HEATH COMPANY PHONE DIRECTORY

The following telephone numbers are direct lines to the departments listed:

Kit orders and delivery information .................................. (616) 982-3411
Credit ................................................................. (616) 982-3561
Replacement Parts ....................................................... (616) 982-3571

Technical Assistance Phone Numbers
8:00 A.M. to 12 P.M. and 1:00 P.M. to 4:30 P.M., EST, Weekdays Only
R.C. Audio, and Electronic Organs .................................. (616) 982-3310
Amateur Radio .......................................................... (616) 982-3296
Test Equipment, Weather Instruments and
Home Clocks ............................................................. (616) 982-3315
Television ................................................................. (616) 982-3307
Aircraft, Marine, Security, Scanners, Automotive,
Appliances and General Products .................................. (616) 982-3496
Computers ............................................................... (616) 982-3309

YOUR HEATHKIT 90-DAY LIMITED WARRANTY

For a period of ninety (90) days after purchase, Heath Company will replace or repair free of charge any parts that
are defective either in materials or workmanship. You can obtain parts directly from Heath Company by writing us at
the address below or by telephoning us at (616) 982-3571. And we'll pay shipping charges to get those parts to you —
anywhere in the world.

We warrant that during the first ninety (90) days after purchase, our products, when correctly assembled, calibrated,
adjusted and used in accordance with our printed instructions, will meet published specifications.

If a defective part or error in design has caused your Heathkit product to malfunction during the warranty period
through no fault of yours, we will service it free upon proof of purchase and delivery at your expense to the Heath
factory, any Heathkit Electronic Center, or any of our authorized overseas distributors.

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

Our warranty does not cover and we are not responsible for damage caused by: incorrect assembly, the use of
corrosive solvent, defective tools, misuse, or fire; or by unauthorized modifications to or uses of our products for
purposes other than as advertised. Our warranty does not include reimbursement for inconvenience, loss of use,
customer assembly or set-up time.

This warranty covers only Heathkit products and is not extended to allied equipment or components used in
conjunction with our products. We are not responsible for accidental or consequential damages. Some states
do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion
may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary
from state to state.

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

HEATH COMPANY
BENTON HARBOR, MI 49022

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.
Heathkit® Manual

for the

MICROPROCESSOR TRAINER
Model ET-3400

595-2021-06
TABLE OF CONTENTS

Introduction ........................................ 3
Assembly Notes ....................................... 4
Parts List ........................................... 7

Step-by-Step Assembly
   Main Circuit Board ............................. 10
   Keyboard Circuit Board ....................... 24
   Support Bracket Assembly .................... 27
   Cabinet Assembly and Wiring .................. 30

Initial Tests
   Voltage Tests ..................................... 34
   Tests Continued .................................. 35
   Operational Tests ................................ 37

Final Assembly ..................................... 42
Operation .......................................... 45

In Case of Difficulty ............................... 91
Troubleshooting Charts ............................ 93
Specifications ...................................... 107
Theory of Operation ............................... 108
Semiconductor Identification Charts .......... 110

Circuit Board
X-Ray View ........... (Illustration Booklet, Page 13)

Schematic .......................... Fold-in
Warranty ............................. Inside front cover
Customer Service .................. Inside rear cover
INTRODUCTION

The ET-3400 Microcomputer Learning System is a practical, low cost microprocessor trainer: designed as a learning tool to teach microprocessor operation, programming, and applications. The ET-3400 Trainer is designed to accompany the EE-3401 Individual Learning Program on microprocessors. All of the programming and hardware interface experiments supplied with this course are implemented on the Trainer. While the Trainer was designed primarily to accompany this course, it is a flexible, general-purpose training unit and microprocessor breadboard. It can be used in many other applications that require a low cost microprocessor-based software development system or as a design aid for developing special interfaces.

MAIN FEATURES

- Uses the popular 6800 Microprocessor.
- Is supplied with 256 bytes of semiconductor RAM (expandable to 512 bytes).
- Features 1K ROM monitor program.
- Has hexadecimal keyboard for rapid data and program entry.
- Has six digits of hexadecimal display for reading out memory addresses, their contents, and register contents.
- Uses breadboarding sockets that permit rapid, solderless assembly of IC logic circuitry to be used with the microprocessor. They are ideal for prototyping special interface circuits.
- The microprocessor address bus, data bus, control lines, and associated signals are buffered and terminated on front panel connectors; allowing complete freedom in experimenting with the microprocessor and its associated circuitry.
- Has eight individual, independent, binary LED indicators for monitoring logic states in the breadboard circuitry.
- Has eight individual, independent, binary data switches that can be used for supplying binary words and logic levels in the breadboarding circuitry.
- The built-in power supplies furnish power to all internal circuitry and have sufficient reserve to power breadboard circuits. The +5 and ±12-volt supply voltages are connected to front panel connectors.
- Has provision for future expansion of memory and I/O capabilities.
ASSEMBLY NOTES

TOOLS

You will need these tools to assemble your kit.

