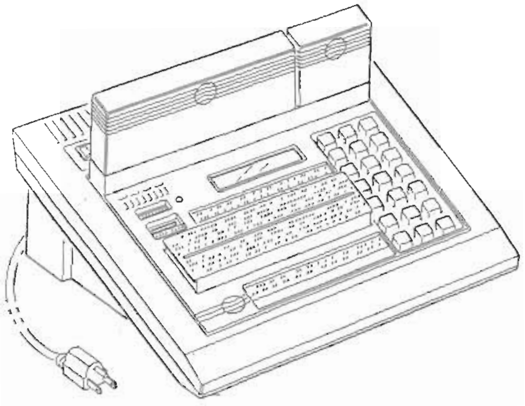
MICROPROCESSOR TRAINER

Model ETW-3800
with ETC-8085 CPU MODULE

User's Manual 595-4330



Heathkit
Zenith Educational Systems

HEATH COMPANY
BENTON HARBOR, MICHIGAN 49022

WARNING

This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only computers certified to comply with the Class B limits may be attached to this equipment. Operation with non-certified computers is likely to result in interference to radio and TV reception.

This equipment uses radio frequency energy for its operation; and if it is not installed and used properly, that is, in strict accordance with the instruction manual, it may cause interference to radio and television reception. It has been type tested and found to comply with the RF emission limits for a Class B computing device which is intended to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause interference to radio and television reception, which you can determine by turning the equipment off and on, try to correct the interference by one or more of the following measures:

- Move the computing device away from the receiver being interfered with.
- Relocate the computing device with respect to the receiver.
- Reorient the receiving antenna.
- Plug the computing device into a different AC outlet so that the computing device and receiver are on different branch circuits.
- Disconnect and remove any experimental connecting leads and/or I/O cables that are not being used. (Unterminated leads and/or I/O cables are a potential source of high RF emission levels.)
- Unplug and remove any experimental circuits that are not being used.
- Obtain results of the experiment as quickly as possible, then turn the computing device off.
- Be certain that the computing devices are plugged into grounded outlet receptacles. (Avoid using AC cheater plugs. Lifting of the power cord ground may increase RF emission levels and may also present a lethal shock hazard to the user.)

NOTE: In order to meet Class B emission limits, the user must comply with the following requirements:

- The I/O cables that interconnect between this computer and any peripheral (such as a printer, modem, etc.) must be shielded.
- The line (power) cord shipped with the computer is shielded. If you replace the line cord, be sure to use only a shielded line cord.

If you need additional help, consult your dealer or ask for assistance from the manufacturer. Customer service information is on the inside back cover of this Manual or on an insert sheet supplied with this equipment. You may also find the following booklet helpful: "How to Identify and Resolve Radio-TV Interference Problems." This booklet is available from the US Government Printing Office, Washington, D.C. 20402, Stock No. 004-000-00345-4.

Table of Contents

INTRODUCTION

The Model ETW-3800 Microprocessor Trainer is a practical learning tool specially designed to help you understand 8-bit microprocessor operation, programming, and applications. A unique feature of this Microprocessor Trainer is separate CPU and memory cartridges. By simply replacing a cartridge you can inexpensively study and experiment with different microprocessors using just one educational trainer.

The included ETC-8085 CPU Module cartridge contains the 80C85AH microprocessor, RAM, ROM and buffer circuitry. The 8085 is a member of the Intel 8080 family. For increased flexibility and expanded Trainer operation, the cartridge can be easily replaced with our other 8-bit microprocessor cartridges.

A replaceable ETC-128 Memory Module cartridge is also available as a separate program storage option. Use this cartridge to save programs for downloading at a later time. You will save time over manually re-entering programs and you will eliminate the possibility of entry errors. The Trainer maintains a full 16-bit data path to the memory cartridge so that it can be configured for 16K by 8, 8K by 16, or up to 64K by 16 bits of data. This also enables the memory cartridge to be shared with the Heathkit/Zenith 16-bit microprocessor trainer.

Other hardware features include:

- 20-character by 2-line liquid crystal display (LCD).
- Hexadecimal keypad with added function keys.
- 8 display and 4 programmable status light emitting diodes (LEDs).
- Binary logic switches.
- Input and output data bus ports.
- Logic probe provides visual and audible indications.
- Configured 1 Hz and 60 Hz square wave signals.
- Large breadboard for building experimental circuits.
- Connector blocks for solderless connections between parts and wires.
- Power supply outputs allow access to +5 VDC, +12 VDC, and −12 VDC voltages.
- Standby power to CPU RAM for saving programs in memory while modifying circuitry with the Trainer turned off.
- RS-232 interface for use with computers and terminals up to 1200 baud.
- Expansion connector for the ET/ETW-3567 Heathkit/Zenith Accessory Backpack.

The ETW-3800 Trainer provides a help key, help menus, and easy-to-understand prompts to guide you through all operations. The Trainer can be operated from its hexadecimal keyboard or from a terminal or a computer using terminal emulation software. Keyboard commands let you examine and modify memory contents, as well as insert data; examine and modify registers; copy blocks of memory to another memory location or to a Memory Module cartridge; and initialize blocks of memory. Other keyboard commands let you access 16 software break points, 16 watch locations and seven watch registers. More commands enable you to load and save files, receive Intel HEX files, single step through programs, disassemble programs, and run programs in real time with or without breakpoints.

The ETW-3800 Microprocessor Trainer is recommended for conducting experiments contained in Heathkit/Zenith Educational Systems learning programs. Associated courses include separate Memory Modules preprogrammed with experiments.

SPECIFICATIONS

CPU Module	Separate plug-in cartridge containing an 8-bit Intel 80C85 microprocessor, A/D-D/A converter, 24K × 8 Monitor EPROM and 24K × 8 RAM.
Display	20 × 2 line LCD with definable characters.
Keyboard	21-key hexadecimal keypad including RESET and NMI (Non-Maskable Interrupt Trap).
Output Port	8-bit output latch in the CPU address space.
Input Port	8-bit input latch in the CPU address space.
Logic Switches	Eight miniature on/off binary switches in a dual in-line package with separate input terminals.
Logic Indicators	Eight green LEDs with separate input terminals.

LOGIC PROBE

Logic High	≥2.0 volts.
Logic Low	≤0.8 volts.
Minimum Glitch Recognition	50 MHz.
Visible Output	Red and green LEDs.
Audible Output with Disable	1.8 kHz, 2.2 kHz.

POWER SUPPLIES

Output Voltages	+5 volts DC at 0.5 amp. +12 volts DC at 0.1 amp. −12 volts DC at 0.1 amp.

Load Regulation	5%.
Standby Supply (to CPU RAM only)	4.7 volts from 0.1F capacitor for RAM locations 6821H - BFFFH, approximately 30 minutes.

GENERAL

Power Requirements	120/240 VAC, 50/60 Hz, 20 watts maximum.
Fuse	1/4-ampere slow-blow.
Dimensions	4-1/4" H × 12" W × 11-1/4" D.
Weight	5 lbs.

The Heath Company reserves the right to discontinue products and to change specifications at any time without incurring any obligation to incorporate new features in products previously sold.

STARTUP

This section of the Manual tells you how to set up your Microprocessor Trainer, shows you how to power up the Trainer and check out basic functions, and lists operating precautions.

SETUP

Your ETW-3800 Microprocessor Trainer requires the installation of the CPU Module to be operational. When the Memory Module is installed, you can save and load programs.

1. Install the two connector block overlays included with the CPU Module on the Trainer's signal connector blocks. The overlay labeled LPIN to GND installs over the bottom connector block. The overlay labeled D0-D7 to $\overline{\text{ADSTR}}$ installs over the top block. Refer to Pictorial 1 on fold-out of page 11.

2. Remove the static protection clip from the ETC-8085 CPU Module and install the Module into the two left connectors. Make sure the lettering on the cartridge is facing you. This Module is keyed to go into the connectors of the Trainer one way only. Do not force the Module in backwards.

   To permanently install the Module, first remove the eight screws that secure the cabinet top to the cabinet bottom. Then insert the CPU Module into the connectors and use the included #6BT x 1.5" screw to anchor the Module to the Trainer through the circuit board. Do not overtighten the screw. When finished, reassemble the cabinet.

3. The ETC-128 Memory Module occupies the right connector of the Trainer. Install this optional cartridge only when you save or download programs to avoid accidentally overwriting your stored programs. Make sure the lettering on the cartridge is facing you. This Module is also keyed so it fits the connector one way only.

Refer to the Appendix for the proper procedure when installing, removing, and storing cartridges.

POWERING UP

1. Make sure the POWER switch is off (push down on the left side of the switch).

2. Plug the line cord of the Trainer into the proper AC outlet.

NOTE: If you do not obtain the proper results in the following steps, push the POWER switch to off and unplug the line cord. Then refer to the "In Case of Difficulty" section to correct the problem before you continue.

3. Push the POWER switch to on. The POWER LED should light. Also, the Logic Probe transducer beeps and the four programmable status LEDs should light. The LCD display reads:

```
Heath/Zenith  80C85A
Educational MPU V1.0
```

4. Adjust the CONTRAST control for best display viewing. This screwdriver adjustment is accessible through the cabinet at the left of the LCD display.

OPERATING PRECAUTIONS

DO	DO NOT
DO turn the power off when inserting or removing the CPU cartridge. DO turn the power off before inserting or removing backpack boards from the ET-3567 Accessory Backpack when used. DO unplug the program cartridge when you are not loading or saving a program. This will prevent accidental writes to the program address range by your programs. DO place the static protection clip on the CPU cartridge when not plugged into the Trainer.	DO NOT remove the ground lead from the line cord. DO NOT use an ungrounded AC power line adapter. DO NOT replace the fuse with any type or value other than the one specified on the fuse label. DO NOT connect any external power sources to any of the Trainer breadboard points. DO NOT connect voltages above +5 volts or any negative voltages to the inputs or outputs of the Trainer, except the analog inputs. The inputs and outputs of the Trainer are protected with series resistors, but prolonged overvoltage conditions will shorten the life of the ICs in the Trainer. DO NOT insert larger than #20 (0.032") solid wire or component leads in the connectors of the Trainer. DO NOT expose your Trainer to moisture. DO NOT plug any Backpack Boards designed for the ETW-3600, ETW-3700, or ETW-5000 Trainers into an ETW-3567 Accessory Backpack attached to the ETW-3800 Trainer.

OPERATION

This section of the Manual describes the operation of your Trainer, explains the keyboard commands, describes how to enter programs, contains sample programs, and explains about memory I/O and the use of the jump and interrupt tables. Alternate line voltage wiring is also shown.

TRAINER

Pictorial 1 (fold-out on page 11) gives a brief description of switches, LEDs and connectors. Also refer to the Pictorial while reading the following paragraphs.

PROGRAMMABLE STATUS LEDS

These four LEDs can be programmed to indicate the current status of a program. They are controlled by the upper nibble (4-bit word) of I/O address 30H. The first LED (leftmost) is turned on by a "0" in bit 4 (the 5th position) of address 30H. The remaining LEDs are similarly controlled by the next higher bit positions: second LED by bit 5, third LED by bit 6, and fourth LED (rightmost) by bit 7.

The Status LEDs can be used to provide a visual indication of the current status of a program with the placement of OUT statements (D3H) anywhere within a program. Or, the Status LEDs can be used to display debugging or output information. NOTE: Writing to I/O address 30H will also affect the lower 4-bits of the I/O address which consists of the A/D range and the input select lines. See Diagram 2 on page 34 for bit locations.

LOGIC PROBE

Two LEDs and a transducer make up the Logic Probe. The red LED lights for a logic high and the green LED lights for a logic low. The Logic Probe is accessible through the lower signal connector block. High and low logic levels also produce an audible response from the nearby transducer. The audible output can be disabled with a jumper to ground from the $\overline{\text{LPAUDC}}$ line on the lower signal connector block.

LOWER SIGNAL CONNECTOR BLOCK

This signal block allows you access to input and output ports, microprocessor control lines, A/D-D/A converter lines, +5 VDC, ±12 VDC, and the logic probe through connectors. These lines are labeled at their terminal points on the connector block and are color coded. (If the two connector block labels included with the ETC-8085 CPU Module have not yet been installed, install them now following the Setup procedure on page 8.) Red areas are for output only, blue areas are input only, and green areas are bi-directional. The signal block consists of two rows of fifty separate connectors with each top connector internally connected to the bottom connector. The connectors will accept up to #20 (0.032") solid wire and most common component leads. All the signal lines are protected by series resistors. The following lines are available at the signal block.

LPIN	Logic Probe Input.
$\overline{\text{LPAUD}}$	Logic Probe Audio Disable. Disable by connecting to ground.
GND	Digital Ground.
+5V	+5 Volt DC Output.
AGND	Analog Ground.

–12V –12 volt DC Output.

+12V +12 volt DC Output.

AI7-AI0 Analog Inputs. Lines to A/D converter, 8-bit 2's complement ±15.10, ±7.55, ±2.5, ±1.25, ±0.25, ±0.125, ±0.025, and ±0.0125 volt ranges available.

AOUT Analog Output, 8-bit 2's complement ±2.5 and ±1.25 volts, includes a 270Ω series resistor.

HDA CPU Hold Acknowledge.

$\overline{\text{INA}}$ CPU Interrupt Acknowledge.

$\overline{\text{HOLD}}$ CPU Hold.

$\overline{\text{INR}}$ CPU Interrupt Request.

$\overline{\text{R6.5}}$ CPU Restart Interrupt 6.5*, high until sampled.

$\overline{\text{R5.5}}$ CPU Restart Interrupt 5.5*, high until sampled.

ADBUSY A/D, D/A Busy Status Output. Active low when A/D is busy.

S0 CPU Machine Cycle Status Line 0.

S1 CPU Machine Cycle Status Line 1.

IO/M CPU Input/Output Memory (or Main Memory Select).

CLK CPU System Clock (1.8432 MHz).

ADINT A/D, D/A Interrupt Output. High to low transition upon completion of a conversion. Reset by read or write signal, or after a system RESET.

$\overline{\text{EXMSL}}$ External Memory Select. Selects external memory locations A000-BFFF when low. Operational only when $\overline{\text{ENEXM}}$ is pulled low.

$\overline{\text{IPL}}$ Input Port Latch. Latches data on falling edge.

IP7-IP0 Input Data Port. Located at 90H.

OP7-OP0 Output Data Port. Located at A0H.

GND Digital Ground.

UPPER SIGNAL CONNECTOR BLOCK

This signal block enables you to access most of the CPU Module's microprocessor, programmable timer, and external I/O select lines through connectors. The lines are labeled at their terminal points and are color coded. Red areas are for output only, blue areas are input only, and green areas are bi-directional. The signal block consists of two rows of fifty separate connectors with each top connector internally connected to the bottom connector. The connectors will accept up to #20 (0.032") solid wire and most common component leads. All the signal lines are protected by series resistors. The following lines are available at the signal block.

D0-D7 CPU Data Bus lines.

GT1 Timer Gate Input 1.

GT0 Timer Gate Input 0.

TCK1 Timer Clock Input 1.

TCK0 Timer Clock Input 0.

TO1 Timer Output 1. A 60 Hz square wave output only when CLK and TCK0 lines are connected.

T00 Timer Output 0. A 1 Hz square wave output only when 60 Hz square wave output from T01 is connected to TCK1.

$\overline{\text{ENEXM}}$ Internal RAM Disable. Disables internal RAM A000H to BFFFH when pulled low.

A15-A0 CPU Address Bus. Low byte is already latched.

$\overline{\text{READ}}$ Memory, I/O active-low read line.

$\overline{\text{WRITE}}$ Memory, I/O active-low write line.

RESET Reset output.

ALE Address Latch Enable. Low byte of address bus latched on rising edge of this line.

READY CPU ready line.

$\overline{\text{I/O 0}}$ Decoded I/O line. Address is 50H-5FH, active low.

$\overline{\text{I/O 1}}$ Decoded I/O line. Address is 60H-6FH, active low.

$\overline{\text{I/O 2}}$ Decoded I/O line. Address is 70H-7FH, active low.

$\overline{\text{ADSTR}}$ A/D Converter Start Line. Active low.

* Internal Restart Interrupt line R7.5, used by the Monitor ROM for RS-232 baud generation, can be reprogrammed as a user interrupt. The R7.5 line is connected to TO2 of the timer clock and can be reprogrammed by changing Counter 2. However, by reprogramming this interrupt, RS-232 capability may be lost until a new system Reset is generated.

MEMORY MODULE CONNECTOR

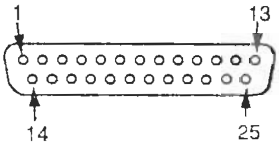
The Memory Module connector accepts optional Memory Module cartridges designed for this 40-pin connector. A full 16-bit data path is implemented to the Memory Module so that it can be configured for 16K × 8, 8K × 16, or up to 64K × 16 bits of data. In addition, other EPROM Memory Modules, preprogrammed with experiments, accompany selected Heathkit/Zenith Educational microprocessor courses.

CPU MODULE CONNECTORS

The CPU Module connectors accept CPU Module cartridges designed for these two dual 62-pin connectors. The interface for the separate, removable ETC-8085 CPU Module is designed so that other 8-bit microprocessor modules can be used with the Trainer. (See Appendix for the proper cartridge removal/insertion procedure.) All microprocessor-specific control lines are routed to two signal connector blocks located above and below the large Breadboard Block. The connector blocks are labeled according to the corresponding microprocessor and I/O device lines.

RS-232 INTERFACE

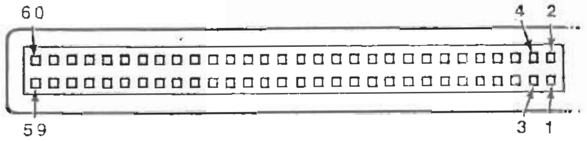
The RS-232 port is located on the right side of the Trainer. It allows the Trainer to be controlled from a terminal or a computer emulating a terminal. All the Trainer keyboard commands are available through the RS-232 port. You can also use the port to download programs written on a computer. The following signals are available at the connectors.



1- Chassis Ground.
2 - RXD (Receive Data).
3 - TXD (Transmit Data).
NOTE: When using a terminal, configure it for 8 data bits, 1 stop bit, no parity, and software handshaking only.

ACCESSORY BACKPACK CONNECTOR

A rear panel connector is provided for the ET/W-3567 Accessory Backpack. Available at the 60-pin connector are the microprocessor's data bus lines, the lower six address lines, control lines, four I/O select lines, and several other I/O lines. All the signal lines are protected by series resistors. By using the Accessory Backpack, you can connect prewired and experimental circuit boards to the Trainer. The following lines are available at the connector.



1 - Analog Input 0.
2 - Analog Input 1.
3 - CPU Hold Acknowledge.
4 - Analog Output.
5 - CPU Hold.
6- Interrupt Acknowledge.
7 - Reset Interrupt 6.5.
8 - Interrupt Request.
9 - Ground.
10 - Reset Interrupt 5.5.
11 - Ground.
12 - CPU Status Line S0.
13 - Ground.
14 - CPU Status Line S1.
15 - Ground.
16 - I/O Memory Select Line.
17 - Ground.
18 - System Clock (1.8432 MHz).
19 - Ground.
20 - CPU Address Line 5.
21 - Ground.
22 - CPU Address Line 4.
23 - Ground.
24 - CPU Address Line 3.
25 - Ground.
26 - CPU Address Line 2.
27 - Ground.
28 - CPU Address Line 1.
29 - Ground.
30 - CPU Address Line 0.
31 - Ground.
32 - CPU Read.
33 - Ground.
34 - CPU Write.
35 - Ground.
36 - CPU Reset Out Line.
37 - Ground.
38 - CPU Address Latch Enable.
39 - Ground.
40 - CPU Ready.
41 - Ground.
42 - General I/O Select 1.
43 - Ground.
44 - General I/O Select 2.
45 - Ground.
46 - CPU Data Line 0.
47 - Ground.
48 - CPU Data Line 1.
49 - Ground.
50 - CPU Data Line 2.
51 - Ground.
52 - CPU Data Line 3.
53 - Ground.
54 - CPU Data Line 4.
55 - Ground.
56 - CPU Data Line 5.
57 - Ground.
58 - CPU Data Line 6.
59 - Ground.
60 - CPU Data Line 7.

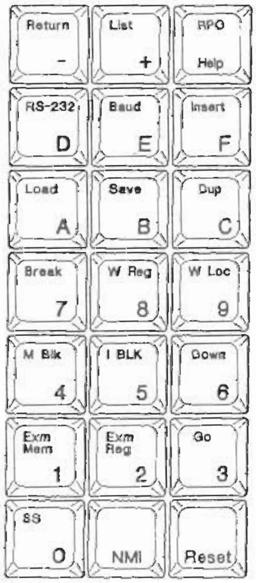
KEYBOARD

The keyboard allows you to quickly enter commands and data to the microprocessor (see Pictorial 2). After you press a command key, a brief prompt helps you complete the entry. All of the keyboard commands are also available through the RS-232 port. Some commands can be accessed with subroutine calls from a program.

After you press the POWER switch, the Trainer turns on in the Select Operation mode and the LCD display shows:

```
Heath/Zenith  80C85A
Educational MPU V1.0
```

The Trainer will interpret the next key entry as a command. After you enter a command, enter data using the 0 to F keys. The following paragraphs discuss the various commands.



PICTORIAL 2

Press the RETURN key (with terminals, use the – key) to exit a function (such as Set Breakpoint, or Single Step) and to save all data contained in the CPU's registers. If you use the RESET key to exit a function, the register values will be replaced by the Monitor ROM's default values. In the Examine Memory, Examine Register, and Insert modes, use the RETURN key to move to the previous memory or register location and use the RPO-HELP key (with terminals, use the ? key) to exit the function and still keep all the data in the registers.

Press the LIST (+) key in BREAK to list all the breakpoints. Likewise, you can list all the Watch Registers and the Watch Locations (addresses) by pressing the LIST key while in each mode.

Press the LIST (+) key in the Monitor mode to disassemble a program. The starting address is defaulted to 7000H and the ending address is defaulted to the current program counter register value. To step through the listing of your program, continuously press the plus (+) key until the ending address is reached. To exit back to the Monitor mode before the ending address is reached, press the minus (–) key. If you are using a terminal and the Trainer's RS-232 port, press Control C on the terminal's keyboard to exit the listing before the ending address is reached.

