RS-232 interface chip, will assert CTS active low at the 8251, permitting transmission of data. With systems or experiments requiring the use

of CTS, the jumper is placed in the CTS position, which connects RS-232 CTS line to the connector. In normal operation, JP18 is left in the +V (TIE-UP) position. JP18 is for COM2 at connector CN4.

JP19 This is a programmable failure jumper, it is normally in position " A ". Placement in " B " of this jumper causes the feedback loop of the analog sample/hold circuit to open, creating an analog to digital converter failure.

JP20 This is a programmable failure jumper, it is normally placed in position " B ". Placement in " A " shorts the CPU RESET input pin low, and places the CPU in eternal reset.

JP21 This is a fiendish programmable failure jumper. Normally placed in position " C ". It has two failure positions, A and B. Failure " A " shorts latched CPU address line A5 to CPU data bus line D2. Position " B " shorts CPU data line D2 to ground only. The jumper is not located physically near the CPU itself, but to the left of the Memory sockets. Therefore, a lot of signal track tracing will be required to find it. The normal placement of this jumper is at position " C ", which is the jumpers' parking spot, where no shorts are created.

JP22 This programmable failure jumper ties the chip select for the 82C55 PPI chip to +Vcc when in the " B " position, disabling the Digital I/O section. With the jumper placed in the " A " position, the trainer will operate correctly. It is also placed far away from the 82C55 PPI chip, so it will require extensive signal track tracing to locate.

JP23 This programmable failure jumper is normally in position " B ". When it is jumpered to " A ", it shorts the enable pins of LED KEYPAD display driver U54 to ground. This causes the LED display to scramble, as this driver is normally toggled by display control lead SL0 NOT. The bit patterns to the display matrix will therefore be routed incorrectly to the display, and cause the display to be unreadable in most cases.

JP24 This programmable failure jumper is normally placed in position "A". If removed or placed into position "B", it will inhibit CPU signal WR NOT from getting to system RAM. Resistor R89 will pull the WR NOT pin of system RAM chip U19 high (inactive). The system cannot function as a result.

JP25 This option jumper can double as a failure jumper. Normally placed in position " B ", it grounds the Bus Expansion Header's INT0 pin, enabling the Manual INT0 pushbutton. The Expansion INT0 pin and the Pushbutton are logically OR'ed. A pullup resistor on the Expansion connector pulls up the pin input, so a pushbutton input would be inhibited if JP25 was not in the " B " position. When an external INT0 is desired, the jumper should be is position " A ".

JP26 This programmable failure jumper is normally parked at position " A ". Placement at " B " will short Address line A1 to ground, creating a critical system failure.

JP27 On trainers with the D/A output option ordered, this jumper may be used as a failure jumper or as an output connector. Placement of the jumper across position " A " shorts out the D/A system output going to the bargraph display, creating a " failure ". Placement at " B ", is the normal operation parking spot. Also the jumper may be used as an output connector by taking the output at pin " A ", and ground at pin " B ".

JP28 This relatively simple failure jumper, when placed at " A ", shorts out the +2.5 volt reference

for the A/D and optional D/A, this creates incorrect analog values. Placement at " B " is the normal condition parking spot.

JP29 This jumper, in conjunction with JP30, provides external access to timer chip U47 (82C53), Timer channel 2. Timer 2 can be used to count or time external events, presented to the trainer for experimentation purposes. Specifically, JP29 selects whether CLK2 input to the 82C55 is fed from timer 0's output, (prescaler), or is externally fed. JP29, pin 1 is grounded, and may be used as the ground side of this external input, with pin 2 as the signal in. Pin 3 is Timer 0's output. The normal configuration for this jumper is placement across pins 2 and 3. If providing an external input signal for Timer 2 be sure its TTL compatible.

JP30 This jumper is similar to JP29, and it options Timer 2's GATE input. In most cases, the GATE input of the timer must be logic high for the counter to work. The gate enable signal here may be set to +5V with jumper JP30 at positions 2-3, or an external gate signal may be fed in at pin 2 of JP30. As in JP29, pin 1 is ground. The normal position for JP30 is placement of a jumper across 2-3. If providing an external gate signal for Timer 2 be sure its TTL compatible.

JP31 This jumper is a programmable failure jumper. When placed in the failure mode, it disconnects Vcc to the 8085 CPU chip. When parked in the correct position, power is restored to the CPU. Failure position is at " B ", and normal placement is at " A ".

JP32 This jumper is a programmable failure jumper. When placed in the failure mode, it shorts the crystal input Y1, X2 of the CPU to ground, killing the oscillator. The normal position of this jumper is at " B ", and the failure position is " A ". Note: The trainer should be powered down and then re-powered after correcting this fault.