1. Follow the instructions carefully. Read the entire step before you perform each operation.

2. The illustrations in the Manual are called Pictorials and Details. Pictorials show the overall operation for a group of assembly steps; Details generally illustrate a single step. When you are directed to refer to a certain Pictorial “for the following steps,” continue using that Pictorial until you are referred to another Pictorial for another group of steps.

3. Most kits use a separate “Illustration Booklet” that contains illustrations (Pictorials, Details, etc.) that are too large for the Assembly Manual. Keep the “Illustration Booklet” with the Assembly Manual. The illustrations in it are arranged in Pictorial number sequence.

4. Position all parts as shown in the Pictorials.

5. Solder a part or a group of parts only when you are instructed to do so.
6. Each circuit part in an electronic kit has its own component number (R2, C4, etc.). Use these numbers when you want to identify the same part in the various sections of the Manual. These numbers, which are especially useful if a part has to be replaced, appear:

— In the Parts List,

— At the beginning of each step where a component is installed,

— In some illustrations,

— In the Schematic,

— In the section at the rear of the Manual.

7. When you are instructed to cut something to a particular length, use the scales (rulers) provided at the bottom of the Manual pages.

SAFETY WARNING: Avoid eye injury when you cut off excess lead lengths. Hold the leads so they cannot fly toward your eyes.

SOLDERING

Soldering is one of the most important operations you will perform while assembling your kit. A good solder connection will form an electrical connection between two parts, such as a component lead and a circuit board foil. A bad solder connection could prevent an otherwise well-assembled kit from operating properly.

It is easy to make a good solder connection if you follow a few simple rules:

1. Use the right type of soldering iron. A 25 to 40-watt pencil soldering iron with a 1/8" or 3/16" chisel or pyramid tip works best.

2. Keep the soldering iron tip clean. Wipe it often on a wet sponge or cloth; then apply solder to the tip to give the entire tip a wet look. This process is called tinning, and it will protect the tip and enable you to make good connections. When solder tends to “ball” or does not stick to the tip, the tip needs to be cleaned and reinned.
PARTS

Resistors will be called out by their resistance value in Ω (ohms), kΩ (kilohms), or MΩ (megohms). Certain types of resistors will have the value printed on the body, while others will be identified by a color code. The colors of the bands and the value will be given in the steps, therefore the following color code is given for information only.

5-BAND RESISTORS
(±1%)

4-BAND RESISTORS
(±10%)
(±5%)

<table>
<thead>
<tr>
<th>Band 1</th>
<th>Band 2</th>
<th>Band 3</th>
<th>Multiplier</th>
<th>Resistance Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Color</td>
</tr>
<tr>
<td>Color</td>
<td>1st Digit</td>
<td>Color</td>
<td>2nd Digit</td>
<td>Color</td>
</tr>
<tr>
<td>Black</td>
<td>0</td>
<td>Black</td>
<td>0</td>
<td>Black</td>
</tr>
<tr>
<td>Brown</td>
<td>1</td>
<td>Brown</td>
<td>1</td>
<td>Brown</td>
</tr>
<tr>
<td>Red</td>
<td>2</td>
<td>Red</td>
<td>2</td>
<td>Red</td>
</tr>
<tr>
<td>Orange</td>
<td>3</td>
<td>Orange</td>
<td>3</td>
<td>Orange</td>
</tr>
<tr>
<td>Yellow</td>
<td>4</td>
<td>Yellow</td>
<td>4</td>
<td>Yellow</td>
</tr>
<tr>
<td>Green</td>
<td>5</td>
<td>Green</td>
<td>5</td>
<td>Green</td>
</tr>
<tr>
<td>Blue</td>
<td>6</td>
<td>Blue</td>
<td>6</td>
<td>Blue</td>
</tr>
<tr>
<td>Violet</td>
<td>7</td>
<td>Violet</td>
<td>7</td>
<td>Violet</td>
</tr>
<tr>
<td>Gray</td>
<td>8</td>
<td>Gray</td>
<td>8</td>
<td>Gray</td>
</tr>
<tr>
<td>White</td>
<td>9</td>
<td>White</td>
<td>9</td>
<td>White</td>
</tr>
</tbody>
</table>

Capacitors will be called out by their capacitance value in μF (microfarads) or pF (picofarads) and type: ceramic, Mylar*, electrolytic, etc. Some capacitors may have their value printed in the following manner:

First digit of capacitor's value: 1
Second digit of capacitor's value: 5
Multiplier: Multiply the first & second digits by the proper value from the Multiplier Chart.