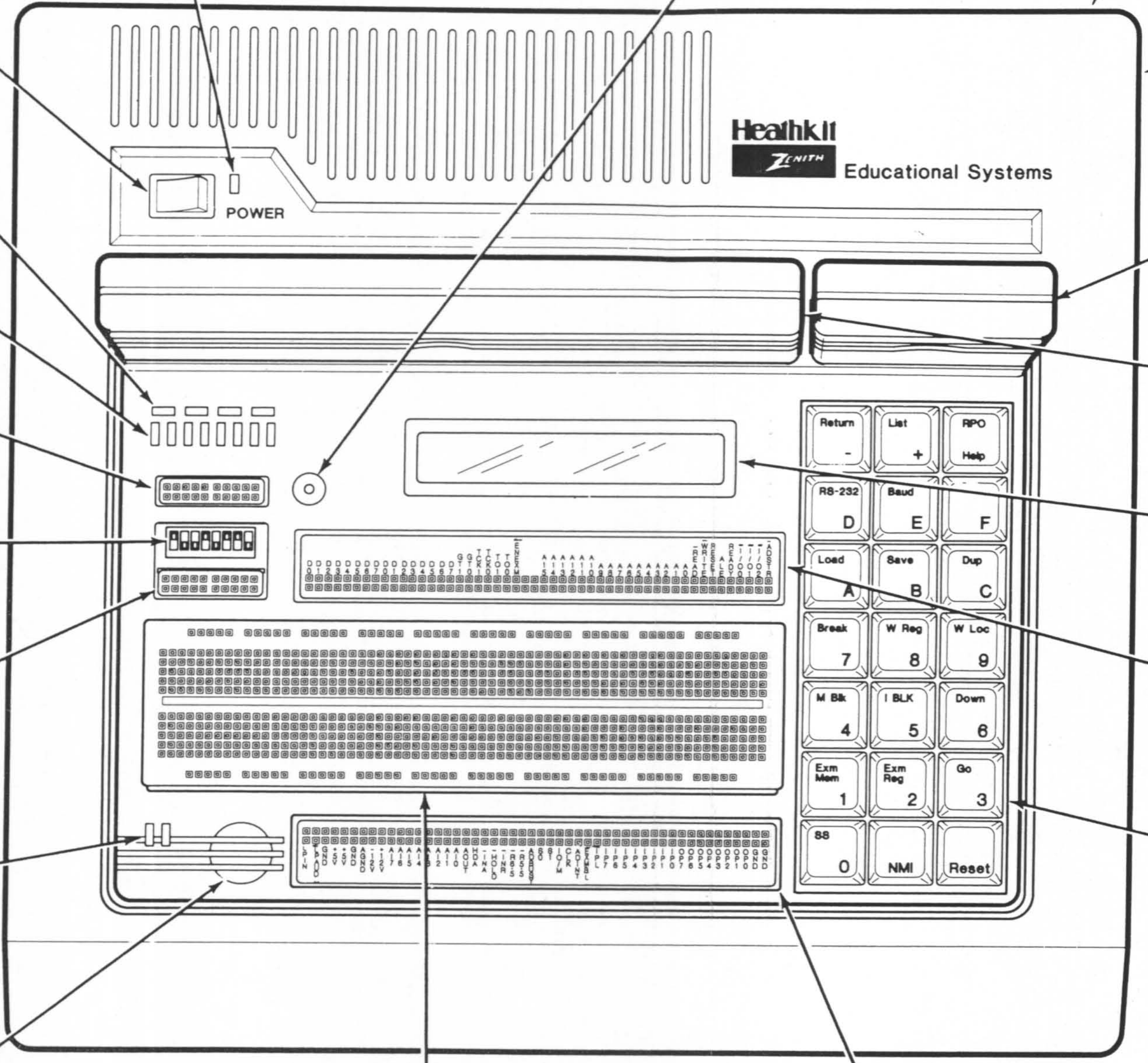
Press the HELP key to display help messages for the SS (single step), W Reg (watch register), and Load at the Monitor ROM level. The RS-232 equivalent for the HELP key is "?".

Press the RS-232 key to transfer control from the keyboard of the Trainer to the RS-232 serial port. When you press this key, all keyboard commands can be executed from a connected terminal or a computer using terminal emulation software. Select commands by entering the single key representing that command. All lower case letters are converted to upper case, except for print strings.

PRESS	DISPLAY READS
RS-232	RS-232 control Baud=1200

On your terminal or computer screen a ready prompt will appear:

```
>
```



POWER LED — Indicates when the Trainer is turned on.

DISPLAY CONTRAST CONTROL — Increases and decreases brightness of the LCD DISPLAY.

ACCESSORY BACKPACK CONNECTOR — Allows the Trainer to be used with the ET/ETW-3567 Accessory Backpack.

POWER SWITCH — Turns the Trainer on and off. Also turns the ET/ETW-3567 Accessory Backpack on and off when installed.

PROGRAMMABLE STATUS LEDS — Shows the current operating status of a program.

LOGIC LEDS — Turn on when a logic 1 is applied. The LED is off with a logic 0 applied.

Inputs for LOGIC LEDs.

LOGIC SWITCHES — Selects either a logic high or a logic low for each associated connector.

Outputs for LOGIC SWITCHES.

LOGIC PROBE LEDS — Provides a visual display of a logic high or a logic low when the LPIN line of the LOWER SIGNAL CONNECTOR BLOCK is connected to a digital circuit.

LOGIC PROBE TRANSDUCER — Provides an audible response to a logic high or a logic low when the LPIN line of the LOWER SIGNAL CONNECTOR BLOCK is connected to a digital circuit.

RS-232 CONNECTOR — Allows the Trainer to be used with terminals and computers.

MEMORY MODULE CONNECTOR

CPU MODULE CONNECTORS

LCD DISPLAY — Displays information.

UPPER SIGNAL CONNECTOR BLOCK — Provides access to various microprocessor and other signal lines.

KEYBOARD — Allows you to enter data and commands.

BREADBOARD — Provides solderless connections for building circuits.

LOWER SIGNAL CONNECTOR BLOCK — Provides access to various microprocessor lines.

PICTORIAL 1

CAUTION: Do not insert larger than #20 (0.032") solid wire or component leads in the connectors of this instrument.

At this point, press a key that corresponds to the following Trainer commands:

PRESS	FOR COMMAND
0	Single Step
1	Examine Memory
2	Examine Registers
3	Go
4	Move Block
5	Initialize Block
6	Receive SHEX File
7	Set Breakpoint
8	Set Watch Register
9	Set Watch Point
A	Load File
B	Save File
C	Duplicate File
D	Select Trainer/RS232
E	Select Baud
F	Insert
L	Look Addresses
X	Display Addresses
+	Disassemble Program

Any other key entry will show a "?" on the screen. When you enter a "?" from the terminal or computer keyboard, a help menu will appear.

Baud E

Press the BAUD key to enter a new data transfer rate for the RS-232 port. When you first turn on the Trainer, the RS-232 port is automatically set for 1200 with 1 start bit, 8 data bits, no parity bit, and 1 stop bit. Other available baud rates are 50, 100, 150, 300, and 600 baud.

PRESS	DISPLAY READS
Baud	Type in new Baud: 0000

Insert F

Press the INSERT key to change the contents of a memory location. This mode works almost the same as Examine Memory. The difference occurs when you enter a new value from the keyboard into RAM. The values from the current address to BFFFH are shifted up one byte in memory. The byte previously at BFFFH is lost. Data in memory locations 0000H - 6820H and C000H - FFFFH is unaffected. Do not use the Insert Memory mode below memory location 6821H; this can produce unpredictable results. NOTE: When you change the contents of a memory location within a program, you must correct any jump destination addresses that are above the inserted address which are affected by the change.

PRESS	DISPLAY READS
Insert	F Look Address: 7000H
7000	Look Address: 7000H 7000: 40 @

Load A

Press the LOAD key to transfer a file stored in the Memory Module to RAM located in the CPU Module. When you select this command, the Trainer asks for a file digit (0-F) that corresponds to the file digit the program was saved under. After you enter a file digit, the display shows the address it will be loaded at. The file is then copied, checksums are compared, and the resulting checksum is displayed.

PRESS	DISPLAY READS
Load	A File digit?
Help	File Digit? type hex digit 0-F

If the file cannot be read, the display will read:

File digit 0-F	File digit ? Error: Load Error

If the file is found, the display will read:

File digit 0-F	Copying to 0000H Verifying Checksum

Save B

Press the SAVE file key to transfer a file to the Memory Module from the CPU Module RAM. The Trainer asks for a start address, number of bytes in the file, and a file digit (0-F). The file is then copied from RAM to the cartridge and a checksum is formed and stored on the cartridge with the start address and number of bytes.

PRESS	DISPLAY READS
Save	B Source: 7000 H
Enter start address	Source: 7000 H # Bytes: 0000 H
Enter number of bytes	# Bytes: 0010 H File Digit ?
Help	File digit ? type hex digit 0-F
Enter file digit	Copying... Verifing Checksum
	Checksum= 07F8

Dup C

Press the DUP (duplicate) key to copy a file from one Memory Module to another. Help prompts take you through each step. First insert the cartridge to copy from and then copy your file into RAM. Next, remove the cartridge and insert the Memory Module you want to copy to. Press any key and the file is copied to the Memory Module. NOTE: Be sure to save all designed programs before using the Dup mode. The copying process overwrites user memory locations 7000H to B000H.

PRESS	DISPLAY READS
Dup	Insert source press a key.
Enter file digit 0-F	Press a key. Copying to RAM...
	Insert destination press a key.
	press a key. Copying to Cartridge!
	Dup complete: 491F

Break 7

Press the BREAK key to set a breakpoint in memory. A breakpoint stops program execution and transfers control back to the keypad or terminal in the Single Step mode if not already in that mode. At this point you can examine memory, examine registers, set another breakpoint, or GO again. The Trainer asks for an address where you want to set a breakpoint. If you press the LIST key the currently set breakpoints are listed. After you specify a hexadecimal address, the breakpoint is added to the breakpoint table. If the address is already in the table, the previous one will be removed. You can set up to 16 different breakpoints at a time. If you add a breakpoint that is already in the table, the old one will be deleted. Refer to Page 30 for a more detailed discussion of breakpoints.

W Reg 8

Press the W REG key to break a program when a register reaches a desired value in the Single Step mode. Press the 0 key to scroll through a list of registers. Press the 8 key to select the register to watch. Press the LIST key to list the currently set watch registers. After selecting a register to watch, select a value to watch for.

PRESS	DISPLAY READS
W Reg	Watch Register: ACC?
Help	ACC? list,8=select,0=next

W Loc 9

Press the W LOC key to break a program when an address location reaches a desired value in the Single Step mode. This mode is similar to W Reg.

PRESS	DISPLAY READS
W Loc	Watch Address: 7000H Watch Value 00 H

M Blk 4

Press the M BLK key to move a block of data in memory to another location in memory. Following the prompts, enter the starting address, number of bytes in the block to move and the destination address. Before the data is moved, the Trainer checks the destination range to make sure that a write does not destroy a source byte. If a source byte already resides in that range, the block of data will copy from the high memory address back down to the low memory address. Keys 0-F are shifted in from the right for each field. RPO moves to the next field and NMI quits the operation.

PRESS	DISPLAY READS
M Blk	4 Source 0000H
	Source ____H # Bytes ____H Dest ____H
	Copy complete >

I Blk 5

Press the I BLK key to set a block of memory to a desired value. First enter a start address and then an end address. Next, set the block of memory equal to a value. The hexadecimal value of the block is displayed along with the ASCII version. Use the 0 to F keys to enter all values. Press the NMI key to exit this mode. Do not copy below memory location 6821H; this can produce unpredictable results.

PRESS	DISPLAY READS
I Brk	Start Address ____H End Address ____H
	End Address Byte = 42 H B

Down 6

Press the DOWN key to download an INTEL HEX file for storing in RAM. You may specify an offset for easier relocating of the file to another location in RAM or when copying a program into a Memory Module after it has been downloaded into RAM.

Exm Mem 1

Pressing the EXM MEM key in the select operation mode lets you examine the contents of any memory or I/O location. The Trainer first asks you to enter either a "1" or a "2" from the keypad. Press the "1" key if you want to examine memory locations or the "2" key if you want to examine I/O locations.

PRESS	DISPLAY READS
Exm Mem	1 - memory 2 - I/O memory

If you select "1" to examine memory locations, the Trainer asks for a 4-digit hexadecimal starting address. The display then shows the value of the byte at that address in hexadecimal form and ASCII characters if they are displayable.

PRESS	DISPLAY READS
1	 Look Address: 7000H
7000	Look Address: 7000H 7000: 41 A

If you select "2" to examine I/O memory locations, the Trainer asks for a 2-digit hexadecimal starting I/O address. The display then shows the value of the byte at that address in hexadecimal form and ASCII characters if they are displayable.

PRESS	DISPLAY READS
2	 I/O Look Address: 30
30	I/O Look Address: 30 30: 40@

To change the current byte, simply enter any key 0 to F and each value will be shifted in on the low byte value. After you press two keys, the address is automatically stepped to the next address. To move to the next address, press the plus (+) key. To move to the previous address, press the minus (–) key. To quit, press the RPO key on the Trainer or the "?" key if you are using a terminal.

Exm Reg 2

Pressing the EXM REG key in the select operation mode lets you examine and modify the CPU registers. Enter a desired register value by pressing the 0 to F keys, which are shifted into the low order value of that register. Press the plus (+) key to display the contents of the next register. Press the minus (–) key to display the contents of the previous register. Press the RPO key to end register examination.

PRESS	DISPLAY READS
Exm Reg	2 ACC= 00__ H

+	ACC= 00 H
	FLAG= 00__ H

+	FLAG= 00 H
	RegBC= 00 00 H

+	RegBC= 0000 H
	RegDE= 0000H

+	RegDE= 0000H
	RegHL= 0000H

+	RegHL= 0000H
	PC= 7000H

+	PC= 7000H
	SP= 6FFFH

+	SP= 6FFFH
	RegI= 00 H

+
RPO
	RegI= 00 H
	ACC = 00 H>

Go 3

Press the GO key to run a program stored in memory. Initially, the default start address (7000H) is displayed. You then select a start address. The Trainer jumps to that address and executes the program. If it encounters a breakpoint, the program halts and automatically enters the Single Step mode. From here you can either press GO again or quit by pressing the Return (–) key. When GO is run from the single step mode, the program will run slower due to the extra time required to check watch points and watch registers.

PRESS	DISPLAY READS
Go	3 Go Address: 7000H

If a breakpoint is encountered the next address is displayed.

	7002 C3 00 70 JMP 7000

If a watch point is encountered the watch address or register is displayed.

	JMP 7000 Address 7100H= 55 H

SS 0

Press the SS key to single step through a program stored in memory. Enter a start address and the Trainer displays the instruction to be executed at that address. You can then single step to the next step, press GO, examine registers, examine or insert memory, or set breakpoints, watch registers, or watch locations. Press the RETURN key to quit at any time. All registers and RAM values are saved at their present values.

PRESS	DISPLAY READS
SS	0 Step Address: 7000H
0000	7000 29 DAD H

After the instruction is executed:

SS	7001 94 SUB H
Help	SS ,GO, Exm Rg,Exm M Brk, Watch, NMI quit

NMI

Press the NMI (Non-Maskable Interrupt) key to interrupt any program or operation being performed on the Trainer. This function does not alter the contents of RAM and saves all register values. If the Trainer was executing a program in real time (i.e. you pressed GO from the menu and not from Single Step), then these values can be checked by the Examine Register function. In addition, the Interrupt Vector Table is not recopied to RAM, allowing you to interrupt your program and examine this table.

PRESS	DISPLAY READS
NMI	Heath/Zenith 80C85A Educational MPU V1.0

The Monitor's NMI can be replaced with your own non-maskable interrupt routine. To replace the Monitor's NMI, you must change the jump address at 680CH - 680EH to the address of your routine. Additional NMIs can also be obtained by clearing the TRAP input to microprocessor U11. The input can be cleared either by writing any character to the LCD display or entering a "IN B1H" (DBH B1H) instruction.

Reset

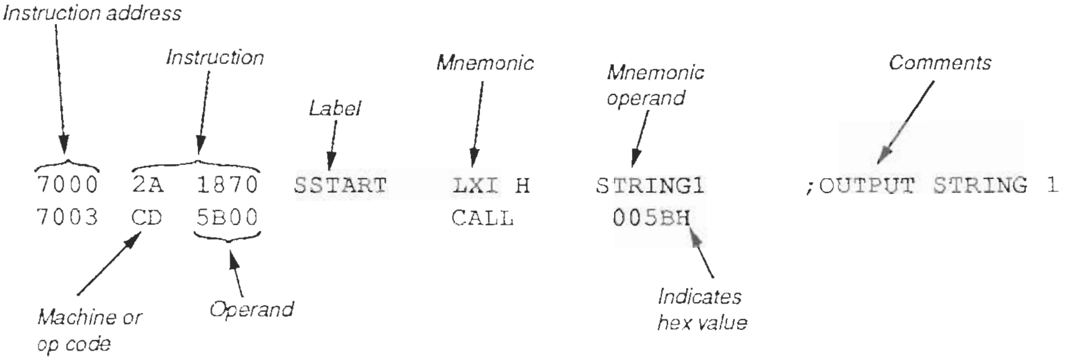
Press the RESET key to reinitialize the Trainer to its power-up state. The Interrupt Vector Table is initialized to its default values. All breakpoints, watch locations, and watch registers are erased.

PROGRAMMING

ENTERING PROGRAMS

Pictorial 3 shows the first two instructions of Sample Program 1 (on Page 18) and indicates the various information they contain. This information is further described in the following paragraphs.

Instruction Address: This is the location of the Opcode to be executed. In order to perform an instruction, the Program Counter must contain the address that is in this column. Breakpoints are not recognized except at instruction addresses.



```
7000  2A  1870  SSTART  LXI H   STRING1   ;OUTPUT STRING 1
7003  CD  5B00          CALL    005BH
```

PICTORIAL 3

Instruction: This is one, two, or three bytes of data as required by the addressing mode used.

Op code: This is an information byte referred to as machine code. It indicates in hexadecimal form the operation to be performed.

Operand: This is additional hexadecimal information needed to perform the operation. It may be zero, one, or two bytes as determined by the addressing mode. The least significant byte is listed first followed by the most significant digit.

Label: This is usually a name applied to a subroutine in the program used more than once. In the sample programs, the address to be entered to begin execution is labeled "Start."

Mnemonic: This is a two-, three-, or four-letter indication of the source instruction.

Mnemonic operand: Again, this is additional information that is required for the operation. It may be a label, address, or data. The "H" sign indicates that the information is a hexadecimal value.

Comments: This is a brief description of what is happening in the program. It makes the program easier to read but has no effect on the program. Comments on most 8085 assemblers begin with a semi-colon.

When you load a program into the Trainer, only the one, two, or three bytes of each instruction are entered. If you make an entry error, press the – key. Pressing the – key backs you up one address at a time. Use the Insert key to add instructions within your program. Pressing the RPO key (or terminal "?" key) returns control back to the Trainer. Or, remember where the error was made and continue entering the program. After you have finished entering the program, correct your error by examining that memory location and changing the entry.

Sample Programs

These sample programs will give you practice entering programs and show the use of several Monitor subroutines.