Summarization of Default Jumper Settings

Should suspicion arise that the trainer may be malfunctioning, or operational difficulty be encountered with the trainer, it is suggested that the jumpers be placed into their default positions, to allow the user to see if in fact the trainer is working correctly. The following is a listing of the "default" jumper positions for the trainer. Certain jumper positions will depend on the memory configurations present and will be marked by " * ", refer to the previous tables for these jumper settings. The Trainer unit should be powered down prior to setting the jumpers, and re-powered when jumpering is complete.

JP1	**Position**	**A**
JP2	**Position**	**1TH**
JP3	**Position**	**+5V**
JP4	**Position**	**32K ***
JP5	**Position**	**32K ***
JP6	**Position**	**I/O ONLY**
JP7	**Position**	**1-2**
JP8	**Position**	**B**
JP9	**Position**	**A**
JP10	**Position**	**B**
JP11	**Position**	**B**
JP12	**Position**	**A**
JP13	**Position**	**B**

JP14	**Position**	**Not Critical ***
JP15	**Position**	**+V**
JP16	**Position**	**9600 (or to your terminal's rate)**
JP17	**Position**	**300**
JP18	**Position**	**+V**
JP19	**Position**	**A**
JP20	**Position**	**B**
JP21	**Position**	**C**
JP22	**Position**	**A**
JP23	**Position**	**B**
JP24	**Position**	**A**
JP25	**Position**	**B**
JP26	**Position**	**A**
JP27	**Position**	**B**
JP28	**Position**	**B**
JP29	**Position**	**2-3**
JP30	**Position**	**2-3**
JP31	**Position**	**A**
JP32	**Position**	**B**

NOTE: If problems arise in addition to checking the jumpers remove any devices connected to the Bus Expansion port, Digital I/O port, signal conditioning card slot, any wires on Screw terminal ST1 and ST2, Parallel Printer Port CN2, and Communication ports CN3 and CN4. The trainer monitor program EPROM should be in place, at socket U18, and an 32 K STATIC RAM (62256LP) in socket U19. This procedure will help isolate faults created externally to the Universal Trainer board from interfering with this test.

Provide power to the trainer via power jack J1, preferably from the 12 volt D.C. wall mount power supply, (Be sure, of course, that it is plugged into a live outlet!) and switch on the unit. If trainer does not display the proper data to the LED display, something may have gone bad or been blown on the trainer. Contact EMAC for repair details.

OPTIONAL ON BOARD HARDWARE FEATURES

REAL TIME CLOCK CALENDAR: The Universal Trainer can be equipped with a real time clock/calendar (RTC). The RTC contains a lithium energy cell which maintains clock information and RAM memory data. The clock keeps time in hundredths of seconds, tenths of seconds, seconds, minutes, hours, day of week, date of month, month and year. The month and year determine the number of days in each month. If you have the real time clock option refer to the specification sheet included in your RTC Supplement for technical and programming specifications. This option can be installed at any time by the user or at the factory before shipping.

32K X 8 NONVOLATILE RAM (RAMDISK): EMAC has a 32K RAMDISK available for source code or data storage. The RAMDISK has a built-in lithium energy cell which maintains data in RAM memory. The RAMDISK normally resides in RAMDISK socket (slot) 1 or 2. Programs or data can be written into and stored for later use in the RAMDISK. The RAMDISK may be removed from the Trainer and replaced at a later time without loss of memory which allows the RAMDISKS(s) to be removed/inserted much like floppy disks.

10 BIT ANALOG TO DIGITAL CONVERTER: An optional 10 bit A/D can be installed in place of the standard 8 bit A/D. The 10 bit A/D provides more precision for use in monitor and control tasks.

1K X 1 SERIAL EEPROM: This optional chip provides 1K bits of serial nonvolatile data storage for the trainer.

OTHER OPTIONS FOR TRAINER SYSTEMS

SIGNAL CONDITIONING CARDS: A low cost method of implementing Analog Data Acquisition, such as temperature, pressure, and strain.

EPROM PROGRAMMER BOARD: A simple solution to burning programs.

PROGRAMMABLE 32 LINE PARALLEL BOARD: 32 programmable digital I/O lines with I/O rack connector compatibility.

SUPPORT SOFTWARE: When your Universal Trainer needs to communicate with a PC/Terminal.

APPENDIX A

APPENDIX B

ASSEMBLY LANGUAGE LISTING
OF THE
MONITOR OPERATING SYSTEM

APPENDIX C

Schematics

1) POWER SUPPLY
2) CPU - MEMORY DECODING
3) BUS EXPANSION AND INTERRUPT CONTROL
4) COMMUNICATION PORTS
5) RAMDISK STORAGE
6) ANALOG INPUT
7) ANALOG OUTPUT
8) KEYPAD AND 14 SEGMENT LED DISPLAY
9) PARALLEL PRINTER PORT
10) BUS/CONTROL STATUS LEDS
11) TIMERS AND SPEAKER
12) DIGITAL I/O AND EEPROM
13) HARDWARE SINGLE STEPPER