To find the tolerance of the capacitor, look up this letter in the Tolerance columns.

EXAMPLES:

151K = 15 X 10 = 150 pF
759 = 75 X 0.1 = 7.5 pF

NOTE: The letter “R” may be used at times to signify a decimal point: as in: 2R2 = 2.2 (pF or μF).

*DuPont Registered Trademark
PARTS LIST

Check each part against the following list. Any part that is packed in an individual envelope with the part number on it should be placed back in the envelope after you identify it until it is called for in a step. Do not discard any packing materials until all parts are accounted for.

The key numbers correspond to the numbers on the "Parts Pictorial" in the separate "Illustration Booklet" on Pages 1 and 2.

To order a replacement part: Always include the PART NUMBER. Use the Parts Order Form furnished with the kit. If one is not available, see "Replacement Parts" inside the rear cover of the Manual. Your Warranty is located inside the front cover. For prices, refer to the separate "Heath Parts Price List."

<table>
<thead>
<tr>
<th>KEY HEATH No.</th>
<th>HEALTH Part No.</th>
<th>QTY.</th>
<th>DESCRIPTION</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELATED</td>
<td>COMPONENTS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESISTORS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTES:
1. All resistors are 10% tolerance unless otherwise noted. A fourth color band of silver indicates a 10% tolerance; a fourth band of gold indicates 5% tolerance.

2. The resistors may be packed in more than one envelope. Open all the resistor envelopes in this pack before you check them against the Parts List.

1/4-Watt Resistors

<table>
<thead>
<tr>
<th>KEY HEATH No.</th>
<th>HEALTH Part No.</th>
<th>QTY.</th>
<th>DESCRIPTION</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 6-153-12</td>
<td>1</td>
<td>15 kΩ (brown-green-orange)</td>
<td>R8</td>
<td></td>
</tr>
<tr>
<td>A1 6-273-12</td>
<td>2</td>
<td>27 kΩ (red-violet-orange)</td>
<td>R1, R42</td>
<td></td>
</tr>
<tr>
<td>A1 6-104-12</td>
<td>3</td>
<td>100 kΩ (brown-black-yellow)</td>
<td>R11, R12, R14</td>
<td></td>
</tr>
<tr>
<td>A1 6-154-12</td>
<td>1</td>
<td>150 kΩ (brown-green-yellow)</td>
<td>R9</td>
<td></td>
</tr>
<tr>
<td>A1 6-224-12</td>
<td>2</td>
<td>220 kΩ (red-red-yellow)</td>
<td>R7, R50</td>
<td></td>
</tr>
<tr>
<td>A1 6-824-12</td>
<td>1</td>
<td>820 kΩ (gray-red-yellow)</td>
<td>R13</td>
<td></td>
</tr>
</tbody>
</table>

Other Resistors

<table>
<thead>
<tr>
<th>KEY HEATH No.</th>
<th>HEALTH Part No.</th>
<th>QTY.</th>
<th>DESCRIPTION</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2 6-680</td>
<td>2</td>
<td>68 Ω, 1/2-watt (blue-gray-black)</td>
<td>R3, R4</td>
<td></td>
</tr>
</tbody>
</table>

CAPACITORS

Electrolytic Capacitors

<table>
<thead>
<tr>
<th>KEY HEATH No.</th>
<th>HEALTH Part No.</th>
<th>QTY.</th>
<th>DESCRIPTION</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1 25-200</td>
<td>1</td>
<td>68 μF tantalum</td>
<td>C13</td>
<td></td>
</tr>
<tr>
<td>B1 25-221</td>
<td>2</td>
<td>2.2 μF tantalum</td>
<td>C8, C9</td>
<td></td>
</tr>
<tr>
<td>B1 25-220</td>
<td>2</td>
<td>10 μF tantalum (10M)</td>
<td>C11, C12</td>
<td></td>
</tr>
<tr>
<td>B2 25-241</td>
<td>2</td>
<td>1200 μF</td>
<td>C6, C7</td>
<td></td>
</tr>
<tr>
<td>B2 25-272</td>
<td>1</td>
<td>6000 μF</td>
<td>C1</td>
<td></td>
</tr>
</tbody>
</table>