Sample Program 1

The following program demonstrates how you can use two simple subroutine calls to create and display a message on the Trainer's LCD. The subroutine located at 005BH outputs a string of characters starting at the location stored in the "HL" register of the CPU until it encounters a zero. Subroutine 0067H delays the program for approximately 1 second every time it is called. The last call to subroutine instruction (CALL 0040H) returns control back to the Monitor ROM.

```
7000  21   1870    SSTART: LXI H,  STRING1     ;OUTPUT STRING 1
7003  CD   5B00            CALL    005BH
7006  CD   6700            CALL    0067H       ;PAUSE
7009  21   2770            LXI H,  STRING2     ;OUTPUT STRING 2
700C  CD   5B00            CALL    005BH
700F  CD   6700            CALL    0067H       :PAUSE
7012  CD   6700            CALL    0067H       ;PAUSE
7015  CD   4000            CALL    0040H       ;RETURN TO MONITOR
7018  50726F6772 STRING1 DB                    'Programming is'
      616D6D696E
      67206973
7026  00                 DB    0
7027  4561737920 STRING2 DB                    'Easy and Fun!!!
      616E642046
      756E212121
7036  00                 DB    0
                         END
SYMBOL TABLE:

SSTART  7000H    STRING1 7018 H    STRING2  7027 H
```

Sample Program 2

The following program is for a real time clock. The program first disables the interrupt so that you can enter the current time by using the plus (+) and minus (–) keys on the Trainer. As soon as you enter the last digit of the time, the interrupt will be enabled and the clock will start. Due to the use of interrupts in the clock program, RS-232 capability is disabled while this program is running.

```
7000  21 7C 71   SSTART: LXI H,   STRING1     ;OUTPUT STRING 1
7003  CD 5B 00           CALL     005BH
7006  3E 01              MVI A,   1           ;GET HOURS TENS DIGIT,
7008                                          ;SET A TO MAXIMUM VALUE
7008  CD 29 71           CALL     GETTIM
700B  CD 18 71           CALL     MUL10       ;MULTIPLY 10 TO A
700E  32 75 71           STA      HOURS       ;STORE HOURS
7011  06 0A              MVI B,   10
7013  B8                 CMP B                ;CHECK FOR 10,11,12
7014  C2 1C 70           JNZ      HNOTTEN
7017  3E 02              MVI A,   2           ;SINCE FIRST DIGIT WAS ONE
7019  C3 1E 70           JMP      GETHOUR1    ;MAX DIGIT IS 2
701C  3E 09      HNOTTEN MVI A,   9           ;SINCE FIRST DIGIT WAS NOT 1
701E  CD 29 71   GETHOUR1 CALL    GETTIM
7021  21 75 71           LXI H,   HOURS
7024  86                 ADD M                ;ADD IN HIGH DIGIT
7025  77                 MOV M,   A           ;STORE RESULT
7026  3E 3A              MVI A,   3AH
7028  CD 46 00           CALL     0046H       ;MOVE TO NEXT POSITION
702B  3E 05              MVI A,   5
702D  CD 29 71           CALL     GETTIM      ;GET MINUTES
7030  CD 18 71           CALL     MUL10       ;MULTIPLY 10 TO A
7033  32 76 71           STA      MINUTES     ;SAVE IT
7036  3E 09              MVI A,   9
7038  CD 29 71           CALL     GETTIM      ;GET LOW NIBBLE
703B  21 76 71           LXI H,   MINUTES
703E  86                 ADD M                ;ADD IN HIGH DIGIT
703F  77                 MOV M,   A
7040  3E 3A              MVI A,   3AH         ;MOVE TO NEXT POSITION
7042  CD 46 00           CALL     0046H
7045  3E 05              MVI A,   5
7047  CD 29 71           CALL     GETTIM      ;GET SECONDS
704A  CD 18 71           CALL     MUL10       ;MULTIPLY 10 TO A
704D  32 77 71           STA      SECONDS     ;SAVE HIGH NIBBLE OF SECONDS
7050  3E 09              MVI A,   9
7052  CD 29 71           CALL     GETTIM      ;GET LOW NIBBLE
7055  21 77 71           LXI H,   SECONDS
7058  86                 ADD M                ;ADD IN HIGH NIBBLE
7059  77                 MOV M,   A           ;SAVE IT
705A  3E 28              MVI A,   40          ;LOAD "FOURTY" WITH 40 DECIMAL
705C  32 78 71           STA      FOURTY      ;
705F                                          ;Current time is saved, set interrupt
                                              vector
705F                                          ;uses timer T2 and R 7.5
705F  F3                 DI                   ;MASK INTERRUPTS TEMPORARILY
7060  21 82 70           LXI H,   T1HZ
7063  22 1F 68           SHLD     681FH       ;SET UP INTERRUPT JUMP TABLE
7066  20                 RIM                  ;ENABLE 7.5 INTERRUPT
7067  E6 03              ANI      03H
7069  F6 08              ORI      08H
706B  30                 SIM
```

```
706C  21 A1 71            LXI H,   STRING3    ;OUTPUT STRING3
706F  CD 5B 00            CALL     005BH
7072
7072                                          ;START CLOCK
7072
7072  3E B6               MVI A,   0B6H       ;SET TIMER 2 TO MODE 3 SQUARE WAVE
                                              GENERATOR
7074  D3 43               OUT      43H
7076  3E 00               MVI A,   00H        ;LEAST SIGNIFICANT BYTE IS LOADED IN
                                              TIMER 2
7078  D3 42               OUT      42H
707A  3E B4               MVI A,   0B4H       ;A LOADED WITH B4H
707C  D3 42               OUT      42H        ;MOST SIGNIFICANT BYTE IS LOADED IN
                                              TIMER 2
707E  FB                  EI                  ;START INTERRUPTS
707F  C3 7F 70   WAIT     JMP      WAIT       ;WAIT FOR INTERRUPT
7082                                          ;***********************************
7082                                          ;     ** TIMER INTERRUPT ROUTINE **
7082                                          ;***********************************
7082  F3         T1HZ     DI                  'DISABLE INTERRUPTS TEMPORARILY
7083  F5                  PUSH PSW            ;SAVE ACCUMULATOR AND CPU STATUS
7084  C5                  PUSH B
7085  E5                  PUSH H              ;SAVE HL REGISTER PAIR ON STACK
7086
7086  21 78 71            LXI H,   FOURTY
7089  7E                  MOV A,M             ;GET FOURTY COUNTER
708A  35                  DCR M               ; AND DECREMENT IT
708B  C2 DF 70            JNZ      RETURN
708C  36 28               MVI M,   40         ;RELOAD FOURTY COUNT WITH 40
708D  3A 77 71            LDA      SECONDS    ;GET SECONDS
7093  3C                  INR A               ;INCREMENT IT
7094  06 3C               MVI B,   60
7096  B8                  CMP B               ;CHECK FOR OVERFLOW
7097  DA BA 70            JC       NSECOV     ;IF NO OVERFLOW SAVE SECONDS ONLY
709A  3A 76 71            LDA      MINUTES    ;GET MINUTES
709D  3C                  INR A               ;INCREMENT IT
709E  06 3C               MVI B,   60
70D0  B8                  CMP B
70A1  DA B5 70            JC       NMINOV     ;IF NO OVERFLOW SAVE MINUTES AND
70A4  3A 75 71            LDA      HOURS      ;GET HOURS
70A7  3C                  INR A               ;INCREMENT IT
70A8  06 0C               MVI B, 12
70AA  B8                  CMP B               ;CHECK FOR OVERFLOW
70AB  DA B0 70            JC       NHOROV     ;IF NO OVERFLOW SAVE MINUTES
70AE  3E 01               MVI A,   1          ;SET TO 1 O'CLOCK
70B0  32 75 71   NHOROV   STA      HOURS      ;SAME HOURS
70B3  3E 00               MVI A,   0          ;CLEAR MINUTES AND FALL THRU
70B5  32 76 71   NMINOV   STA      MINUTES    ;SAVE MINUTES
70B8  3E 00               MVI A,   0          ;CLEAR SECONDS AND FALL THRU
70BA  32 77 71   NSECOV   STA      SECONDS    ;SAVE SECONDS
70BD  21 98 71            LXI H,   STRING2
70C0  CD 5B 00            CALL     005BH      ;BACKUP TO START OF DISPLAY
70C3  3A 75 71            LDA      HOURS      ;GET HOURS AND OUTPUT
70C6  CD E4 70            CALL     DECIMAL
70C9  3E 3A               MVI A,   3AH        ;OUTPUT COLON
70CB  CD 46 00            CALL     0046H      ;PUT CHARACTER
70CE  3A 76 71            LDA      MINUTES    ;GET MINUTES AND OUTPUT
70D1  CD E4 70            CALL     DECIMAL
```

```
70D4  3E 3A                MVI A,   3AH       ;OUTPUT COLON
70D6  CD 46 00             CALL     0046H     ;PUT CHARACTER
70D9  3A 77 71             LDA      SECONDS   ;GET SECONDS AND OUTPUT
70DC  CD E4 70             CALL     DECIMAL
70DF             RETURN                       ;RETURN FROM INTERRUPT
70DF  E1                   POP H
70F0  C1                   POP B
70E1  F1                   POP PSW
70E2  FB                   EI
70E3  C9                   RET
70E4                                          ;SUBROUTINE WHICH TRANSLATES NUMBER IN
                                              A TO BCD AND DISPLAYS
70E4  06 0A      DECIMAL   MVI B,   0AH
70E6  B8                   CMP B
70E7  DA 14 71             JC       DECOUT
70EA  C6 06                ADI      6         ;ADD 6 TO NUMBER
70EC  06 1A                MVI B,   1AH
70EE  B8                   CMP B
70EF  DA 14 71             JC       DECOUT
70F2  C6 06                ADI      6         ;ADD ANOTHER 6 TO NUMBER
70F4  06 2A                MVI B,   2AH
70F6  B8                   CMP B
70F7  DA 14 71             JC       DECOUT
70FA  C6 06                ADI      6         ;ADD ANOTHER 6 TO NUMBER
70FC  06 3A                MVI B,   3AH
70FE  B8                   CMP B
70FF  DA 14 71             JC       DECOUT
7102  C6 06                ADI      6         ;ADD FOURTH 6 TO NUMBER
7104  06 4A                MVI B,   4AH
7106  B8                   CMP B
7107  DA 14 71             JC       DECOUT
710A  C6 06                ADI      6         ;ADD FIFTH 6 TO NUMBER
710C  06 5A                MVI B,   5AH
710E  B8                   CMP B
710F  DA 14 71             JC       DECOUT
7112  C6 06                ADI      6         ;ADD LAST 6 TO NUMBER
7114  CD 55 00   DECOUT    CALL     0055H     ;PRINT OUT BYTE IN A
7117  C9                   RET
7118                                          ; END OF DECIMAL OUT SUBROUTINE
7118                                          ; MULTIPLY 10 TO ACCUMULATOR
7118  37                   MUL10    STC
7119  3F                   CMC
711A  17                   RAL
711B  32 79 71             STA      SHL1
711E  37                   STC
711F  3F                   CMC
7120  17                   RAL
7121  37                   STC
7122  3F                   CMC
7123  17                   RAL
7124  21 79 71             LXI H,   SHL1
7127  86                   ADD M
7128  C9                   RET
7129                                          ;END OF MULTIPLY BY 10
7129                                          ;
7129                                          ;GET DIGIT
7129  32 7A 71   GETTIM    STA      MDIGIT    ;SAVE MAXIMUM VALUE
712C  21 7B 71             LXI H,   CURTIM
```

```
712F  3E 00                MVI A,  0
7131  77                   MOV M,A          ;SET CURTIM AND A INITIALLY TO ZERO
7132  3E 30                MVI A,  30H      ;CONVERT TO ASCII
7134  CD 46 00             CALL    0046H    ;PUTC
7137  3E 08                MVI A,  08H      ;BACK UP SPACE
7139  CD 46 00             CALL    0046H    ;PUTC
713C  CD 43 00   REPTIM    CALL    0043H    ;GET CHARACTER (GETC)
713F  06 2D                MVI B,  2DH
7141  B8                   CMP B            ;IF MINUS RETURN
7142  C2 51 71             JNZ     GTNOR
7145  3A 7B 71             LDA     CURTIM   ;WRITE OUT DIGIT
7148  C6 30                ADI     30H      ;CONVERT TO ASCII
714A  CD 46 00             CALL    0046H    ;PUTC
714D  3A 7B 71             LDA     CURTIM   ;GET SELECTED DIGIT AND RETURN
7150  C9                   RET
7151  06 2B      GTNOR     MVI B,  2BH      ;IF PLUS INCREMENT CURTIM
7153  B8                   CMP B
7154  C2 3C 71             JNZ     REPTIM   ;IGNORE  CHARACTER  GO  BACK  &  GET  NEW
                                            ONE
7157  21 7B 71             LXI H,  CURTIM
715A  3A 7A 71             LDA     MDIGIT   ;INCREMENTS CURTIM
715D  34                   INR M
715E  BE                   CMP M            ;COMPARE TO MAX
715F  D2 64 71             JNC     VALOK    ;IF CURTIM <= MDIGIT TAKE IT
7162  36 00                MVI M,  0        ;PUTS CURTIM DOWN TO 0
7164  7E         VALOK     MOV A,M          ;PUTS CURTIM IN A
7165  C6 30                ADI     30H      ;CONVERT TO ASCII
7167  CD 46 00             CALL    0046H    ;PUTC
716A  3E 08                MVI A,  08H
716C  CD 46 00             CALL    0046H    ;PUTC BACKSPACE
716F  C3 3C 71             JMP     REPTIM   ;REPEAT UNTIL '-' TYPED
7172
7172  CD 40 00             CALL    0040H    ;End of Program
7175  00         HOURS     DB      0
7176  00         MINUTES   DB      0
7177  00         SECONDS   DB      0
7178  00         FOURTY    DB      0
7179  00         SHL1      DB      0
717A  00         MDIGIT    DB      0
717B  00         CURTIM    DB      0
717C  45 6E 74 65 72STRING1 DB              'Enter Current Time:'
      20 43 75 72 72
      65 6E 74 20 54
      69 6D 65 3A
718F  0D                   DB      0DH
7190  5F 5F 3A 5F 5F       DB               '__:__:__'
      3A 5F 5F
7198  08 08 08 08 STRING2  DB      08H,08H,08H,08H
719C  08 08 08 08          DB      08H,08H,08H,08H
71A0  00                   DB      0
71A1  0D          STRING3  DB      0DH
71A2  20 20 20 20 54       DB               '          The Time is:'
      68 65 20 54 69
      6D 65 20 69 73
      3A
71B2  0D                   DB      0DH
```

```
71B3  20 20 20 20 20      DB               '         __:__:__'
      20 5F 5F 3A 5F
      5F 3A 5F 5F
71C1  00                  DB       0
71C2                      END

SYMBOL TABLE:

CURTIM    717B H    DECIMAL   70E4 H    DECOUT    7114 H
FOURTY    7178 H    GETHOUR1  701E H    GETTIM    7129 H
GTNOR     7151 H    HNOTTEN   701C H    HOURS     7175 H
MDIGIT    717A H    MINUTES   7176 H    MUL10     7118 H
NHOROV    70B0 H    NMINOV    70B5 H    NSECOV    70BA H
REPTIM    713C H    RETURN    70DF H    SECONDS   7177 H
SHL1      7179 H    SSTART    7000 H    STRING1   717C H
STRING2   7198 H    STRING3   71A1 H    T1HZ      7083 H
VALOK     7164 H    WAIT      707F H
```

Sample Program 3

This program tests the user memory and stack area. It is a simple two pass test in which a continuous count is stored in consecutive memory locations and then verified. One incrementing count and one decrementing count is used.

```
7000  21 DE 70   SSTART: LXI H,   STRING1     ;OUTPUT STRING 1
7003  CD 5B 00           CALL     005BH
7006  16 00              MVI D,   00H
7008  21 21 68           LXI H,   6821H
700B  14         TEST1A  INR D                ;LOOPS FROM 6821H TO BFFFH SKIPPING
                                              OVER ITSELF
700C  72                 MOV M,D              ;CONTINUOUSLY STORING AN INCREMENTAL
700D  23                 INX H                ;COUNT IN CONSECUTIVE MEMORY
700E  3E 70              MVI A,   70H
7010  BC                 CMP H
7011  C2 0B 70           JNZ      TEST1A
7014  3E 00              MVI A,   00H
7016  BD                 CMP L
7017  C2 0B 70           JNZ      TEST1A
701A  21 38 71           LXI H,   PRGEND      ;SKIPS OVER MEMORY TEST PROGRAM AREA
701D  14         TEST1B  INR D
701E  72                 MOV M,D
701F  23                 INX H
7020  3E BF              MVI A,   0BFH
7022  BC                 CMP H
7023  C2 1D 70           JNZ      TEST1B
7026  3E FF              MVI A,   0FFH
7028  BD                 CMP L
7029  C2 1D 70           JNZ      TEST1B
702C                                          ;VERIFY CORRECT CONTENTS OF MEMORY
702C  16 00              MVI D,   00H
702E  21 21 68           LXI H,   6821H
7031  14         TEST1C  INR D
7032  7E                 MOV A,M
7033  BA                 CMP D
7034  C2 CB 70           JNZ      ERROR       ;TEST FOR A MATCH IF NOT ERROR
7037  23                 INX H
7038  3E 70              MVI A,   70H
703A  BC                 CMP H
703B  C2 31 70           JNZ      TEST1C
```

```
703E   3E 00               MVI A,   00H
7040   BD                  CMP L
7041   C2 31 70            JNZ      TEST1C
7044   21 38 71            LXI H,   PRGEND
7047   14         TEST1D   INR D
7048   7E                  MOV A,M
7049   BA                  CMP D
704A   C2 CB 70            JNZ      ERROR       ;TEST FOR A MATCH IF NOT ERROR
704D   23                  INX H
704E   3E BF               MVI A,   0BFH
7050   BC                  CMP H
7051   C2 47 70            JNZ      TEST1D
7054   3E FF               MVI A,    0FFH
7056   BD                  CMP L
7057   C2 47 70            JNZ      TEST1D
705A                                            ;FIRST TEST COMPLETE
705A   21 ED 70            LXI H,   STRING2
705D   CD 5B 00            CALL     005BH
7060                                            ;START OF SECOND PASS TEST
7060   16 00               MVI D,   00H
7062   21 FF BF            LXI H,   0BFFFH
7065   14         TEST2A   INR D                ;LOOPS  FROM  BFFFH  TO  6821H  SKIPPING
                                                OVER ITSELF
7066   72                  MOV M,D              ;CONTINUOUSLY STORING AN INCREMENTAL
7067   2B                  DCX H                ;COUNT IN CONSECUTIVE MEMORY
7068   01 37 71            LXI B,   PRGEND-1
706B   78                  MOV A,B
706C   BC                  CMP H
706D   C2 65 70            JNZ      TEST2A
7070   79                  MOV A,C
7071   BD                  CMP L
7072   C2 65 70            JNZ      TEST2A
7075   21 FF 6F            LXI H,   6FFFH       ;SKIPS OVER MEMORY TEST PROGRAM AREA
7078   14         TEST2B   INR D
7079   72                  MOV M,D
707A   2B                  DCX H
707B   3E 68               MVI A,   68H
707D   BC                  CMP H
707E   C2 78 70            JNZ      TEST2B
7081   3E 20               MVI A,   20H
7083   BD                  CMP L
7084   C2 78 70            JNZ      TEST2B
7087                                            ;VERIFY CORRECT CONTENTS OF MEMORY
7087   16 00               MVI D,   00H
7089   21 FF BF            LXI H,   0BFFFH
708C   14         TEST2C   INR D
708D   7E                  MOV A,M
708E   BA                  CMP D
708F   C2 CB 70            JNZ      ERROR       ;TEST FOR A MATCH IF NOT ERROR
7092   2B                  DCX H
7093   01 37 71            LXI B,   PRGEND-1
7096   78                  MOV A,B
```

```
7097   BC                        CMP H
7098   C2 8C 70                  JNZ      TEST2C
709B   79                        MOV A,C
709C   BD                        CMP L
709D   C2 8C 70                  JNZ      TEST2C
70A0   21 FF 6F                  LXI H,   6FFFH
70A3   14              TEST2D    INR D
70A4   7E                        MOV A,M
70A5   BA                        CMP D
70A6   C2 CB 70                  JNZ      ERROR        ;TEST FOR A MATCH IF NOT ERROR
70A9   2B                        DCX H
70AA   3E 68                     MVI A,   068H
70AC   BC                        CMP H
70AD   C2 A3 70                  JNZ      TEST2D
70B0   3E 20                     MVI A,   020H
70B2   BD                        CMP L
70B3   C2 A3 70                  JNZ      TEST2D
70B6                                                   ;END OF SECOND PASS
70B6   21 00 71                  LXI H,   STRING3
70B9   CD 5B 00                  CALL     005BH
70BC   CD 67 00                  CALL     0067H
70BF   21 13 71                  LXI H,   STRING4
70C2   CD 5B 00                  CALL     005BH
70C5   CD 67 00                  CALL     0067H
70C8   CD 40 00                  CALL     0040H        ;RETURN CONTROL BACK TO MONITOR
70CB   54                        ERROR    MOV D,H      ;ERROR HANDLER
70CC   5D                        MOV E,L
70CD   21 26 71                  LXI H,   ERRSTR
70D0   CD 5B 00                  CALL     005BH
70D3   62                        MOV H,D
70D4   6B                        MOV L,E
70D5   CD 58 00                  CALL     0058H
70D8   CD 67 00                  CALL     0067H
70DB   C3 DB 70        LOOP      JMP      LOOP
70DE   4D 65 6D
       6F 72           STRING1 DB                      'Memory Test...'
       20 54 65 73
       74 2E 2E 2E
70EC   00                        DB       0
70ED   0D              STRING2 DB         0DH
70EE   50 61 73 73 20            DB                    'Pass one complete'
       6F 6E 65 20 63
       6F 6D 70 6C 65
       74 65
70FF   00                        DB       0
7100   0D              STRING3 DB         0DH
7101   50 61 73 73 20            DB                    'Pass two complete'
       74 77 6F 20 63
       6F 6D 70 6C 65
       74 65
7112   00                        DB       0
7113   0D              STRING4 DB         0DH
```

```
7114  4D 65 6D 6F 72        DB                'Memory check O.K.'
      79 20 63 68 65
      63 6B 20 4F 2E
      4B 2E
7125  00                    DB        0
7126  0D           ERRSTR   DB        0DH
7127  4D 45 4D 4F 52        DB                'MEMORY ERROR AT:'
      59 20 45 52 52
      4F 52 20 41 54
      3A
7137  00                    DB        0
7138  00           PRGEND   DB        0
7139                        END

SYMBOL TABLE:

ERROR     70CB H    ERRSTR    7126 H    LOOP      701B H
PRGEND    7138 H    SSTART    7000 H    STRING1   70DE H
STRING2   70ED H    STRING3   7100 H    STRING4   7113 H
TEST1A    700B H    TEST1B    701B H    TEST1C    7031 H
TEST1D    7047 H    TEST2A    7065 H    TEST2B    7078 H
TEST2C    708C H    TEST2D    70A3 H
```

Sample Program 4

The following program lets you convert your Trainer into a digital DC voltmeter. This program uses the CPU Module's A/D Converter and other circuitry to monitor and display a range of positive and negative input voltages. A total resistance of 380 kΩ in series with the voltmeter's input (the analog AI0 line) reduces the voltage input to the full scale range of the A/D Converter.

This program is written so that the voltmeter can measure voltages ranging from +12.0 to -12.0 VDC full scale. The displayed voltage is determined by multiplying the input to the A/D Converter by the fraction 15/16. This is done by multiplying the input by 15 at address 7020H (0F hex value) and doing 4 LSRs (logical shift rights) at address 7025H to divide by 16.

To change the voltage range of this voltmeter, you must change the analog input line and/or the series resistor and alter the program. Use the following table for the proper series resistor and program alterations. Enter the hex value of the numerator at address 7020H and the number of LSRs to get the correct denominator at address 7025H.

VOLTAGE FULL SCALE READING (V_{fs})	A/D RANGE RANGE BIT	EXTERNAL SERIES RESISTANCE (R_{ES})	INTERNAL SERIES RESISTANCE (R_{IS}+1%)	A/D INPUT LINE(S)	MULTIPLYING FRACTION
–0.0125 to +0.0125	Clear	None	1kΩ	AI2	25/256
–0.025 to +0.025	Set	None	1kΩ	AI2	25/128*
–0.125 to +0.125	Clear	None	10kΩ	AI1	25/256**
–0.25 to +0.25	Set	None	10kΩ	AI1	25/128**
–1.25 to +1.25	Clear	None	100kΩ	AI0,AI3,AI4,AI5	125/128**
–2.5 to +2.5	Set	None	100kΩ	AI0,AI3,AI4,AI5	25/128
–6.0 to +6.0	Clear	380kΩ	100kΩ	AI0,AI3,AI4,AI5	15/32
–12.0 to +12.0	Set	380kΩ	100kΩ	AI0,AI3,AI4,AI5	15/16
–7.5 to +7.5	Clear	None	604kΩ	AI6,AI7	151/256
–15.1 to +15.1	Set	None	604kΩ	AI6,AI7	151/128

*Display is in thousandths of volts.
**Display is in hundredths of volts.