Other Capacitors

<table>
<thead>
<tr>
<th>KEY HEATH No.</th>
<th>HEALTH Part No.</th>
<th>QTY.</th>
<th>DESCRIPTION</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>B3 20-102</td>
<td>1</td>
<td>100 pF mica</td>
<td>C23</td>
<td></td>
</tr>
<tr>
<td>B4 21-176</td>
<td>12</td>
<td>.01 μF ceramic</td>
<td>C4, C5, C14 through C22, C24</td>
<td></td>
</tr>
<tr>
<td>B5 27-85</td>
<td>2</td>
<td>.22 μF Mylar</td>
<td>C2, C3</td>
<td></td>
</tr>
</tbody>
</table>
### DIODES

<table>
<thead>
<tr>
<th>No.</th>
<th>Part No.</th>
<th>QTY.</th>
<th>Description</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>56-56</td>
<td>4</td>
<td>1N4149 diode</td>
<td>D7 through D10</td>
</tr>
<tr>
<td>C1</td>
<td>57-42</td>
<td>2</td>
<td>3A1 diode</td>
<td>D1, D2</td>
</tr>
<tr>
<td>C1</td>
<td>57-65</td>
<td>4</td>
<td>1N4002 diode</td>
<td>D3, D4, D5, D6</td>
</tr>
</tbody>
</table>

### INTEGRATED CIRCUITS (IC's)

**NOTES:**

1. Integrated circuits are marked for identification in one of the following four ways:
   a. Part number.
   b. Type number. (For integrated circuits, this refers only to the numbers; the letters may be different or missing.)
   c. Part number and type number.
   d. Part number with a type number other than the one listed.

2. Some of the IC's may be packed in conductive foam. Do not remove the IC's from the foam until you are instructed to do so.

<table>
<thead>
<tr>
<th>No.</th>
<th>Part No.</th>
<th>QTY.</th>
<th>Description</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>442-30</td>
<td>1</td>
<td>µA309K</td>
<td>IC31</td>
</tr>
<tr>
<td>D2</td>
<td>442-644</td>
<td>1</td>
<td>LM78L12</td>
<td>IC29</td>
</tr>
<tr>
<td>D2</td>
<td>442-646</td>
<td>1</td>
<td>MC79L12AC</td>
<td>IC30</td>
</tr>
<tr>
<td>D3</td>
<td>442-616</td>
<td>1</td>
<td>LM3302N, LM2901N</td>
<td>IC18</td>
</tr>
<tr>
<td>D3</td>
<td>443-717</td>
<td>1</td>
<td>74126N</td>
<td>IC4</td>
</tr>
<tr>
<td>D3</td>
<td>443-726</td>
<td>2</td>
<td>74S00</td>
<td>IC5, IC21</td>
</tr>
<tr>
<td>D3</td>
<td>443-838</td>
<td>2</td>
<td>74LS243</td>
<td>IC9, IC10</td>
</tr>
<tr>
<td>D4</td>
<td>443-720</td>
<td>1</td>
<td>40097</td>
<td>IC13</td>
</tr>
<tr>
<td>D4</td>
<td>443-721</td>
<td>2</td>
<td>2112-2</td>
<td>IC14 through IC17</td>
</tr>
<tr>
<td>D4</td>
<td>443-804</td>
<td>6</td>
<td>74LS259</td>
<td>IC23 through IC28</td>
</tr>
<tr>
<td>D4</td>
<td>443-807</td>
<td>4</td>
<td>74LS42</td>
<td>IC2, IC3, IC20, IC22</td>
</tr>
<tr>
<td>D4</td>
<td>443-840</td>
<td>1</td>
<td>MC8875</td>
<td>IC19</td>
</tr>
<tr>
<td>D5</td>
<td>443-824</td>
<td>4</td>
<td>74LS241</td>
<td>IC1, IC6, IC7, IC8</td>
</tr>
<tr>
<td>D6</td>
<td>444-17</td>
<td>1</td>
<td>MCM6830A</td>
<td>IC12</td>
</tr>
<tr>
<td>D7</td>
<td>443-827</td>
<td>1</td>
<td>MC6800P</td>
<td>IC11</td>
</tr>
</tbody>
</table>

### SWITCHES — INSULATORS

<table>
<thead>
<tr>
<th>No.</th>
<th>Part No.</th>
<th>QTY.</th>
<th>Description</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>60-34</td>
<td>1</td>
<td>Rocker switch</td>
<td>SW1</td>
</tr>
<tr>
<td>E2</td>
<td>60-621</td>
<td>1</td>
<td>Switch assembly (May be slide or rocker switches.)</td>
<td></td>
</tr>
<tr>
<td>E3</td>
<td>64-839</td>
<td>17</td>
<td>Pushbutton switch (May look different than one shown.)</td>
<td></td>
</tr>
<tr>
<td>E4</td>
<td>73-4</td>
<td>1</td>
<td>Rubber grommet</td>
<td></td>
</tr>
<tr>
<td>E5</td>
<td>75-724</td>
<td>1</td>
<td>Insulator plate</td>
<td></td>
</tr>
<tr>
<td>E6</td>
<td>75-788</td>
<td>2</td>
<td>Insulating paper</td>
<td></td>
</tr>
</tbody>
</table>

### HARDWARE

**NOTE:** The hardware may be in more than one packet. Open all the hardware packets according to their size before you check the hardware.

Hardware is shown actual size. To identify a piece of hardware, place it over the illustration.

<table>
<thead>
<tr>
<th>No.</th>
<th>Part No.</th>
<th>QTY.</th>
<th>Description</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>250-163</td>
<td>3</td>
<td>#4 x 5/16&quot; self-tapping screw</td>
<td></td>
</tr>
<tr>
<td>F2</td>
<td>250-138</td>
<td>2</td>
<td>6-32 x 3/16&quot; screw</td>
<td></td>
</tr>
<tr>
<td>F3</td>
<td>250-56</td>
<td>13</td>
<td>6-32 x 1/4&quot; screw</td>
<td></td>
</tr>
<tr>
<td>F4</td>
<td>250-475</td>
<td>10</td>
<td>#6 x 3/8&quot; hex head screw</td>
<td></td>
</tr>
<tr>
<td>F5</td>
<td>250-32</td>
<td>1</td>
<td>6-32 x 3/8&quot; flat head screw</td>
<td></td>
</tr>
<tr>
<td>F6</td>
<td>250-162</td>
<td>2</td>
<td>6-32 x 1/2&quot; screw</td>
<td></td>
</tr>
<tr>
<td>F7</td>
<td>250-559</td>
<td>8</td>
<td>#6 x 5/8&quot; self-tapping screw</td>
<td></td>
</tr>
<tr>
<td>F8</td>
<td>250-1137</td>
<td>2</td>
<td>#6 x 1-1/8&quot; self-tapping screw</td>
<td></td>
</tr>
<tr>
<td>F9</td>
<td>252-3</td>
<td>6</td>
<td>6-32 nut</td>
<td></td>
</tr>
<tr>
<td>F10</td>
<td>254-1</td>
<td>11</td>
<td>#6 lockwasher</td>
<td></td>
</tr>
<tr>
<td>F11</td>
<td>255-23</td>
<td>4</td>
<td>Spacer</td>
<td></td>
</tr>
<tr>
<td>F12</td>
<td>259-1</td>
<td>3</td>
<td>#6 solder lug</td>
<td></td>
</tr>
<tr>
<td>F13</td>
<td>259-22</td>
<td>1</td>
<td>Spade lug</td>
<td></td>
</tr>
<tr>
<td>F14</td>
<td>260-56</td>
<td>2</td>
<td>Fuse clip</td>
<td></td>
</tr>
</tbody>
</table>

### WIRE — BRAID — LINE CORD

<table>
<thead>
<tr>
<th>No.</th>
<th>Part No.</th>
<th>QTY.</th>
<th>Description</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>344-51</td>
<td>18&quot; Brown wire</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>344-52</td>
<td>3&quot; Red wire</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>344-53 *</td>
<td>9&quot; Orange wire</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>344-54</td>
<td>20&quot; Yellow wire</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>344-71</td>
<td>18&quot; White-brown wire</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>344-74</td>
<td>9&quot; White-yellow wire</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>344-73</td>
<td>9&quot; White-orange wire</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>344-99</td>
<td>18&quot; White stranded wire</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>345-1</td>
<td>3&quot; Flat braid</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89-49</td>
<td>1 Line cord</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>346-1</td>
<td>2&quot; Sleeving</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### TERMINAL STRIPS — CONNECTORS — SOCKETS