To determine other series resistors and fractions when the full scale voltage is known, use these two equations:

$R_{ES} = 20V_{fs} - R_{IS}$

FRACTION = 10 X (V_{fs}/256), when display is in tenths of volts.

FRACTION = 100 X (V_{fs}/256), when display is in hundredths of volts.

FRACTION = 1000 X (V_{fs}/256), when display is in thousandths of volts.

To determine full scale voltage when R_{ES} is known, use the following equation:

$V_{fs} = (R_{ES}/20) + (R_{IS}/20)$

```
7000                                               ;ETC-8085 HEATH VOLTMETER
7000                                               ;
7000  21 7C 70   SSTART    LXI H,    STRING1       ;PRINTS OUT FIRST STRING
7003  CD 5B 00             CALL      005BH         ;PUTSTRING
7006  3E F8                MVI A,    0F8H          ;SETS INPUT TO AI0
7008  D3 30                OUT       30H
700A                                               ;
700A  21 AD 70   MAINLOP   LXI H,    STRING2       ;BACKUP TO START OF DISPLAY
700D  CD 5B 00             CALL      005BH         ;PUTSTRING
7010  DB 80                IN        80H           ;GETS A/D VALUE
7012  06 00                MVI B,    0
7014  B8                   CMP B
7015  FA 38 70             JM        SIGNPOS       ;JUMP IF VOLTAGE IS POSITIVE
7018  47                   MOV B,A
7019  3E 2D                MVI A,    2DH
701B  CD 46 00   OUTVAL    CALL      0046H         ;OUTPUT PLUS OR MINUS SIGN
701E                                               ;DO SCALLING FACTOR
701E  48                   MOV C,B
701F  2E 0F                MVI L,    15
7021  CD 40 70             CALL      MULTIPLY      ;MULTIPLY B TIMES 15 RESULT IN H,L
7024  0E 04                MVI C,    4             ;ROTATED RIGHT 4 TIMES OR DIVIDE BY 64
7026  CD 6B 70             CALL      ROTR
7029  CD 6D 00             CALL      006DH         ;PRINT IN DECIMAL
702C                                               ;DELAY LOOP
702C  01 7A 1D             LXI B,                  1D7AH
702F  0B         DLOOP     DCX B
7030  78                   MOV A,B
7031  B1                   ORA C
7032  C2 2F 70             JNZ D     LOOP
7035                                               ;END OF DELAY
7035  C3 0A 70             JMP       MAINLOP
7038                                               ;
7038                                               ; SUBROUTINE SECTION **********************
7038                                               ;
7038  2F         SIGNPOS   CMA                     ;SINCE POSITIVE VOLTAGE COMPLIMENT IT
7039  3C                   INR A
703A  47                   MOV B,A
703B  3E 2B                MVI A,    2BH           ;OUTPUT PLUS
703D  C3 1B 70             JMP       OUTVAL
7040                                               ;
7040                                               ;** MULTIPLICATION SUBROUTINE L TIMES C IS
                                                   STORED IN H,L
```

```
7040                                          MULTIPLY
7040  F5                  PUSH PSW
7041  C5                  PUSH B
7042  D5                  PUSH D
7043  55                  MOV D,L             ;D HOLDS ORIGINAL MULTIPLICAN
7044  2E 00               MVI L,   0          ;H & L ARE CLEARED
7046  26 00               MVI H,   0
7048  06 00               MVI B,   0          ;B IS CLEARED BECAUSE C IS ROTATED IN TO IT
704A  1E 09               MVI E,   9          ;E IS USED AS LOOP COUNTER
704C  1D         MLOP1    DCR E               ;BEGINNING OF MAIN MULTIPLYING LOOP
704D  CA 67 70   JZ       DONE                ;DO UNTIL E IS ZERO (8 TIMES)
7050  7A                  MOV A,D
7051  1F                  RAR                 ;ROTATE D RIGHT VIA ACCUMULATOR
7052  57                  MOV D,A
7053  D2 5C 70            JNC      SKIP1      ;IF CARRY = 1 THEN ADD ELSE SKIP
7056  79                  MOV A,C
7057  85                  ADD L               ;B,C + H,L -> H,L (16 BIT ADDITION)
7058  6F                  MOV L,A             ;VIA ACCUMULATOR
7059  78                  MOV A,B
705A  8C                  ADC H
705B  67                  MOV H,A
705C  79         SKIP1    MOV A,C
705D  37                  STC                 ;SETS CARRY TO ZERO BEFORE ROTATING LEFT
705E  3F                  CMC
705F  17                  RAL
7060  4F                  MOV C,A             ;ROTATE RIGHT B AND C VIA ACCUMULATOR
7061  78                  MOV A,B             ;FOR NEXT PLACE ADDITION
7062  17                  RAL
7063  47                  MOV B,A
7064  C3 4C 70            JMP      MLOP1      ;END OF MAIN LOOP
7067  D1         DONE     POP D
7068  C1                  POP B
7069  F1                  POP PSW
706A  C9                  RET
706B                                          ;END OF MULTIPLY SUBROUTINE
706B                                          ;
706B                                          ;ROTATED H,L C TIMES TO RIGHT.  ROTATE IN
                                              ZEROS
706B  F5         ROTR     PUSH PSW
706C  C5                  PUSH B
706D  7C         RLOP1    MOV A,H
706E  37                  STC                 ;SETS CARRY FLAG
706F  3F                  CMC                 ;COMPLIMENTS CARRY FLAG TO MAKE IT ZERO
7070  1F                  RAR                 ;ROTATES H TO THE RIGHT W/CARRY VIA
                                              ACCUMULATOR
7071  67                  MOV H,A
7072  7D                  MOV A,L
7073  1F                  RAR                 ;ROTATES L TO RIGHT VIA ACCUMULATOR
7074  6F                  MOV L,A
7075  0D                  DCR C               ;CHECKS TO SEE IF DONE ROTATING
7076  C2 6D 70            JNZ      RLOP1
7079  C1                  POP B
707A  F1                  POP PSW
707B  C9                  RET
707C                                          ;END OF ROTATE SUBROUTINE
707C                                          ;
```

```
707C   45 54 57 2D 33 STRING1 DB             'ETW-3800 Voltmeter'
       38 30 30 20 56
       6F 6C 74 6D 65
       74 65 72
 708E  0D                    DB       0DH
 708F  5F 5F 5F 5F 5F        DB                     '_____ tenthsVolts'
       20 74 65 6E 74
       68 73 56 6F 6C
       74 73
 70A0  08 08 08 08           DB       08H,08H,08H,08H
 70A4  08 08 08 08           DB       08H,08H,08H,08H
 70A8  08 08 08 08           DB       08H,08H,08H,08H
 70AC  00                    DB       00H
 70AD  08 08 08 08  STRING2  DB       08H,08H,08H,08H
 70B1  08 08 00              DB       08H,08H,00H
 70B4
 70B4
 70B4                        END      SSTART

SYMBOL TABLE:

DELOOP      702F H      DONE        7067 H      MAINLOP     700A H
MLOP1       704C H      MULTIPLY    7040 H      OUTVAL      701B H
RLOP1       706E H      ROTR        706B H      SIGNPOS     7038 H
SKIP1       705C H      STRING1     707C H      STRING2     70AD H
```

MEMORY ORGANIZATION

A memory map of the CPU Module is located in the Appendix. The major sections shown in the memory map are:

User code area — Available for user purposes.

User data and stack area — Used to hold the stack of the user program and Monitor. The top of the user stack is initialized to the top of this area and builds downward. For the majority of its operations, the Monitor uses its own stack. However, for some operations, the Monitor will also use part of the user stack. For this reason, you should allocate more than enough memory for your applications (over 2000 bytes). This will enable you to make use of calls/returns and pushes/pops without being concerned about stack space.

Monitor data and stack area — Used by the Monitor to hold important system variables. Altering these variables may cause unpredictable results.

Memory module area — When the cartridge is installed, the addresses C000H to FFFFH are available for storing user programs.

I/O area — This is the location of the I/O memory locations for all the I/O peripheral devices.

Keyboard area — In this I/O memory area are three addresses corresponding to separate columns of the keyboard, one address per column.

Output port — I/O memory locations A0H to A3H are used to latch values in the output port.

Input port — Locations 90H to 93H are used to read values from the input port.

LCD registers — Used to hold the liquid crystal display commands and addresses of display registers.

General I/O areas — Available for user purposes.

Unused — Available for user purposes.

Monitor ROM area — Contains the Monitor routine and several general purpose routines. The ROM is addressed beginning at 0000H and ends at 5FFFH.

INTERRUPTS

In order to maintain full control of your CPU, you must have a way to halt the execution of a program to service an internal CPU function or allow an external peripheral access to the CPU. Generally speaking, an interrupt is a temporary break in the normal execution of a program. When the CPU encounters an interrupt, it jumps to the area in memory that holds the subroutine to service the interrupt.

The starting addresses for all interrupt routines are stored in the CPU Module's Monitor ROM at addresses 0000H to 003FH. This dedicated area of memory is called the Interrupt Jump Table (see Appendix). This table jumps to an interrupt jump table stored in RAM starting at 6800H. By changing the address of the interrupt table in RAM, you can program your own interrupt service routines. Those interrupts not used by the Monitor jump to the Monitor. The Reset Jump Vector in ROM cannot be reprogrammed.

To use your interrupt service routine instead of the Monitor's, place the starting address of your service routine at the appropriate address, as specified in the table located in the Appendix.

BREAKPOINTS

Breakpoints (also known as a breakpoint instructions) are interrupts that you place in a program when you want to execute part of a program and then stop. Breakpoints are usually inserted into programs during the debugging process as a way of displaying registers, memory locations, etc., at critical points in a program. To set a breakpoint, use the breakpoint key (key number 7). Up to 16 different breakpoints can be set at one time.

Breakpoints must be placed at instruction addresses. If you set a breakpoint at an improper location, it could inadvertently change or even "crash" your program. Properly inserted, your program will run until it encounters the breakpoint. When the program stops, the address of the break instruction (contained in the program counter) and the instruction will be displayed. You may examine and make changes to any register or memory location. The instruction displayed when the program stopped will be the next one executed when you press the SS or the GO key.

JUMP TABLE

At the beginning of the CPU Module's EPROM (address 0040H) is a jump table containing the addresses of each subroutine. For jump instructions to some of the system-level subroutines, see the jump table located in the Appendix.

When in the Single Step mode, the Monitor ROM will set the next step at the instruction after your call, since the Monitor cannot modify the code in the ROM to add a software interrupt.

FURTHER INFORMATION

The preceding sections are a very brief overview of the CPU's instruction set and its use. For more information about microprocessors and programming, refer to the appropriate Heathkit/Zenith Educational Systems courses and their related experiments using the versatile ETW-3800 Microprocessor Trainer.

ALTERNATE LINE VOLTAGE WIRING

Your Microprocessor Trainer has been factory-wired for 120 VAC line voltage, the most often used voltage in the United States. However, in other countries, 220/240 VAC is the most common line voltage. To change the operating line voltage for your Trainer, complete the following steps. CAUTION: Completing these steps allows operation of your Trainer ONLY ON 220/240 VAC.

IMPORTANT: The plug on the line cord furnished with your Microprocessor Trainer does NOT meet the U.S. National Electrical Code requirements for use on line voltages above 120 VAC. Therefore, you MUST cut the plug from the line cord and install an appropriate plug that matches your 220/240 VAC outlet and meets the electrical code requirements. In addition, you must perform the following steps:

Refer to Pictorial 4 for the following steps.

NOTE: These steps should be completed by qualified service personnel only.

( ) Disconnect the Trainer from the AC outlet, if this has not already been done.

( ) Remove the eight #6 × 3/8" self-tapping screws that secure the cabinet top to the cabinet bottom, and lift the top off.

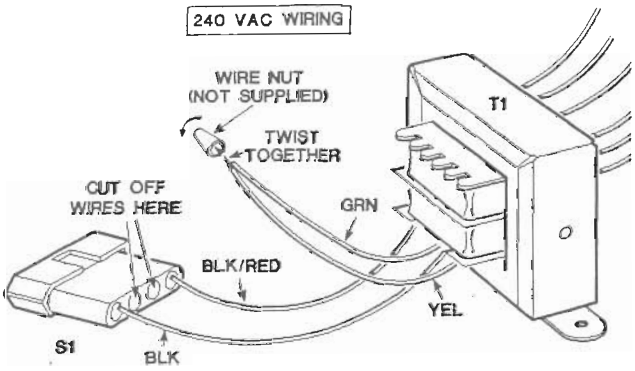
( ) Cut the yellow and green power transformer leads from pins 2 and 3 of the 4-pin socket. Cut the leads as close to the socket as possible. Then remove 3/8" of insulation from the end of each wire.

( ) Twist the ends of the yellow and green power transformer leads together. Then twist a wire nut (not supplied) clockwise over the wires as shown. A wire nut can be obtained locally.

( ) Replace the 1/4-ampere line fuse with a 1/8-ampere, slow-blow fuse (not supplied).

( ) Reposition the cabinet top over the cabinet bottom. Then reinstall the eight #6 × 3/8" self-tapping screws you removed earlier.

Your ETW-3800 Microprocessor Trainer is now wired for operation on 220/240 VAC only.



PICTORIAL 4

IN CASE OF DIFFICULTY

This section of the manual will help you locate and correct minor difficulties which may occur in your Trainer. Any difficulty you might experience can probably be traced to improper setup or interconnections. Use the following Troubleshooting Chart to help narrow down the cause of a problem.

If you cannot resolve the problem refer to the "Customer Service" information inside the rear cover of your Manual. Your warranty is also located inside the rear cover.

TROUBLESHOOTING CHART

DIFFICULTY	POSSIBLE CAUSE
Power LED does not light.	1. Power cord not plugged into AC outlet. 2. Power switch not turned on. 3. Fuse blown or missing.
No display.	1. CPU Module not installed or not properly inserted. (See Appendix for correct insertion procedure.)
Load errors.	1. Memory Module not properly inserted.
Bad checksum.	1. Program in the Memory Module has been modified since it was last saved. Check program for incorrect values.

CIRCUIT DESCRIPTION

As you read this section, refer to the Schematic Diagrams (fold-in).

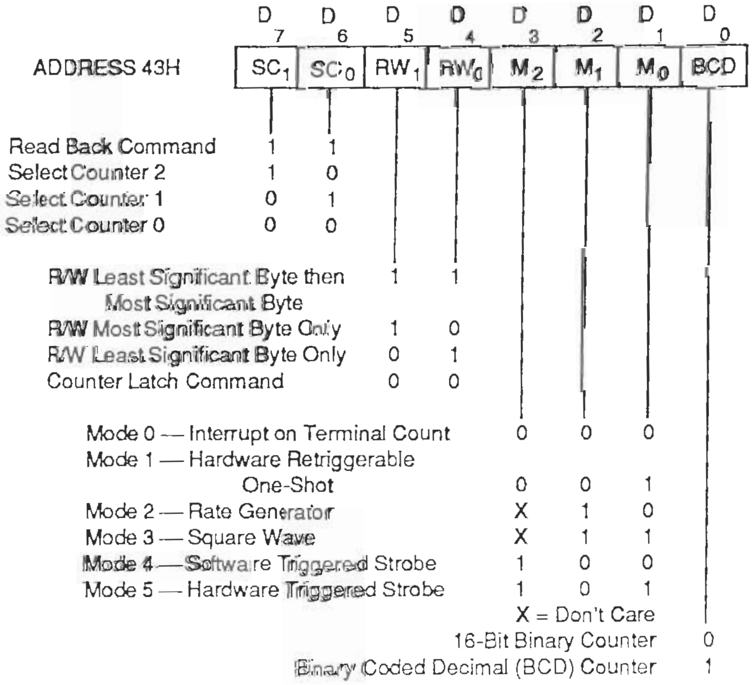
CPU MODULE

The replaceable CPU Module is a computer system within a cartridge and its operation is very complex. It contains not only a CPU (control processing unit) but also memory, timer, I/O interfaces, A/D and D/A converter, interrupt logic and a sophisticated monitor program.

The CPU within the cartridge is a VLSI (Very Large Scale Integrated) 8-bit 80C85 microprocessor (U11) that is object-code compatible with the 8080 family of microprocessors. For detailed information concerning the CPU, refer to INTEL's Embedded Controller Handbook, Vol. 1.

Memory within the cartridge consists of 24K by 8 EPROM and 24K by 8 Static RAM. The EPROM contains all firmware needed to operate the Trainer from both the keyboard and the RS-232 port. The 24K by 8 Static RAM supplies plenty of memory for complex experiments and demonstrations.

A timer system is made up of three independent 16-bit counters (U216). These three counters have six programmable timer modes. The counters and their modes are at address 43H, the Timer Control Register, as shown below.



	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
ADDRESS 43H	SC_1	SC_0	RW_1	RW_0	M_2	M_1	M_0	BCD

	SC_1	SC_0
Read Back Command	1	1
Select Counter 2	1	0
Select Counter 1	0	1
Select Counter 0	0	0

	RW_1	RW_0
R/W Least Significant Byte then Most Significant Byte	1	1
R/W Most Significant Byte Only	1	0
R/W Least Significant Byte Only	0	1
Counter Latch Command	0	0

	M_2	M_1	M_0
Mode 0 — Interrupt on Terminal Count	0	0	0
Mode 1 — Hardware Retriggerable One-Shot	0	0	1
Mode 2 — Rate Generator	X	1	0
Mode 3 — Square Wave	X	1	1
Mode 4 — Software Triggered Strobe	1	0	0
Mode 5 — Hardware Triggered Strobe	1	0	1

X = Don't Care

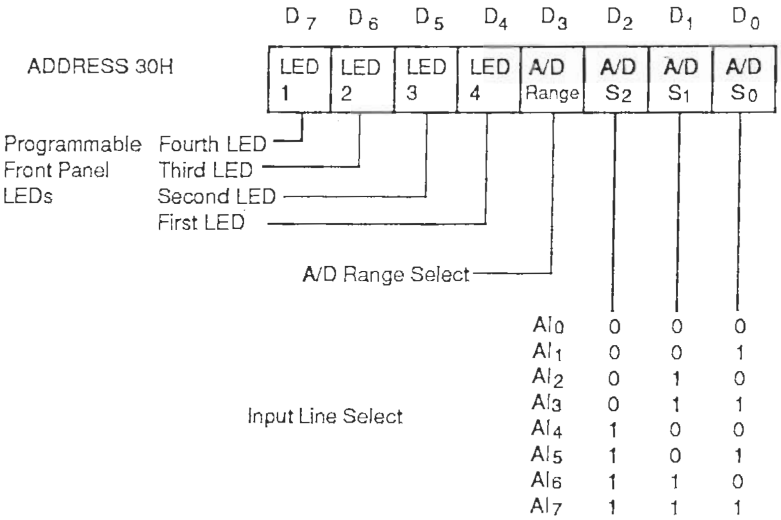
	BCD
16-Bit Binary Counter	0
Binary Coded Decimal (BCD) Counter	1

Diagram 1

The third timer, counter 2, is used to generate the timing for RS-232 baud generation. Its output is directly connected to the R7.5 line of the CPU. Counter 2 can be reprogrammed for your own interrupt routines, however, the Module will lose RS-232 communication capability. For more information on how to use the timer, refer to Heathkit/Zenith Educational Courses or INTEL's Microprocessor and Peripheral Handbook, Vol. II.

The A/D and D/A converter (U25) receives its input from eight multiplexed input channels through the 8-channel analog multiplexer/demultiplexer (U257). Each input channel is selected by programming bits at locations D_2, and D_1, and D_0 of the Programmable Status Register at I/O address 30H as shown below.

Programmable Status Register



	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
ADDRESS 30H	LED 1	LED 2	LED 3	LED 4	A/D Range	A/D S_2	A/D S_1	A/D S_0

	D_2	D_1	D_0
AI_0	0	0	0
AI_1	0	0	1
AI_2	0	1	0
AI_3	0	1	1
AI_4	1	0	0
AI_5	1	0	1
AI_6	1	1	0
AI_7	1	1	1

Diagram 2

Using the fourth bit, the Range Select bit (D_3), a sine wave up to 25 kHz can be digitized with a four-sample resolution. This allows up to 256 discrete levels, in 8-bit two's complement form, of sampling. An operational amplifier (op amp) buffers the input to either a full scale range of ± 1.25 volts or ±2.50 volts depending on the condition of range bit D_3. The following table shows the possible Full Scale Ranges that are available using the appropriate Input Select Line and whether the Range Bit is enabled or disabled. For more information on the A/D-D/A converter (AD7569) used in the CPU module, refer to the Analog Devices' Data Conversion Products Databook, 1988.

INPUT SELECT LINE	FULL SCALE RANGE		INTERNAL SERIES RESISTANCE (R_{si})
	Range Bit (R_b)		
	Disabled	Enabled	
AI_0, AI_3, AI_4, AI_5	±1.25±0.01	±2.5±0.02	100kΩ±1%
AI_1	±0.125±0.001	±0.250±0.002	10kΩ±1%
AI_2	±0.0125±0.0001	±0.025±0.0002	1kΩ±1%
AI_6, AI_7	±7.55±0.006	±15.1±0.12	604kΩ±1%

Other ranges that are not shown can be configured by selecting an external series resistor and using this equation:

$$V_{fs} = \frac{(R_{se} + R_{si})}{100k\Omega} \times \frac{(1.25V \times \overline{R}_b) + (2.5V \times R_b)}{1}$$

R_{se} = External Series Resistance (normally zero unless added)
R_{si} = Internal Series Resistance (fixed according to table)
$\overline{R}_b$ = Not Range Bit (Enabled = 0, Disabled = 1)
R_b = Range Bit (Enabled = 1, Disabled = 0)

Five external hardware interrupts and 7 software interrupts are available, of which 10 are maskable. One of the two non-maskable interrupts is TRAP (NMI key) and it is external; the other is the RESET line. To interrupt a program and save the register values, use the NMI key. To get absolute control of the processor and to reinitialize it, use RESET.

MEMORY MODULE

The optional Memory Module plug-in cartridge contains two 8K × 8 EEPROMs. $\overline{\text{WRITE}}$ and $\overline{\text{READ}}$ signals from the CPU Module determine the direction of data flow into or out of memory.

LOGIC PROBE

The Logic Probe is accessible through the lower signal connection block. Logic pulses are fed to a high-speed comparator (U31) which accepts only the correct logic ($\leq$0.8V = 0, $\geq$2.0V = 1) levels. The output of U31 is then applied to multivibrator U35 which captures high-speed pulses and generates a 0.1 second pulse. The outputs of U35 and U31 are combined at U36 so that if either a DC state or a high-speed pulse occurs, the appropriate LED will light and a high or low audible tone will be generated. The audible tone can be disabled by grounding the LPAUDC terminal point located at the lower signal connector block. The ground resets U32, which disables the audible tone.

RS-232 INTERFACE

The RS-232 port is a serial I/O port. All RS-232 lines pass through the driver/receiver U33 to the CPU Module cartridge socket. From there the transmit/receive lines go to the serial communications interface buffer (U113) located inside the CPU Module. These two lines are then routed to the CPU. The CPU does not have its own serial port interface circuitry. Therefore, all baud and bit generation is done in software by the Monitor.

I/O PORTS

The 8-bit input and output ports are connected to the CPU Module through latches U22 and U21. Output data is latched by an OPPRT (output port) signal from the I/O decoding section of the CPU Module at address A0H. Input data is sampled when the INPRT (input port) line goes high. This occurs with a CPU read from location 90H. Input data may also be latched with a falling edge signal from the IPL input on the lower signal connector block.

LOGIC SWITCHES

One side of the 8-section logic switch is connected to ground. The other side of each section is connected through a 1000 ohm pull-up resistor in resistor pack R215 to the +5 volts DC power supply. The connectors below the switches provide convenient connection for two wires to each section. With a switch in the lower (closed) position, that terminal provides a logic 0 level (ground); in the up (open) position the level will be a logic 1.

LOGIC LEDS

Eight LEDs are accessible through an 8-section connector block located below the LEDs. Each section can accept two wires, which connect to an LED driver. A 10K pull-down resistor at each terminal holds the input to a logic 0 when no connection is made at the connector block. The driver output passes through an LED and a current-limiting resistor. With a logic 0 input the LED is off. When the input rises to a logic 1 the LED lights.

PROGRAMMABLE STATUS LEDS

With no input to the Status LEDs, they are off. When a logic low is applied, the LED lights and a path is completed through a 150 ohm current-limiting resistor to the +5 VDC power supply. Since these are high-impedance inputs, the LEDs will not load down the connected circuit.

KEYBOARD

Address lines A_3, A_4, and A_5 from the CPU connect to keyboard decoder U29. When U29 is enabled, a logic 0 is applied to one of the keyboard key columns and a logic 1 to the other columns. If a key is closed in the column with a logic 0 on it, a logic 0 is placed on the data line for that row of keys. The CPU determines which key is closed by knowing the address that is on the line and which data line is 0. The diodes in series with the three address lines serve as buffers to prevent two adjacent keys from shorting the column select lines together.

+5 VOLT SUPPLIES

Diodes D101 and D102 rectify the voltage from one secondary winding of transformer T1. Capacitors C101 and C102 filter the resulting voltage before it is applied to regulators U101 and U105. Diodes D109 and D110 provide reverse bias protection.

+12 VOLT SUPPLY

Diodes D104 and D106 rectify the voltage from the remaining secondary winding of transformer T1. Capacitors C104 and C105 filter the resulting voltage before it is applied to current regulator U103. Diode D107 provides transient protection for the regulator. Capacitor C107 further filters the output voltage. Diode D111 provides reverse bias protection.

–12 VOLT SUPPLY

The negative 12-volt supply consists of D103, D105, C108, C109, U104, D108, D112, and C112. This circuit operates similarly to the positive 12-volt supply, but produces a regulated –12 volts.

REPLACEMENT PARTS LIST

Component values and circuit component numbers are those referred to on the Schematics and on the Circuit Board X-Ray Views. If a circuit component number is not listed in the Parts List, that component is not used in the circuit.

To order a replacement part, always include the PART NUMBER. Use the Parts Order Form furnished with this unit. If a Parts Order Form is not available, refer to "Replacement Parts" inside the rear cover of this Manual.

A replacement part may look slightly different than the original part, or may have different printing on it. In any case, the performance of the replacement part will meet or exceed the requirements of the original part. For example: A 15-volt capacitor (10 µF, 15 V) may be replaced with a 25-volt capacitor (10 µF, 25 V).

CABINET

ELECTRONIC PARTS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
C52, C53	21-71	.001 µF (1000 pF) ceramic capacitor (may be marked 102)
C51	27-127	.047 µF (474) Mylar capacitor
L51	45-615	RF choke
T1	54-1054	Power transformer
SW1	61-58	Power switch
D1	412-634	Red LED (light emitting diode)
F1	421-33	Fuse, 1/4-ampere slow-blow

CONNECTORS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
	432-1279	Flat connector lug
	230-6329	2-pin plug with wires
	230-6330	2-pin socket with wires
	230-6344	Green wire with connectors

CABLES-LINE CORD

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
	230-6334	Line cord
	134-1693	LED (2-wire) cable
	134-1692	8-wire power cable
	134-2046	8-wire RS-232 cable

PLASTIC AND METAL PARTS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
	92-929	Cabinet bottom
	92-930	Cabinet top
	94-691	Cabinet insert
	204-3001	AC chassis
	204-3066	AC shield
	205-2005	Backpack mounting plate

HARDWARE

NOTE: Metric equivalent hardware may be used. Do not mix standard and metric hardware.

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
	810-13	3 × 8mm screw
	810-17	3.5 × 10mm flat head screw
	250-1412	4-40 × 3/8" black screw
	254-9	#4 lockwasher
	252-2	#4 nut
	255-757	4-40 spacer
	250-1434	6-BT × 3/8" self-tapping screw
	250-1436	8-32 × 3/8" screw
	254-2	#8 lockwasher
	252-4	#8 nut

MISCELLANEOUS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
	230-6353	Foam cushion
	75-736	Strain relief
	230-6355	Paper insulator
	230-6356	Filter circuit board (may be marked 85-2789-1)
	261-49	Foot
	230-6358	Fuseholder
	230-6345	Small sleeving
	230-6346	Large sleeving
	331-7	Solder
	354-5	Cable tie
	75-918	Top insert insulator
		Blue and white label
		Caution label
		Caution marking label
		Warning label

POWER SUPPLY CIRCUIT BOARD

RESISTORS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
R101	6-271-12	270 Ω, 1/4-watt, 5% (red-vio-brn)

CAPACITORS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
C101	25-903	6800 µF electrolytic
C102-C103	27-145	.22 µF (224) Mylar
C104	25-875	1000 µF electrolytic
C105	21-786	.1 µF (104) axial-lead ceramic
C107	25-885	100 µF electrolytic
C108	25-875	1000 µF electrolytic
C109	21-786	.1 µF (104) axial-lead ceramic
C112	25-885	100 µF electrolytic
C113	27-145	.22 µF (224) Mylar

DIODES-INTEGRATED CIRCUITS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
D101-D106	57-42	1N5401 or 3A1 diode
D107-D112	57-65	1N4002 diode

NOTE: Integrated circuits may be marked for identification in any of the following four ways.

1. Part number.
2. Type number. (For integrated circuits, this refers only to the numbers and letters shown in BOLD print. Disregard any other numbers or letters shown on the IC.)
3. Part number and type number.
4. Part number with a type number other than the one shown.

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
U101	442-30	LM309K
U103	442-664	79M12
U104	442-663	78M12
U105	442-30	LM309K

CONNECTORS-SOCKET

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
P101	432-876	8-pin plug
P102	432-943	2-pin plug
P103	432-877	10-pin plug
	432-1279	Flat connector lug
	434-189	Transistor socket

HARDWARE

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
	250-1425	6-32 × 1/2" screw

MISCELLANEOUS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
	230-6271	Power supply circuit board (may be marked 85-3191-1)
	215-698	Flat heat sink
	352-31	Thermal compound

MAIN CIRCUIT BOARD

RESISTORS

NOTE: The following resistors have a tolerance of 5% unless otherwise listed. A 5% tolerance is indicated by a gold fourth color band.

1/4-Watt Resistors

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
R11-R19	6-271-12	270 Ω (red-viol-brn)
R34-R35	6-271-12	270 Ω (red-viol-brn)
R36	6-4322-12	43.2 kΩ (red-blk-blk-red)
R37	6-2002-12	20 kΩ (yel-org-red-red)
R110-R156	6-271-12	270 Ω (red-viol-brn)
R157-R164	6-101-12	100 Ω (brn-blk-brn)
R165-R172	6-271-12	270 Ω (red-viol-brn)
R177-R178	6-271-12	270 Ω (red-viol-brn)
R181-R188	6-151-12	150 Ω (brn-grn-brn)
R189-R196	6-271-12	270 Ω (red-viol-brn)
R216-R223	6-271-12	270 Ω (red-viol-brn)
R311	6-271-12	270 Ω (red-viol-brn)
R330	6-471-12	470 Ω (yel-viol-brn)
R435-R466	6-271-12	270 Ω (red-viol-brn)

1/8-Watt Resistors

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
R23-R29	6-181-11	180 Ω (brn-gry-brn)
R31	6-5230-11	523 Ω, 1% (grn-red-org-blk)
R32	6-1000-11	100 Ω, 1% (brn-blk-blk-blk)
R33	6-103-11	10 kΩ (brn-blk-org)
R38-R39	6-181-11	180 Ω (brn-gry-brn)
R41-R49	6-101-11	100 Ω (brn-blk-brn)
R210	6-181-11	180 Ω (brn-gry-brn)
R224-R231	6-122-11	1200 Ω (brn-red-red)
R240-R247	6-122-11	1200 Ω (brn-red-red)
R310	6-181-11	180 Ω (brn-gry-brn)
R312-R313	6-104-11	100 kΩ (brn-blk-yel)
R314	6-130-11	10 kΩ (brn-blk-org)
R315	6-8250-11	825 Ω, 1% (gry-red-grn-blk)
R316	6-130-11	10 kΩ (brn-blk-org)
R318	6-5490-11	549 Ω, 1% (grn-yel-wht-blk)
R319	6-130-11	10 kΩ (brn-blk-org)
R320-R322	6-181-11	180 Ω (brn-gry-brn)
R323	6-222-11	2200 Ω (red-red-red)
R324	6-223-11	22 kΩ (red-red-org)
R325	6-273-11	27 kΩ (red-viol-org)
R326	6-272-11	2700 Ω (red-viol-red)
R327	6-130-11	10 kΩ (brn-blk-org)
R328	6-130-11	10 kΩ (brn-blk-org)
R410-R432	6-101-11	100 Ω (brn-blk-brn)

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION

Other Resistors

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
R21	10-1137	2 kΩ potentiometer
RN173-RN176	9-128	10 kΩ resistor pack
RN179-RN180	9-128	10 kΩ resistor pack
RN211-RN213	9-128	10 kΩ resistor pack
RN215	9-118	1 kΩ resistor pack

CAPACITORS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
C11-C14	25-866	22 μF electrolytic
C15	25-978	.1 F
C19	25-927	22 μF electrolytic
C21-C23	21-786	.1 μF ceramic
C24	21-761	.01 μF (103) glass
C25-C28	21-786	.1 μF ceramic
C31	25-863	4.7 μF electrolytic
C32	25-863	4.7 μF electrolytic
C33	25-866	22 μF electrolytic
C34	21-786	.1 μF ceramic
C35-C36	25-866	22 μF electrolytic
C37	25-927	22 μF electrolytic
C38	21-786	.1 μF ceramic
C39	27-161	.01 μF Mylar
C310	21-786	.1 μF ceramic
C311	21-761	.01 μF (103) glass
C312	21-761	.01 μF (103) glass
C313	27-161	.01 μF Mylar
C314	21-811	.33 μF (334) axial-lead ceramic
C315	21-786	.1 μF ceramic
C316	21-761	.01 μF

DIODES-TRANSISTORS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
D11	57-607	1N5817
D29	56-655	1N6263
D37, D39	57-607	1N5817
D210-D211	56-655	1N6263
D310-D313	57-607	1N5817
J31	417-902	NPD 5566

INTEGRATED CIRCUITS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
U21	443-1673	74ACT374
U22	443-1590	74ACT373
U24	443-1591	74ACT04
U25	443-1443	74ACT244
U27	443-1443	74ACT244
U31	442-820	NE521
U32	442-665	79L05
U33	443-1467	MC145406
U35	443-1592	74HCT423
U36	443-1593	74HCT32
U326	442-740	LM556

CONNECTORS-SOCKETS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
P11	432-1816	60-pin male connector
P12	432-1727	40-pin female memory cartridge connector
P13-P14	432-1720	100-pin signal connector block
P15	432-1064	10-pin male connector
P21-P22	432-1719	16-pin connector block
P25	432-1811	14-pin LCD module connector
P32	432-1268	8-pin male RS-232 cable connector
P212-P215	432-1656	62-pin female CPU cartridge connector
	432-1610	Breadboard block

MISCELLANEOUS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
D21-D28	412-657	Green LED
D31-D34	412-657	Green LED
D35	412-634	Red LED
D36	412-657	Green LED
L31-L32	475-39	Inductor
S101-S121	64-955	Pushbutton switch
S225	60-656	8-section DIP switch
T101	473-29	Transducer
V301	411-902	LCD display module
	85-3399-1	Circuit board
	462-1238	1 set of keycaps

CPU MODULE[1]

RESISTORS

NOTE: All resistors are rated at 1/8-watt and have a tolerance of 5% unless otherwise stated.

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
R11	627-223-18	22 kΩ (red-red-org)
R12	627-104-18	100 kΩ (brn-blk-yel)
R13	627-223-18	22 kΩ (red-red-org)
R14-R16	627-223-18	22 kΩ (red-red-org)
R18, R19	627-223-18	22 kΩ (red-red-org)
R22	627-1002-18	10 kΩ, 1% (brn-blk-blk-red)
R25	627-1001-18	1 kΩ, 1% (brn-blk-blk-blk)
R26-R28	627-1003-18	100 kΩ, 1% (brn-blk-blk-org)
R29	627-6043-18	604 kΩ, 1% (blu-blk-yel-org)
R31	627-1003-18	100 kΩ, 1% (brn-blk-blk-org)
R34	627-513-18	51 kΩ (grn-brn-blk)
R110-R112	627-223-18	22 kΩ (red-red-org)
R142, R143	627-103-18	10 kΩ (brn-blk-org)
R210	627-6043-18	604 kΩ, 1% (blu-blk-yel-org)
R211	627-1003-18	100 kΩ, 1% (brn-blk-blk-org)

OTHER RESISTORS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
RP1, RP2	9-143	22 kΩ resistor pack
RP4, RP5	9-169	22 kΩ resistor pack

CAPACITORS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
C11	631-26-39	22 pF
C12	632-6	1.0 µF
C13-C19	21-786	0.1 µF
C22	631-26-55	100 pF
C23	631-27-32	0.1 µF
C24	25-987	10 µF
C25, C26	631-26-79	1000 pF
C27	21-786	0.1 µF
C29	21-786	0.1 µF
C31	21-786	0.1 µF
C35	631-27-32	0.1 µF
C38	631-27-32	0.1 µF
C110-C115	21-786	0.1 µF
C116, C117	25-987	10 µF
C118	21-786	0.1 µF
C210-C213	21-786	0.1 µF

DIODES

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
D11, D12	57-607	1N5817

INTEGRATED CIRCUITS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
U11	443-1678	80C85A
U12	643-281	74HC374
U13	444-822	20V8
U14	643-178 (or 443-1500)[2]	62256
U15	643-278	74ACT241
U16-U18	643-109	74ACT245
U19	643-102	74ACT240
U21	444-823	20V8
U22	643-183	74ACT138
U23	643-279	74HC273
U24	643-117	74ACT244
U25	643-282	AD7569
U26	442-836 (or 642-12)[2]	79L05
U32	442-627 (or 642-3)[2]	78L05
U110	643-286	74HC132
U112	442-835	8054
U113	643-117	74ACT244
U114	444-821	27C256
U118	643-280	74HCT74
U216	643-283	82C54
U257	642-13	74HC4051
U341	642-7	LF353

[1] Parts are for reference only. CPU Module to be serviced by Heath Company only.

[2] Surface-mount component.

MISCELLANEOUS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
Y11	404-724	3.6864 MHz crystal
	85-3575-1	Module circuit board
	92-919-1	Case back
	92-931	Case front
	250-1322	#6BT × 5/8" self-threading screw
	250-1630	#6BT × 1.5" self-threading screw
	260-735	Ground clip
	266-1329	Static protection clip
	390-3259	Trainer upper connector block overlay
	390-3273	Trainer lower connector block overlay
	390-3298	Module label
	434-312	28-pin IC socket
	434-368	24-pin IC socket

MEMORY MODULE

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
RESISTORS		
R11-R12	6-472-12	4.7 kΩ, 1/4-watt, 5%
CAPACITORS		
C301-C302	21-786	.1 µF
DIODES		
D11-D12	56-655	1N6263

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
INTEGRATED CIRCUITS		
U301-U302	643-179	58C65
MISCELLANEOUS		
	85-3434-1	Circuit board
	92-932	Case front with ground clips
	92-917	Case back
	250-1322	#6 × 5/8" BT self-threading screw
	266-1330	Memory static protection clip
	390-3211	Label
	434-312	IC socket

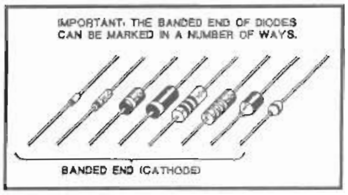
SEMICONDUCTOR IDENTIFICATION

TRAINER

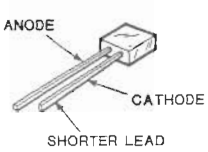
DIODES-LEDS-TRANSISTORS

COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
D101-D106	57-42	1N5401 or 3A1	A1
D107-D112	57-65	1N4002	A1
D29, D210, C211	56-655	1N6263	A1
D11, D37-D39, D310-D311	57-607	1N5817	A1
D21-D28, D31-D34, D36	412-657	Green LED	A2
D1, D35	412-634	Red LED	A2
J31	417-902	NPD 5566	A3

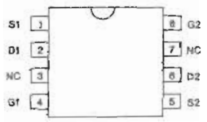
A1


IMPORTANT: THE BANDED END OF DIODES CAN BE MARKED IN A NUMBER OF WAYS.
BANDED END (CATHODE)

A2



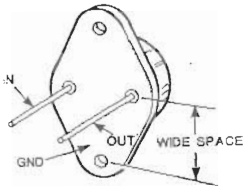
A3



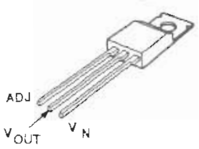
INTEGRATED CIRCUITS

COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
U101, U105	442-30	LM309K	B1
U103	442-708	LM317T	B2
U104	442-709	LM337T	B3

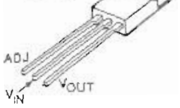
B1



B2

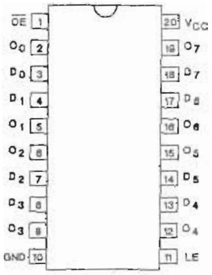


B3

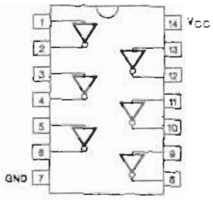


COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
U21	443-1673	74ACT374	B4
U22	443-1590	74ACT373	B4
U24	443-1591	74ACT04	B5
U25, U27	443-1443	74ACT244	B6
U31	442-820	NE521	B7
U32	442-665	79L05	B8
U33	443-1467	MC145406	B9
U35	443-1592	74HCT423	B10
U36	443-1593	74HCT32	B11
U326	442-740	LM556	B12

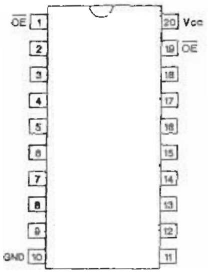
B4



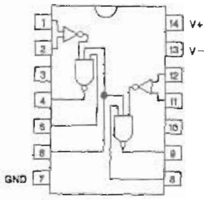
B5



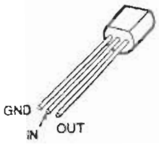
B6



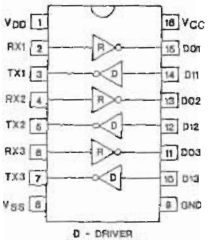
B7



B8

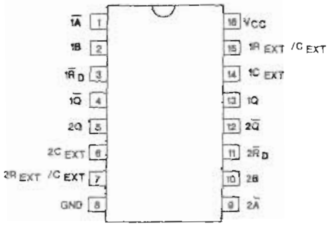


B9

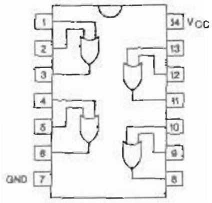


D - DRIVER
R - RECIEVER

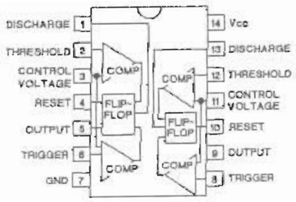
B10



B11



B12

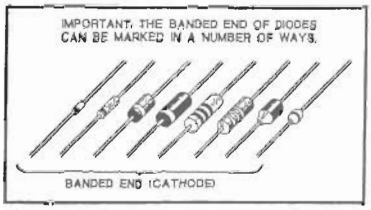


CPU MODULE

DIODES

COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
D11, D12	57-607	1N5817	A1

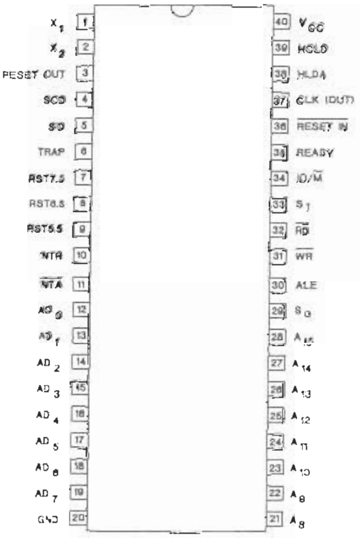
A1


IMPORTANT: THE BANDED END OF DIODES CAN BE MARKED IN A NUMBER OF WAYS.
BANDED END (CATHODE)

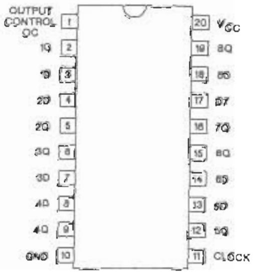
INTEGRATED CIRCUITS

COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
U11	443-1678	80C85A	B1
U12	643-281	74HC374	B2
U13	444-822	20V8	B3
U14	643-178 (or 443-1500)[1]	62256	B4
U15	643-278	74ACT241	B5
U16-U18	643-109	74ACT245	B6
U19	643-102	74ACT240	B7
U21	444-823	20V8	B3

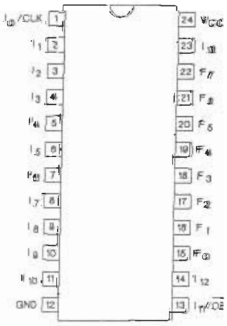
B1



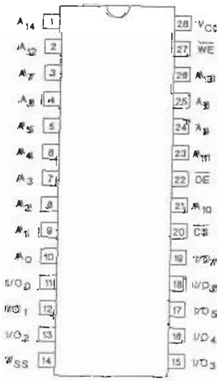
B2



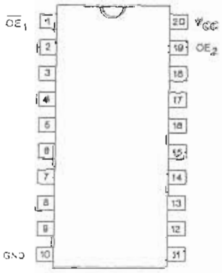
B3



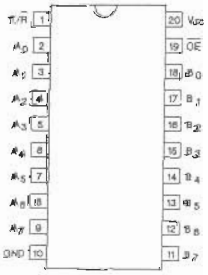
B4



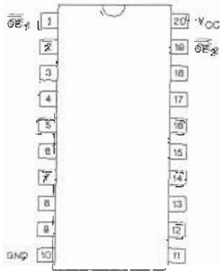
B5



B6



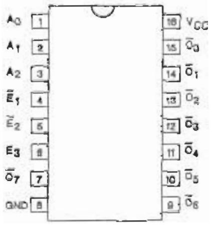
B7



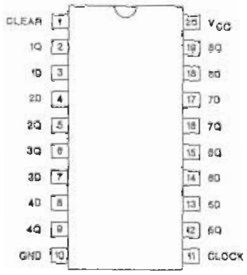
[1] Surface-mount component.

COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
U22	643-183	74ACT138	B8
U23	643-279	74HC273	B9
U24	643-117	74ACT244	B10
U25	643-282	AD7569	B11
U26	442-836 (or 642-12)[1]	79L05	B12
U32	442-627 (or 642-3)[1]	78L05	B13
U110	643-286	74HC132	B14
U112	442-835	8054	B15
U113	643-117	74ACT244	B10

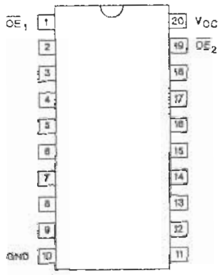
B8



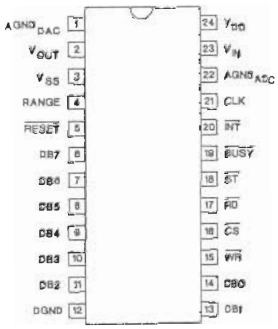
B9



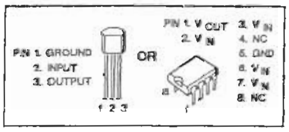
B10



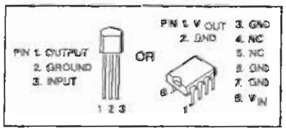
B11



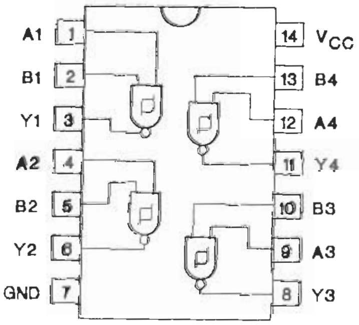
B12



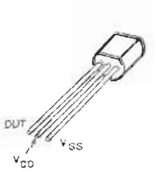
B13



B14



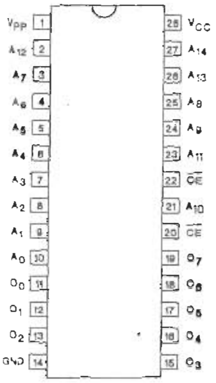
B15



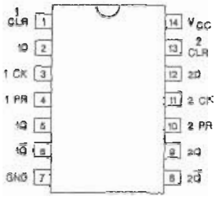
[1] Surface-mount component.

COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
U114	444-821	27C256	B16
U118	643-280	74HCT74	B17
U216	643-283	82C54	B18
U257	642-13	74HC4051	B19
U341	642-7	LF353	B20

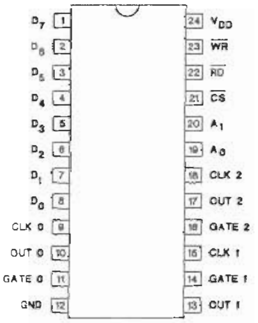
B16



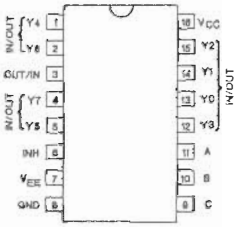
B17



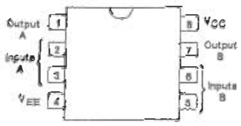
B18



B19



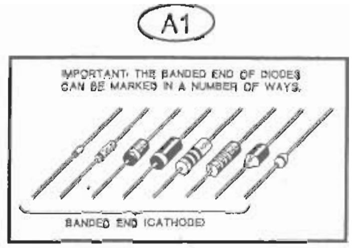
B20



MEMORY MODULE

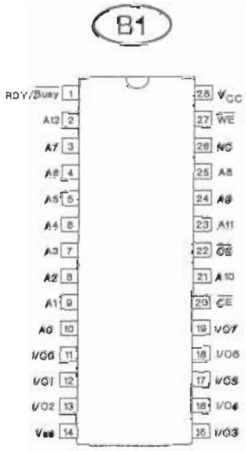
DIODES

COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
D11, D12	56-655	1N6263	A1


A1
IMPORTANT: THE BANDED END OF DIODES CAN BE MARKED IN A NUMBER OF WAYS.
BANDED END (CATHODE)

INTEGRATED CIRCUITS

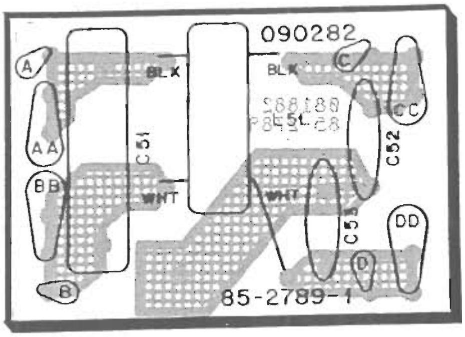
COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
U301, U302	643-179	58C65	B1



CIRCUIT BOARD X-RAY VIEWS

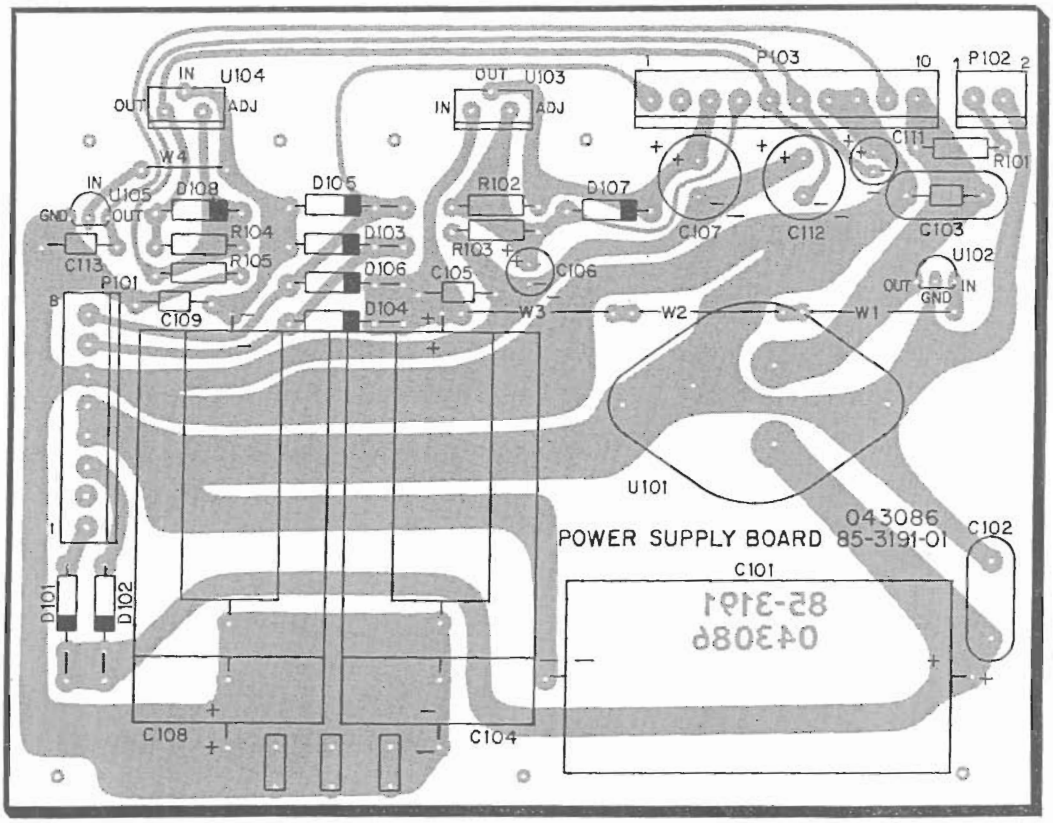
To find the PART NUMBER of a component for the purpose of ordering a replacement part:

1. Find the circuit component part number on the appropriate X-Ray View.
2. Locate the same number in the "Circuit Component Number" column of the corresponding "Parts List".
3. Adjacent to the circuit component number, you will find the PART NUMBER and DESCRIPTION, which you must supply when you order a replacement part.



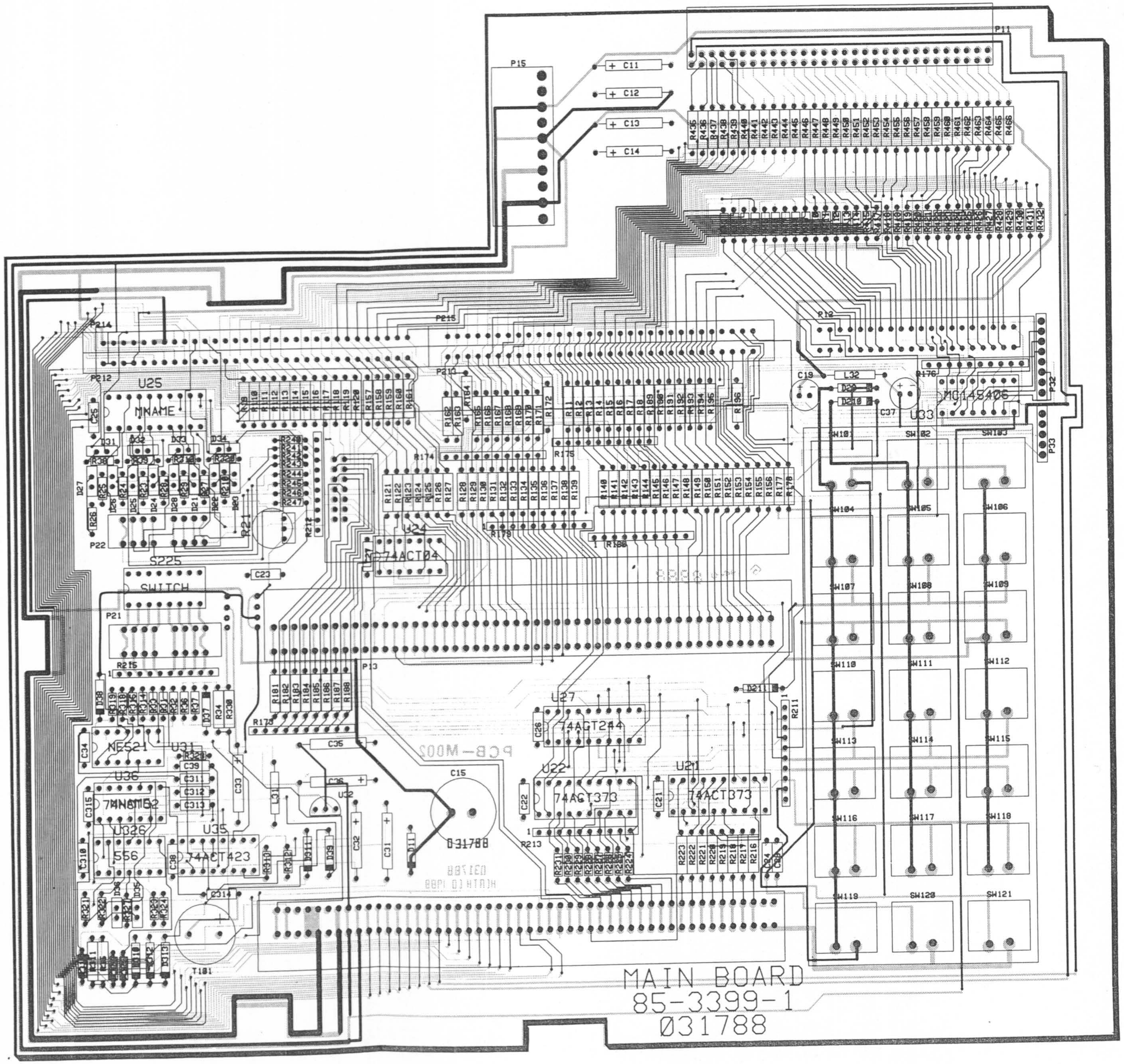
FILTER CIRCUIT BOARD

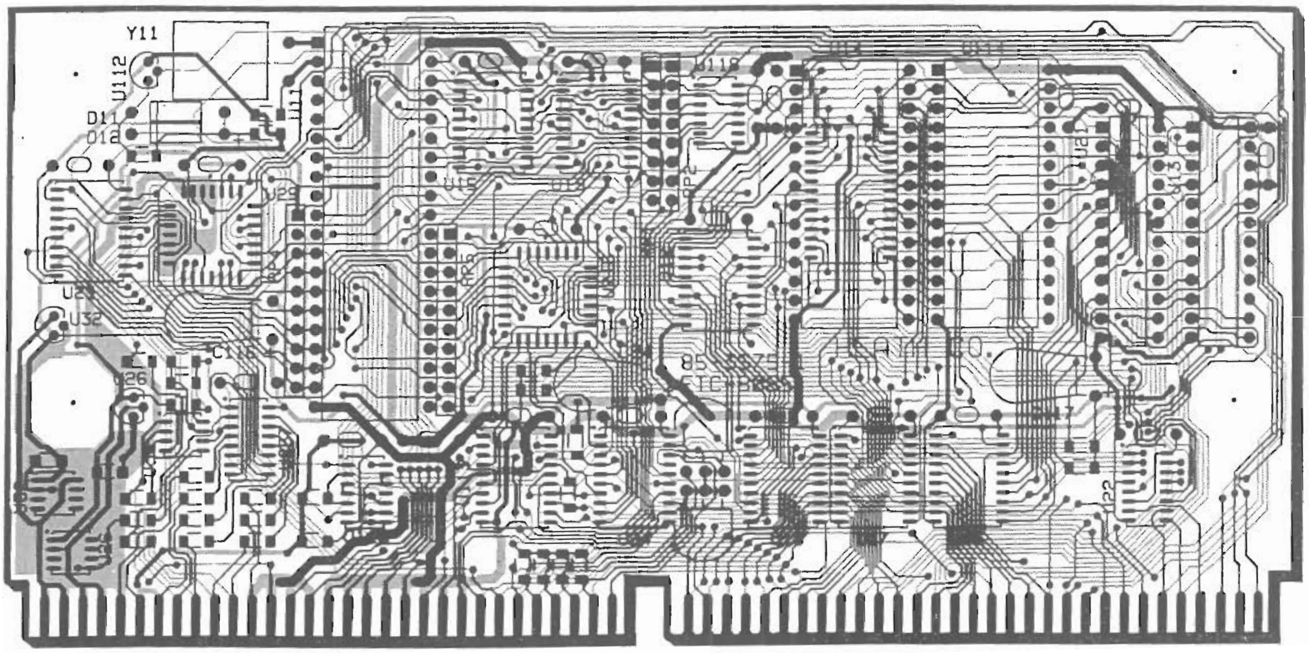
(Shown from the component side)



POWER SUPPLY CIRCUIT BOARD

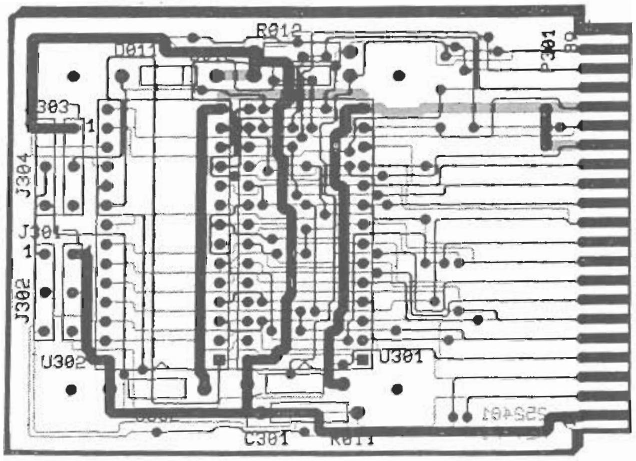
(Shown from the component side)


P15
C11
C12
C13
C14
P11
P214
P215
P212
P213
P12
U25
MNAME
C25
D31
D32
D33
D34
R30
R29
R210
D27
P22
S225
SWITCH
C23
P21
R215
U24
74ACT04
R174
R175
R179
R180
P13
R178
R176
C19
L32
D29
D210
C37
U33
MC145406
P32
P33
SW101
SW102
SW103
SW104
SW105
SW106
SW107
SW108
SW109
SW110
SW111
SW112
SW113
SW114
SW115
SW116
SW117
SW118
SW119
SW120
SW121
D211
R211
U27
C26
74ACT244
U22
C22
74ACT373
U21
C21
74ACT373
R213
C15
031788
PCB-M002
C35
C36
U32
C33
C31
C32
D11
D39
C34
NE521
U31
U36
C315
74HC112
U326
556
C310
U35
C38
74ACT423
C314
T101
R173
D37
D38
R34
R330
MAIN BOARD
85-3399-1
Ø31788



ETC-8085 CPU MODULE

(Shown from the component side)



ETC-128 MEMORY MODULE

(Shown from the component side)

APPENDIX

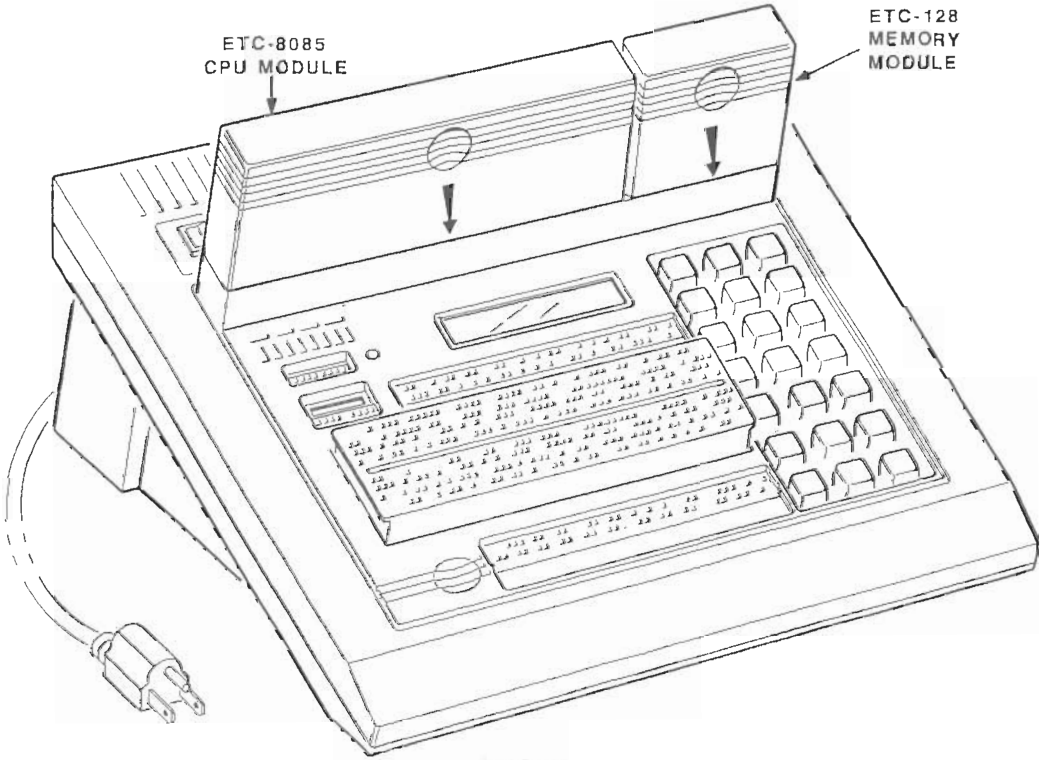
CARTRIDGE INSERTION AND REMOVAL

With proper care and handling, your ETW-3800 Microprocessor Trainer will provide you with years of service. Knowing how to properly install and remove the cartridges used with the Trainer will help you to protect your investment. Refer to Pictorial 5 while following these steps to correctly insert and remove cartridges.

1. Make sure the Trainer power switch is off before removing or inserting a cartridge.

2. Remove the static protection clip from the cartridge. A cartridge should always have a static protection clip on whenever it is not in the Trainer.

3. Insert the cartridge into its appropriate connector with the lettering facing you. The CPU Module installs in the two left connectors and the Memory Module installs into the right connector. Both the CPU and the Memory Module are keyed to go into the connectors of the Trainer one way only. Do not force the cartridges in backwards or they will be damaged.



PICTORIAL 5

MEMORY MAP

ADDRESS	MEMORY
0000H 003FH	Monitor Interrupt Table
0040H 006FH	Monitor Subroutine Jump Table
0070H 5FFFH	Monitor ROM Area
6000H 67FFH	Monitor Data and Stack Area
6800H 6820H	User Programmable Interrupt Vector Jump Table
6821H 6FFFH	User Data and Stack Area
7000H 8FFFH	User Program Area
A000H BFFFH	8K expansion (External) Memory Area
C000H FFFFH	Memory Module Area

I/O MEMORY MAP*

00H 0FH	Keyboard Column 1
20H 1FH	Keyboard Column 2
20H 2FH	Keyboard Column 3
30H 3FH	Local Latch (Bit 7 controls LED #4 Bit 6 controls LED #3 Bit 5 controls LED #2 Bit 4 controls LED #1 Bit 3 controls A/D Range Bit 2 controls A/D Input Line Select Bit 1 controls A/D Input Line Select Bit 0 controls A/D Input Line Select)
40H, 44H, 48H, 4CH	Timer Register 1 (Use 40H to access.)
41H, 45H, 49H, 4DH	Timer Register 2 (Use 41H to access.)
42H, 46H, 4AH, 4EH	Timer Register 3 (Use 42H to access.)
43H, 47H, 4BH, 4FH	Timer Register 4 (Use 43H to access.)
50H 5FH	General I/O 0
60H 6FH	General I/O 1
70H 7FH	General I/O 2
80H 83H	A/D Input D/A Output (Use 80H to access.)
90H 93H	Input Port
A0H A3H	Output Port
B0H, B2H	LCD Data Register (Use B0H to access.)
B1H, B3H	LCD Command Register (Use B1H to Access.)

*These locations are not fully decoded. Use only the address specified.

SUBROUTINE JUMP TABLE*

EPROM ADDRESS	LABEL	INTRODUCTION
0040	JMP MAIN	Return Control to Monitor
0043	JMP GETC	Returns ASCII Character in A
0046	JMP PUTC	Outputs ASCII Character in A
0049	JMP SETRS232	Select RS-232 for I/O
004C	JMP SETLCD	Select LCD, Keypad for I/O
004F	JMP GETHBYT	Get a Hex Byte, Return in A
0052	JMP GETHWRD	Get a Hex Word, Return in HL
0055	JMP PUTHBYT	Print Byte in A, Out in Hex
0058	JMP PUTHWRD	Print Word in HL, Out in Hex
005B	JMP PUTSTRING	Print a Null Terminated String Pointed to by HL
005E	JMP COPYUP	Copy Block of Memory, B=# Copy, Push Source, Push Destination
0061	JMP COPYDOWN	Copy Block of Memory, B=# Copy, Push Source, Push Destination
0064	JMP COPY	Push Incordec, # of Bytes, Destination, D=Source
0067	JMP PAUSE	Wait Over a Second
006A	JMP PUTDSWRD	Convert Signed Value in HL to Decimal and Print Out
006D	JMP PUTUDWRD	Convert Unsigned Value in HL to Decimal and Print Out

* You must use the CALL command (CDH) or a conditional CALL when calling a subroutine from the Jump Table. For example:

STEP	ADDRESS	LABEL	COMMENT
A	NNNX	CALL 0043H	Control transfers to Jump Table location 0043
B			Control transfers to GETC routine
C			GETC
D	NNN(X+3)		Control returns back to the program immediately after the call

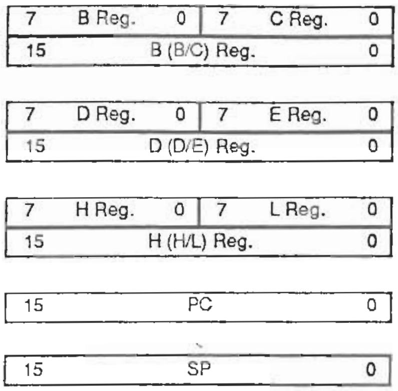
INTERRUPT VECTOR TABLE

ADDRESS	INTERRUPT SOURCE
6801H, 6802H	Reset 1H
6804H, 6805H	Reset 2H
6807H, 6808H	Reset 3H
680AH, 680BH	Reset 4H
680DH, 680EH	TRAP (NMI)
6810H, 6811H	Reset 5H
6813H, 6814H	Reset 5.5H
6816H, 6817H	Reset 6H
6819H, 681AH	Reset 6.5H
681CH, 681DH	Reset 7H
681FH, 6820H	Reset 7.5H

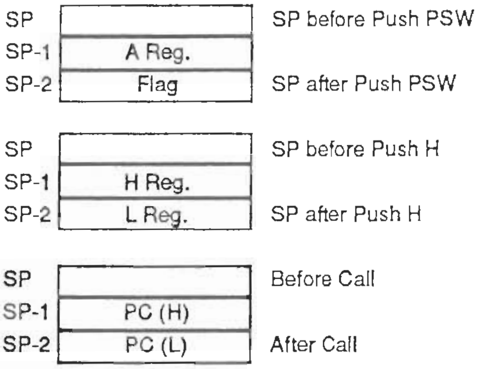
ASCII CHART

ASCII CHARACTER SET (7-BIT CODE)								
MS Dig. LS Dig.	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P		p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	•	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	V	l	|
D	CR	GS	–	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

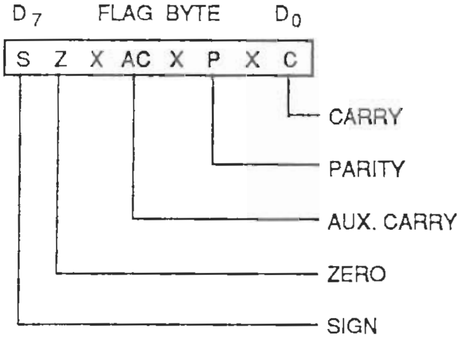
PROGRAMMING MODEL



STACK



NOTE: Push B and Push D function similar to Push H (high order byte then low order byte).



X = Undefined (usually zero).

TABLE OF OPCODES

		LEAST SIGNIFICANT DIGIT															
	HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
MOST SIGNIFICANT DIGIT	0	NOP	LXI B, + +	STAX B	INX B	INR B ●	DCR B ●	MVI B, +	RLC ● ●		DAD B ● ●	LDAX B	DCX B	INR C ●	DCR C ●	MVI C, +	RRC ● ●
	1		LX1 D, + +	STAX D	INX D	INR D ●	DCR D ●	MVI D, +	RAL ● ●		DAD D ● ●	LDAX D	DCX D	INR E ●	DCR E ●	MVI E, +	RAR ● ●
	2	RIM	LXI H, + +	SHLD + +	INX H	INR H ●	DCR H ●	MVI H, +	DAA		DAD H ● ●	LHLD + +	DCX H	INR L ●	DCR L ●	MVI L, +	CMA
	3	SIM	LXI SP, + +	STA + +	INX SP	INR M ●	DCR M ●	MVI M, +	STC ● ●		DAD SP ● ●	LDA + +	DCX SP	INR A ●	DCR A ●	MVI A, +	CMC ● ●
	4	MOV B,B	MOV B,C	MOV B,D	MOV B,E	MOV B,H	MOV B,L	MOV B,M	MOV B,A	MOV C,B	MOV C,C	MOVC,D	MOV C,E	MOV C,H	MOV C,L	MOV C,M	MOV C,A
	5	MOV D,B	MOV D,C	MOV D,D	MOV D,E	MOV D,H	MOV D,L	MOV D,M	MOV D,A	MOV E,B	MOV E,C	MOV E,D	MOV E,E	MOV E,H	MOV E,L	MOV E,M	MOV E,A
	6	MOV H,B	MOV H,C	MOV H,D	MOV H,E	MOV H,H	MOV H,L	MOV H,M	MOV H,A	MOV L,B	MOV L,C	MOV L,D	MOV L,E	MOV L,H	MOV L,L	MOV L,M	MOV L,A
	7	MOV M,B	MOV M,C	MOV M,D	MOV M,E	MOV M,H	MOV M,L	HLT	MOV M,A	MOV A,B	MOV A,C	MOV A,D	MOV A,E	MOV A,H	MOV A,L	MOV A,M	MOV A,A
	8	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A	ADC B	ADC C	ADC D	ADC E	ADC H	ADC L	ADC M	ADC A
	9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB M	SUB A	SBB B	SBB C	SBB D	SBB E	SBB H	SBB L	SBB M	SBB A
	A	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A	XRA B	XRA C	XRA D	XRA E	XRA H	XRA L	XRA M	XRA A
	B	ORA B	ORA C	ORA D	ORA E	ORA H	ORA L	ORA M	ORA A	CMP B	CMP C	CMP D	CMP E	CMP H	CMP L	CMP M	CMP A
	C	RNZ	POP B	JNZ + +	JMP + +	CNZ + +	PUSH B	ACI +	RST 0	RZ	RET	JZ + +		CZ + +	CALL + +	ACI +	RST 1
	D	RNC	POP D	JNC + +	OUT +	CNC + +	PUSH D	SUI +	RST 2	RC		JC + +	IN +	CC + +		SBI +	RST 3
	E	RPO	POP H	JPO + +	XTHL	CPO + +	PUSH H	ANI +	RST 4	RPE	PCHL	JPE + +	XCHG	CPE + +		XRI +	RST 5
	F	RP	POP PSW	JP + +	DI	CP + +	PUSH PSW	ORI +	RST 6	RM	SPHL	JM + +	EI	CM + +		CPI +	RST 7

Unshaded Op Code Indicates no flags affected (each + indicates an extra instruction byte)

Shaded Op Code indicates all flags affected

- ● All Flags Except Carry Affected
- ● ● Only Carry Flag Affected

FLAG REGISTER							
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	Z	0	A	0	P	1	C

FLAGS S=SIGN Z=ZERO A=AUX CARRY P=PARITY C=CARRY

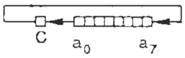
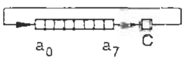
INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES

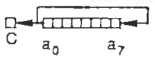
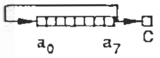
Mnemonic	Description	Boolean Expression	Machine Coding Operand(s)	Bytes	Clock Cycles
ACI	Add immediate to A with carry	Ar + n + C → Ar	CE n	2	7
ADC A	Add register A to A with carry	Ar + Ar + C → Ar	8F	1	4
ADC B	Add register B to A with carry	Ar + Br + C → Ar	88	1	4
ADC C	Add register C to A with carry	Ar + Cr + C → Ar	89	1	4
ADC D	Add register D to A with carry	Ar + Dr + C → Ar	8A	1	4
ADC E	Add register E to A with carry	Ar + Er + C → Ar	8B	1	4
ADC H	Add register H to A with carry	Ar + Hr + C → Ar	8C	1	4
ADC L	Add register L to A with carry	Ar + Lr + C → Ar	8D	1	4
ADC M	Add memory to A with carry	Ar + (HL)$_m$ + C → Ar	8E	1	7
ADD A	Add register A to A	Ar + Ar → Ar	87	1	4
ADD B	Add register B to A	Ar + Br → Ar	80	1	4
ADD C	Add register C to A	Ar + Cr → Ar	81	1	4
ADD D	Add register D to A	Ar + Dr → Ar	82	1	4
ADD E	Add register E to A	Ar + Er → Ar	83	1	4
ADD H	Add register H to A	Ar + Hr → Ar	84	1	4
ADD L	Add register L to A	Ar + Lr → Ar	85	1	4
ADD M	Add memory to A	Ar + (HL)$_m$ → Ar	86	1	7
ADI	Add immediate to A	Ar + n → Ar	C6 n	2	7
ANA A	And register A with A	Ar • Ar → Ar	A7	1	4
ANA B	And register B with A	Ar • Br → Ar	A0	1	4
ANA C	And register C with A	Ar • Cr → Ar	A1	1	4
ANA D	And register D with A	Ar • Dr → Ar	A2	1	4
ANA E	And register E with A	Ar • Er→ Ar	A3	1	4
ANA H	And register H with A	Ar • Hr → Ar	A4	1	4
ANA L	And register L with A	Ar • Lr → Ar	A5	1	4
ANA M	And memory with A	Ar • (HL)$_m$ → Ar	A6	1	7
ANI	And immediate with A	Ar • n → Ar	E6 n	2	7
CALL	Call unconditional	PC → ST; aa → PC	CD aa	3	18
CC	Call on carry	Continue: C False Call aa: C True	DC aa	3	9/18
CM	Call on minus	Continue: N False Call aa: N True	FC aa	3	9/18
CMA	Complement A	$\overline{Ar}$ → Ar	2F	1	4
CMC	Complement carry	$\overline{C}$ → C	3F	1	4
CMP A	Compare register A with A	Ar – Ar; Flags Set	BF	1	4
CMP B	Compare register B with A	Ar – Br; Flags Set	B8	1	4
CMP C	Compare register C with A	Ar – Cr; Flags Set	B9	1	4
CMP D	Compare register D with A	Ar – Dr; Flags Set	BA	1	4
CMP E	Compare register E with A	Ar – Er; Flags Set	BB	1	4
CMP H	Compare register H with A	Ar – Hr; Flags Set	BC	1	4
CMP L	Compare register L with A	Ar – Lr; Flags Set	BD	1	4
CMP M	Compare memory with A	Ar – (HL)$_m$; Flag Set	BE	1	7

Mnemonic	Description	Boolean Expression	Machine Coding Operand(s)	Bytes	Clock Cycles
CNC	Call on no carry	Continue: C True Call aa: C False	D4 aa	3	9/18
CNZ	Call on no zero	Continue: Z True Call aa: Z False	C4 aa	3	9/18
CP	Call on positive	Continue: N True Call aa: N False	F4 aa	3	9/18
CPE	Call on parity even	Continue: P False Call aa: P True	EC	3	9/18
CPI	Compare immediate with A	Ar – n; Flags Set	FE n	2	7
CPO	Call on parity odd	Continue: P True Call aa: P False	E4 aa	3	9/18
CZ	Call on zero	Continue: Z False Call aa: Z True	CC aa	3	9/18
DAA	Decimal adjust A	Ar → Ar*	27	1	4
DAD B	ADD B & C to H& L	HLr + BCr → HLr	09	1	10
DAD D	Add D & E to H & L	HLr + DEr → HLr	19	1	10
DAD H	Add H & L to H & L	HLr + HLr → HLr	29	1	10
DAD SP	Add stack pointer to H & L	HLr + SPr → HLr	39	1	10
DCR A	Decrement register A	Ar – 1 → Ar	3D	1	4
DCR B	Decrement register B	Br – 1 → Br	05	1	4
DCR C	Decrement register C	Cr – 1 → Cr	0D	1	4
DCR D	Decrement register D	Dr – 1 → Dr	15	1	4
DCR E	Decrement register E	Er – 1 → Er	1D	1	4
DCR H	Decrement register H	Hr – 1 → Hr	25	1	4
DCR L	Decrement register L	Lr – 1 → Lr	2D	1	4
DCR M	Decrement memory	$(HL)_m - 1 \rightarrow (HL)_m$	35	1	10
DCX B	Decrement B & C	BCr – 1 → BCr	0B	1	6
DCX D	Decrement D & E	DEr – 1 → DEr	1B	1	6
DCX H	Decrement H & L	HLr – 1 → HLr	2B	1	6
DCX SP	Decrement stack pointer	SPr – 1 → SPr	3B	1	6
DI	Disable Interrupt	Set Interrupt Mask	F3	1	4
EI	Enable Interrupt	Clear Interrupt Mask	FB	1	4
HLT	Halt	Stop CPU and Wait For Interrupt	76	1	5
IN	Input	$(a)_{10}$ → Ar	DB a	2	10
INR A	Increment register A	Ar + 1 → Ar	3C	1	4
INR B	Increment register B	Br + 1 → Br	04	1	4
INR C	Increment register C	Cr + 1 → Cr	0C	1	4
INR D	Increment register D	Dr + 1 → Dr	14	1	4
INR E	Increment register E	Er + 1 → Er	1C	1	4
INR H	Increment register H	Hr + 1 → Hr	24	1	4
INR L	Increment register L	Lr + 1 → Lr	2C	1	4
INR M	Increment memory	$(HL)_m + 1 \rightarrow (HL)_m$	34	1	10
INX B	Increment B & C registers	BCr + 1 → BCr	03	1	6
INX D	Increment D & E registers	DEr + 1 → DEr	13	1	6

Mnemonic	Description	Boolean Expression	Machine Coding Operand(s)	Bytes	Clock Cycles
INX H	Increment H & L registers	HLr + 1 → HLr	23	1	6
INX SP	Increment stack pointer	SPr + 1 → SPr	33	1	6
JC	Jump on carry	Continue: C False aa → PCr: C True	DA aa	3	7/10
JM	Jump on minus	Continue: N False aa → PCr: N True	FA aa	3	7/10
JMP	Jump unconditional	aa → PCr	C3 aa	3	10
JNC	Jump on no carry	Continue: C True aa → PCr: C False	D2 aa	3	7/10
JNZ	Jump on no zero	Continue: Z True aa → PCr: Z False	C2 aa	3	7/10
JP	Jump on positive	Continue: N True aa → PCr: N False	F2 aa	3	7/10
JPE	Jump on parity even	Continue: P False aa → PCr: P True	EA aa	3	7/10
JPO	Jump on parity odd	Continue: P True aa → PCr: P False	E2 aa	3	7/10
JZ	Jump on zero	Continue: Z False aa → PCr: Z True	CA aa	3	7/10
LDA	Load A direct	$(aa)_m$ → Ar	3A aa	3	13
LDAX B	Load A indirect	$(BC)_m$ → Ar	0A	1	7
LDAX D	Load A indirect	$(DE)_m$ → Ar	1A	1	7
LHLD	Load H & L direct	$(aa+1)_m$:$(aa)_m$ → HLr	2A aa	3	16
LXI B	Load immediate register Pair B & C	aa → BCr	01 aa	3	10
LXI D	Load immediate register Pair D & E	aa → DEr	11 aa	3	10
LXI H	Load immediate register Pair H & L	aa → HLr	21 aa	3	10
LXI SP	Load immediate stack pointer	aa → SPr	31 aa	3	10
MOV M,A	Move register A to memory	Ar → $(HL)_m$	77	1	7
MOV M,B	Move register B to memory	Br → $(HL)_m$	70	1	7
MOV M,C	Move register C to memory	Cr → $(HL)_m$	71	1	7
MOV M,D	Move register D to memory	Dr → $(HL)_m$	72	1	7
MOV M,E	Move register E to memory	Er → $(HL)_m$	73	1	7
MOV M,H	Move register H to memory	Hr → $(HL)_m$	74	1	7
MOV M,L	Move register L to memory	Lr → $(HL)_m$	75	1	7
MOV A,A	Move register A to register A	Ar → Ar	7F	1	4
MOV A,B	Move register B to register A	Br → Ar	78	1	4
MOV A,C	Move register C to register A	Cr → Ar	79	1	4
MOV A,D	Move register D to register A	Dr → Ar	7A	1	4
MOV A,E	Move register E to register A	Er → Ar	7B	1	4
MOV A,H	Move register H to register A	Hr → Ar	7C	1	4
MOV A,L	Move register L to register A	Lr → Ar	7D	1	4
MOV A,M	Move memory to register A	(HL)r → Ar	7E	1	7
MOV B,A	Move register A to register B	Ar → Br	47	1	4
MOV B,B	Move register B to register B	Br → Br	40	1	4
MOV B,C	Move register C to register B	Cr → Br	41	1	4

Mnemonic	Description	Boolean Expression	Machine Coding Operand(s)	Bytes	Clock Cycles
MOV B,D	Move register D to register B	Dr → Br	42	1	4
MOV B,E	Move register E to register B	Er → Br	43	1	4
MOV B,H	Move register H to register B	Hr → Br	44	1	4
MOV B,L	Move register L to register B	Lr → Br	45	1	4
MOV B,M	Move memory to register B	$(HL)_m$ → Br	46	1	7
MOV C,A	Move register A to register C	Ar → Cr	4F	1	4
MOV C,B	Move register B to register C	Br → Cr	48	1	4
MOV C,C	Move register C to register C	Cr → Cr	49	1	4
MOV C,D	Move register D to register C	Dr → Cr	4A	1	4
MOV C,E	Move register E to register C	Er → Cr	4B	1	4
MOV C,H	Move register H to register C	Hr → Cr	4C	1	4
MOV C,L	Move register L to register C	Lr → Cr	4D	1	4
MOV C,M	Move memory to register C	$(HL)_m$ → Cr	4E	1	7
MOV D,A	Move register A to register D	Ar → Dr	57	1	4
MOV D,B	Move register B to register D	Br → Dr	50	1	4
MOV D,C	Move register C to register D	Cr → Dr	51	1	4
MOV D,D	Move register D to register D	Dr → Dr	52	1	4
MOV D,E	Move register E to register D	Er → Dr	53	1	4
MOV D,H	Move register H to register D	Hr → Dr	54	1	4
MOV D,L	Move register L to register D	Lr → Dr	55	1	4
MOV D,M	Move memory to register D	$(HL)_m$ → Dr	56	1	7
MOV E,A	Move register A to register E	Ar → Er	5F	1	4
MOV E,B	Move register B to register E	Br → Er	58	1	4
MOV E,C	Move register C to register E	Cr → Er	59	1	4
MOV E,D	Move register D to register E	Dr → Er	5A	1	4
MOV E,E	Move register E to register E	Er → Er	5B	1	4
MOV E,H	Move register H to register E	Hr → Er	5C	1	4
MOV E,L	Move register L to register E	Lr → Er	5D	1	4
MOV E,M	Move memory to register E	$(HL)_m$ → Er	5E	1	7
MOV H,A	Move register A to register H	Ar → Hr	67	1	4
MOV H,B	Move register B to register H	Br → Hr	60	1	4
MOV H,C	Move register C to register H	Cr → Hr	61	1	4
MOV H,D	Move register D to register H	Dr → Hr	62	1	4
MOV H,E	Move register E to register H	Er → Hr	63	1	4
MOV H,H	Move register H to register H	Hr → Hr	64	1	4
MOV H,L	Move register L to register H	Lr → Hr	65	1	4
MOV H,M	Move memory to register H	$(HL)_m$ → Hr	66	1	7
MOV L,A	Move register A to register L	Ar → Lr	6F	1	4
MOV L,B	Move register B to register L	Br → Lr	68	1	4
MOV L,C	Move register C to register L	Cr → Lr	69	1	4
MOV L,D	Move register D to register L	Dr → Lr	6A	1	4
MOV L,E	Move register E to register L	Er → Lr	6B	1	4
MOV L,H	Move register H to register L	Hr → Lr	6C	1	4
MOV L,L	Move register L to register L	Lr → Lr	6D	1	4
MOV L,M	Move memory to register L	$(HL)_m$ → Lr	6E	1	7

Mnemonic	Description	Boolean Expression	Machine Coding Operand(s)	Bytes	Clock Cycles
MVI A	Move immediate register A	n → Ar	3E n	2	7
MVI B	Move immediate register B	n → Br	06 n	2	7
MVI C	Move immediate register C	n → Cr	0E n	2	7
MVI D	Move immediate register D	n → Dr	16 n	2	7
MVI E	Move immediate register E	n → Er	1E n	2	7
MVI H	Move immediate register H	n → Hr	26 n	2	7
MVI L:	Move immediate register L	n → Lr	2En	2	7
NVI M	Move immediate memory	n → $(HL)_m$	36 n	2	10
NOP	No-operation	——	00	1	4
ORA A	Or register A with A	Ar + Ar → Ar	B7	1	4
ORA B	Or register B with A	Ar + Br → Ar	B0	1	4
ORA C	Or register C with A	Ar + Cr → Ar	B1	1	4
ORA D	Or register D with A	Ar + Dr → Ar	B2	1	4
ORA E	Or register E with A	Ar + Er → Ar	B3	1	4
ORA H	Or register H with A	Ar + Hr → Ar	B4	1	4
ORA L	Or register L with A	Ar + Lr → Ar	B5	1	4
ORA M	Or memory with A	Ar + (HL)r → Ar	B6	1	7
ORI	Or immediate with A	Ar + n → Ar	F6 n	2	7
OUT	Output	Ar → $(a)_{10}$	D3 a	2	10
PCHL	H & L to program counter	HLr → PCr	E9	1	6
POP B	Pop register Pair B & C off stack	ST → BCr; SP + 2 → SPr	C1	1	10
POP D	Pop register Pair D & E off stack	ST → DEr; SP + 2 → SPr	D1	1	10
POP H	Pop register Pair H & L off stack	ST → HLr; SP + 2 → SPr	E1	1	10
POP PSW	Pop A and Flags off stack	ST → AFr; SP + 2 → SPr	F1	1	10
PUSH B	Push register Pair B & C on stack	SP – 2 → SP; BCr → ST	C5	1	12
PUSH D	Push register Pair D & E on stack	SP – 2 → SP; DEr → ST	D5	1	12
PUSH H	Push register Pair H & L on stack	SP – 2 → SP; HLr → ST	E5	1	12
PUSH PSW	Push A and Flags on stack	SP – 2 → SP; AFr → ST	F5	1	12
RAL	Rotate A left through carry		17	1	4
RAR	Rotate A right through carry		1F	1	4
RC	Return on carry	Continue: C False RET: C True	D8	1	6/12
RET	Return	ST → PCr; SP + 2 → SPr	C9	1	10
RIM	Read Interrupt Mask	Ir → Ar	20	1	4

Mnemonic	Description	Boolean Expression	Machine Coding Operand(s)	Bytes	Clock Cycles
RLC	Rotate A left		07	1	4
RM	Return on no minus	Continue: N False RET: N True	F8	1	6/12
RNC	Return on no carry	Continue: C True RET: C False	D0	1	6/12
RNZ	Return on no zero	Continue: Z True RET: Z False	C0	1	6/12
RP	Return on positive	Continue: N True RET: N False	F0	1	6/12
RPE	Return on parity even	Continue: P False RET: P True	E8	1	6/12
RPO	Return on parity odd	Continue: P True RET: P False	E0	1	6/12
RRC	Rotate A right		0F	1	4

RST 0	Restart	PCr → ST;
		SPr – 2 → SPr;
		0000H → PCr
			C7	1	12

RST 1	Restart	PCr → ST;
		SPr – 2 → SPr;
		0008H → PCr
			CF	1	12

RST 2	Restart	PCr → ST;
		SPr – 2 → SPr;
		0010H → PCr
			D7	1	12

RST 3	Restart	PCr → ST;
		SPr – 2 → SPr;
		0018H → PCr
			DF	1	12

RST 4	Restart	PCr → ST;
		SPr – 2 → SPr;
		0020H → PCr
			E7	1	12

RST 5	Restart	PCr → ST;
		SPr – 2 → SPr;
		0028H → PCr
			EF	1	12

RST 6	Restart	PCr → ST;
		SPr – 2 → SPr;
		0030H → PCr
			F7	1	12

RST 7	Restart	PCr → ST;
		SPr – 2 → SPr;
		0038H → PCr
			FF	1	12

RZ	Return on zero	Continue: Z False RET: Z True	C8	1	6/12
SBB A	Subtract register A from A with borrow	Ar – Ar – C → Ar	9F	1	4
SBB B	Subtract register B from A with borrow	Ar – Br – C → Ar	98	1	4
SBB C	Subtract register C from A with borrow	Ar – Cr – C → Ar	99	1	4

Mnemonic	Description	Boolean Expression	Machine Coding Operand(s)	Bytes	Clock Cycles
SBB D	Subtract register D from A with borrow	Ar – Dr – C → Ar	9A	1	4
SBB E	Subtract register E from A with borrow	Ar – Er – C → Ar	9B	1	4
SBB H	Subtract register H from A with borrow	Ar – Hr – C → Ar	9C	1	4
SBB L	Subtract register L from A with borrow	Ar – Lr – C → Ar	9D	1	4
SBB M	Subtract memory from A with borrow	Ar – (HL)r – C → Ar	9E	1	7
SBI	Subtract immediate from A with borrow	Ar – n – C → Ar	DE n	2	7
SHLD	Store H & L direct	HLr → $(aa+1)_m$;$(aa)_m$	22 aa	3	16
SIM	Set interrupt mask	Ar→ Ir	30	1	4
SPHL	H & L to stack pointer	HLr → SPr	F9	1	6
STA	Store A direct	Ar → $(aa)_m$	32 aa	3	13
STAX B	Store A indirect	Ar → $(BC)_m$	02	1	7
STAX D	Store A indirect	Ar → $(DE)_m$	12	1	7
STC	Set carry	1 → C	37	1	4
SUB A	Subtract register A from A	Ar – Ar → Ar	97	1	4
SUB B	Subtract register B from A	Ar – Br → Ar	90	1	4
SUB C	Subtract register C from A	Ar – Cr → Ar	91	1	4
SUB D	Subtract register D from A	Ar – Dr → Ar	92	1	4
SUB E	Subtract register E from A	Ar – Er → Ar	93	1	4
SUB H	Subtract register H from A	Ar – Hr → Ar	94	1	4
SUB L	Subtract register L from A	Ar – Lr → Ar	95	1	4
SUB M	Subtract memory from A	Ar – $(HL)_m$ → Ar	96	1	7
SUI	Subtract immediate from A	Ar – n → Ar	D6 n	2	7
XCHG	Exchange D & E H & L registers	DEr→ HLr:HLr→ DEr	EB	1	4
XRA A	Exclusive Or register A with A	Ar ⊕ Ar → Ar	AF	1	4
XRA B	Exclusive Or register B with A	Ar ⊕ Br → Ar	A8	1	4
XRA C	Exclusive Or register C with A	Ar ⊕ Cr → Ar	A9	1	4
XRA D	Exclusive Or register D with A	Ar ⊕ Dr → Ar	AA	1	4
XRA E	Exclusive Or register E with A	Ar ⊕ Er → Ar	AB	1	4
XRA H	Exclusive Or register H with A	Ar ⊕ Hr → Ar	AC	1	4
XRA L	Exclusive Or register L with A	Ar ⊕ Lr → Ar	AD	1	4
XRA M	Exclusive Or memory with A	Ar ⊕ (HL)r → Ar	AE	1	7
XRI	Exclusive Or immediate with A	Ar ⊕ n → Ar	EE n	2	7
XTHL	Exchange top of stack with H & L	HLr → ST: ST → HLr	E3	1	16

NOTE: Two possible cycle times, (6/12) indicate instruction cycles dependent on condition flags.
* Ar is translated from Binary representation into BCD (Binary Coded Decimal).

ADDRESSING: a = 8-bit I/O addressing.
aa = 16-bit address (least significant byte first, most significant byte last).
n = 8-bit data.
$(a)_{IO}$ = is data at I/O) address.
$(aa)_m$ = is data at address aa.
ST = represents top of stack.
Xr = 8-bit register.
XXr = 16-bit register.

QUICK REFERENCE GUIDE

MEMORY MAP

ADDRESS	MEMORY
0000H 003FH	Monitor Interrupt Table
0040H 006FH	Monitor Subroutine Jump Table
0070H 5FFFH	Monitor ROM Area
6000H 67FFH	Monitor Data and Stack Area
6800H 6820H	User Programmable Interrupt Vector Jump Table
6821H 6FFFH	User Data and Stack Area
7000H BFFFH	User Program Area
A000H BFFFH	8K Expansion (External) Memory Area
C000H FFFFH	Memory Module Area

I/O MEMORY MAP

00H 0FH	Keyboard Column 1
20H 1FH	Keyboard Column 2
20H 2FH	Keyboard Column 3
30H 3FH	Local Latch (Bit 7 controls LED #4 Bit 6 controls LED #3 Bit 5 controls LED #2 Bit 4 controls LED #1 Bit 3 controls A/D Range Bit 2 controls A/D Input Line Select Bit 1 controls A/D Input Line Select Bit 0 controls A/D Input Line Select)
40H, 44H, 48H, 4CH	Timer Register 1 (Use 40H to access.)
41H, 45H, 49H, 4DH	Timer Register 2 (Use 41H to access.)
42H, 46H, 4AH, 4EH	Timer Register 3 (Use 42H to access.)
43H, 47H, 4BH, 4FH	Timer Register 4 (Use 43H to access.)
50H 5FH	General I/O 0
60H 6FH	General I/O 1
70H 7FH	General I/O 2
80H 83H	A/D Input/D/A Output (Use 80H to access.)
90H 93H	Input Port
A0H A3H	Output Port
B0H, B2H.	LCD Data Register (Use B0H to access.)
B1H, B3H	LCD Command Register (Use B1H to Access.)

SUBROUTINE JUMP TABLE*

EPROM ADDRESS	LABEL	INTRODUCTION
0040	JMP MAIN	Return Control to Monitor
0043	JMP GETC	Returns ASCII Character in A
0046	JMP PUTC	Outputs ASCII Character in A
0049	JMP SETRS232	Select RS-232 for I/O
004C	JMP SETLCD	Select LCD, Keypad for I/O
004F	JMP GETHBYT	Get a Hex Byte, Return in A
0052	JMP GETHWRD	Get a Hex Word, Return in HL
0055	JMP PUTHBYT	Print Byte in A, Out in Hex
0058	JMP PUTHWRD	Print Word in HL, Out in Hex
005B	JMP PUTSTRING	Print a Null Terminated String Pointed to by HL
005E	JMP COPYUP	Copy Block of Memory, B=# Copy, Push Source, Push Destination
0061	JMP COPYDOWN	Copy Block of Memory, B=# Copy, Push Source, Push Destination
0064	JMP COPY	Push Incordec, # of Bytes, Destination, D=Source
0067	JMP PAUSE	Wait Over a Second
006A	JMP PUTDSWRD	Convert Signed Value in HL to Decimal and Print Out
006D	JMP PUTUDWRD	Convert Unsigned Value in HL to Decimal and Print Out

* You must use the CALL command (CDH) when calling a subroutine from the Jump Table.

INTERRUPT VECTOR TABLE

ADDRESS	INTERRUPT SOURCE
6801H, 6802H	Reset 1H
6804H, 6805H	Reset 2H
6807H, 6808H	Reset 3H
680AH, 680BH	Reset 4H
680DH, 680EH	TRAP (NMI)
6810H, 6811H	Reset 5H
6813H, 6814H	Reset 5.5H
6816H, 6817H	Reset 6H
6819H, 681AH	Reset 6.5H
681CH, 681DH	Reset 7H
681FH, 6820H	Reset 7.5H

KEY NAMES/FUNCTIONS

– -RETURN: Exit a function.

+ -LIST: Enter "+" character or disassemble code.

E-BAUD: Enter letter "E" or change baud rate.

D-RS232: Enter letter "D" or transfer control to serial port.

A-LOAD: Enter letter "A" or load program at selected address.

7-BRK: Enter numeral "7" or permit entry of break points.

8-W REG: Enter numeral "8" or set register break value.

4-M BLK: Enter numeral "4" or move a block of memory.

1-EXM MEM: Enter numeral "1" or display memory value.

0-SS: Enter numeral "0" or single step a program.

2-EXM REG: Enter numeral "2" or examine and modify CPU registers.

HELP-RPO: Get help or exit a function.

NMI: Interrupt any program or operation.

RESET: Reset system.

3-GO: Enter numeral "3" or execute a program.

6-DOWN: Enter numeral "6" or download a file to RAM.

5-I BLK: Enter numeral "5" or set a block of memory to a value.

9-W LOC: Enter numeral "9" or set address break value.

C-DUP: Enter letter "C" or copy memory cartridge.

B-SAVE: Enter letter "B" or save file to Memory Module.

SIGNAL CONNECTOR BLOCKS

UPPER BLOCK

D0	CPU Data Bus Line
D1	CPU Data Bus Line
D2	CPU Data Bus Line
D3	CPU Data Bus Line
D4	CPU Data Bus Line
D5	CPU Data Bus Line
D6	CPU Data Bus Line
D7	CPU Data Bus Line
GTI	Timer Gate Input 1
GTI	Timer Gate Input 2
TCK1	Timer Clock Input 1
TCK0	Timer Clock Input 0
TO1	Timer Output 1
TO0	Timer Output 0
ENEXM	Internal RAM Disable
A15	CPU Address Bus
A14	CPU Address Bus
A13	CPU Address Bus
A12	CPU Address Bus
A11	CPU Address Bus
A10	CPU Address Bus
A9	CPU Address Bus
A8	CPU Address Bus
A7	CPU Address Bus
A6	CPU Address Bus
A5	CPU Address Bus
A4	CPU Address Bus
A3	CPU Address Bus
A2	CPU Address Bus
A1	CPU Address Bus
A0	CPU Address Bus
READ	Memory, I/O Read Line
WRITE	Memory, I/O Write Line
RESET	Reset Output
ALE	Address Latch Enable
READY	CPU Ready Line
I/O 0	Decoded I/O Line
I/O 1	Decoded I/O Line
I/O 2	Decoded I/O Line
ADSTR	A/D Converter Start Line

LOWER BLOCK

LPIN	Logic Probe Input
LPAUD	Logic Probe Audio Disable
GND	Digital Ground
+5V	+5 Volt DC Output
AGND	Analog Ground
–12V	–12 Volt DC Output
+12V	+12 Volt DC Output
AI7	Analog Input
A16	Analog Input
A15	Analog Input
A14	Analog Input
A13	Analog Input
A12	Analog Input
A11	Analog Input
A10	Analog Input
AOUT	Analog Output
HDA	CPU Hold Acknowledge
INA	CPU Interrupt Acknowledge
HOLD	CPU Hold
INR	CPU Interrupt Request
R6.5	CPU Restart Interrupt 6.5
R5.5	CPU Restart Interrupt 5.5
ADBUSY	A/D, D/A Busy Status Output
S0	CPU Machine Cycle Status Line 0
S1	CPU Machine Cycle Status Line 1
IO/M	CPU Input/Output Memory
CLK	CPU System Clock
ADINT	A/D, D/A Interrupt Output
EXMSL	External Memory Select
IPL	Input Port Latch
IP7	Input Data Port
IP6	Input Data Port
IP5	Input Data Port
IP4	Input Data Port
IP3	Input Data Port
IP2	Input Data Port
IP1	Input Data Port
IP0	Input Data Port
OP7	Output Data Port
OP6	Output Data Port
OP5	Output Data Port
OP4	Output Data Port
OP3	Output Data Port
OP2	Output Data Port
OP1	Output Data Port
OP0	Output Data Port
GND	Digital Ground

NOTE:
Red areas are output only. Blue areas are input only. Green areas are bi-directional.

BACKPACK CONNECTOR

1- Analog Input 0
2- Analog Input 1
3- CPU Hold Acknowledge
4- Analog Output
5- CPU Hold
6- Interrupt Acknowledge
7- Reset Interrupt 6.5
8- Interrupt Request
9- Ground
10- Reset Interrupt 5.5
11- Ground
12- CPU Status Line S0
13- Ground
14- CPU Status Line S1
15- Ground
16- I/O Memory Select Line
17- Ground
18- System Clock
19- Ground
20- CPU Address Line 5
21- Ground
22- CPU Address Line 4
23- Ground
24- CPU Address Line 3
25- Ground
26- CPU Address Line 2
27- Ground
28- CPU Address Line 1
29- Ground
30- CPU Address Line 0
31- Ground
32- CPU Read
33- Ground
34- CPU Write
35- Ground
36- CPU Reset Out Line
37- Ground
38- CPU Address Latch Enable
39- Ground
40- CPU Ready
41- Ground
42- General I/O Select 1
43- Ground
44- General I/O Select 2
45- Ground
46- CPU Data Line 0
47- Ground
48- CPU Data Line 1
49- Ground
50- CPU Data Line 2
51- Ground
52- CPU Data Line 3
53- Ground
54- CPU Data Line 4
55- Ground
56- CPU Data Line 5
57- Ground
58- CPU Data Line 6
59- Ground
60- CPU Data Line 7

NOTE:
All signal lines are protected by series resistors.

SERVICE INFORMATION

TECHNICAL CONSULTATION

You can write or call our Technical Consultants for help with any Heath product, or for answers to questions about the use of these products.

The completeness and accuracy of the consultation you receive depends entirely on the information you provide. Be sure to include:

1. The Model Number.
2. Date of purchase.
3. An exact description of the difficulty. Include switch positions, connections to other units, operating procedures, and any other information you think might be helpful.
4. List everything you have done in attempting to correct the difficulty.

REPLACEMENT PARTS

If a replacement part listed in your manual is needed, send a letter including the following information to Heath Company, Parts Department, Benton Harbor, MI 49022:

1. Heath part number and description from your Manual's Parts List.
2. Model Number of the product.

If your product is in the Warranty period, add:

1. Date, location, and invoice number of the purchase.
2. Nature of the defect.

FACTORY SERVICE

You can return your product to the Heath Company Service Department to have it repaired or replaced (at Heath Company's option) for a minimum fee. Products that have been modified will not be accepted for service.

SHIPPING INSTRUCTIONS

Check the product to see that all parts are in place. Then wrap the product in heavy paper. Place the product in a strong carton, and put at least three inches of resilient packing material (shredded paper, excelsior, etc.) on all sides between the product and the carton.

Seal the carton with gummed paper tape. Ship it by prepaid UPS or insured Parcel Post to:

HEATH COMPANY
SERVICE DEPARTMENT
Benton Harbor, Michigan 49022

Include a letter, containing the following information:

1. Your name and return address.
2. Date of purchase.
3. Complete description of the difficulty.
4. Your authorization to ship the repaired product back to you C.O.D. for the service and shipping charges.

Heath Phone Directory for Assistance and Information

Technical Assistance (8:00 A.M. to 4:30 P.M. Eastern Time, Weekdays only):

• Educational Products	616-982-3980
• Amateur Radio	616-982-3296
• Test Equipment, Weather Instruments, Clocks	616-982-3315
• Television	616-982-3307
• Home Products, Stereo, Security, Telephone, Marine, Automotive	616-982-3496
• Computer Hardware	616-982-3309
• Replacement Parts Orders; (please have Heath part number available when you call)	616-982-3571

YOUR HEATH ASSEMBLED PRODUCT ONE-YEAR LIMITED WARRANTY

Welcome to the Heath family. We believe you will be pleased with the performance of your new Heath assembled product. Please read this Consumer Protection Plan carefully. It is a "LIMITED WARRANTY" as defined in the U.S. Consumer Product Warranty and Federal Trade Improvement Act. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

HEATH'S RESPONSIBILITY

PARTS — Replacements for factory defective parts will be supplied free for one year from date of purchase. Replacement parts are warranted for the remaining portion of the original warranty period. You can obtain warranty parts direct from Heath Company by writing or telephoning us at (616) 982-3571. And we will pay shipping charges to get those parts to you... anywhere in the world.

SERVICE LABOR — For a period of one year from the date of purchase, any malfunction caused by defective parts or workmanship will be corrected at no charge to you. Heath Company reserves the right to repair or replace the product, at our option. You must deliver the product at your expense to the Heath factory, any Heath/Zenith Computers and Electronics Center (units of Veritechnology Electronics Corporation), or any of our authorized overseas distributors.

TECHNICAL CONSULTATION — You will receive free consultation on any problem you might encounter in the use of your Heath product. Just drop us a line or give us a call. Sorry, we cannot accept collect calls.

NOT COVERED — Repair service, adjustments, calibration, and damage due to misuse, abuse, or negligence are not covered by the warranty. Unauthorized service or modification of the product or of any furnished component will void this warranty in its entirety. This warranty does not include reimbursement for inconvenience, loss of use, set-up time, or unauthorized service.

This warranty covers only Heath assembled products and is not extended to other equipment or components that a customer uses in conjunction with our products.

SUCH REPAIR AND REPLACEMENT SHALL BE THE SOLE REMEDY OF THE CUSTOMER AND THERE SHALL BE NO LIABILITY ON THE PART OF HEATH FOR ANY SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING BUT NOT LIMITED TO ANY LOSS OF BUSINESS OR PROFITS, WHETHER OR NOT FORESEEABLE.

Some states do not allow the exclusion of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

OWNER'S RESPONSIBILITY

EFFECTIVE WARRANTY DATE — Warranty begins on the date of first consumer purchase. You must supply a copy of your proof of purchase when you request warranty service.

OPERATING MANUAL — Read your operating instructions carefully so that you will fully understand the proper operation and function of your product.

ACCESSORY EQUIPMENT — Performance malfunctions involving connections to (or interfacing with) other non-Heath equipment are not covered by this warranty and are the owner's responsibility.

SHIPPING UNITS — Follow the packing instructions published in your manual. Damage due to inadequate packing cannot be repaired under warranty.

If you are not satisfied with our service (warranty or otherwise) or our products, write directly to our Director of Customer Service, Heath Company, Benton Harbor MI 49022. He will make certain your problems receive immediate, personal attention.