<table>
<thead>
<tr>
<th>KEY</th>
<th>HEATH No.</th>
<th>QTY.</th>
<th>DESCRIPTION</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>431-2</td>
<td>1</td>
<td>2-lug terminal strip</td>
<td></td>
</tr>
<tr>
<td>G2</td>
<td>431-86</td>
<td>1</td>
<td>6-lug terminal strip</td>
<td></td>
</tr>
<tr>
<td>G3</td>
<td>432-874</td>
<td>13</td>
<td>4-pin connector block</td>
<td></td>
</tr>
<tr>
<td>G4</td>
<td>432-973</td>
<td>11</td>
<td>8-pin connector block</td>
<td></td>
</tr>
<tr>
<td>G5</td>
<td>432-875</td>
<td>1</td>
<td>Large connector block</td>
<td></td>
</tr>
<tr>
<td>G6</td>
<td>432-921</td>
<td>2</td>
<td>3-pin IC socket</td>
<td></td>
</tr>
<tr>
<td>G7</td>
<td>434-336</td>
<td>1</td>
<td>TO-3 socket</td>
<td></td>
</tr>
<tr>
<td>G8</td>
<td>434-298</td>
<td>12</td>
<td>14-pin IC socket</td>
<td></td>
</tr>
<tr>
<td>G9</td>
<td>434-299</td>
<td>16</td>
<td>16-pin IC socket</td>
<td></td>
</tr>
<tr>
<td>G10</td>
<td>434-311</td>
<td>4</td>
<td>20-pin IC socket</td>
<td></td>
</tr>
<tr>
<td>G11</td>
<td>434-307</td>
<td>1</td>
<td>24-pin IC socket</td>
<td></td>
</tr>
<tr>
<td>G12</td>
<td>434-253</td>
<td>1</td>
<td>40-pin IC socket</td>
<td></td>
</tr>
</tbody>
</table>

### CIRCUIT BOARDS — CABINET — BRACKET

<table>
<thead>
<tr>
<th>KEY</th>
<th>HEATH No.</th>
<th>QTY.</th>
<th>DESCRIPTION</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>85-2033-3</td>
<td>1</td>
<td>Main circuit board</td>
<td></td>
<td></td>
</tr>
<tr>
<td>85-2010-1</td>
<td>1</td>
<td>Keyboard circuit board</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H1</td>
<td>92-611</td>
<td>1</td>
<td>Cabinet top</td>
<td></td>
</tr>
<tr>
<td>H2</td>
<td>92-612</td>
<td>1</td>
<td>Cabinet bottom</td>
<td></td>
</tr>
<tr>
<td>H3</td>
<td>204-2291</td>
<td>1</td>
<td>Support bracket</td>
<td></td>
</tr>
</tbody>
</table>

### LIGHT-EMITTING DIODES (LED’s) — FUSE

<table>
<thead>
<tr>
<th>KEY</th>
<th>HEATH No.</th>
<th>QTY.</th>
<th>DESCRIPTION</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>411-831</td>
<td>6</td>
<td>7-segment LED</td>
<td>H, I, N, Z, V, C</td>
</tr>
<tr>
<td>J2</td>
<td>412-640</td>
<td>1</td>
<td>3/8&quot; red LED</td>
<td>LED1</td>
</tr>
</tbody>
</table>

### Light-Emitting Diodes (Led’s) — Fuse (cont’d.)

<table>
<thead>
<tr>
<th>KEY</th>
<th>HEATH No.</th>
<th>QTY.</th>
<th>DESCRIPTION</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>J3</td>
<td>412-616</td>
<td>8</td>
<td>1/4&quot; red LED</td>
<td>LED2 through</td>
</tr>
<tr>
<td>J4</td>
<td>421-42</td>
<td>1</td>
<td>3/8-ampere, 3AG, slow-blow fuse</td>
<td>F1</td>
</tr>
</tbody>
</table>

### MISCELLANEOUS

<table>
<thead>
<tr>
<th>KEY</th>
<th>HEATH No.</th>
<th>QTY.</th>
<th>DESCRIPTION</th>
<th>CIRCUIT Comp. No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td>54-920</td>
<td>1</td>
<td>Power transformer</td>
<td>T1</td>
</tr>
<tr>
<td>K2</td>
<td>260-700</td>
<td>1</td>
<td>LED grommet</td>
<td></td>
</tr>
<tr>
<td>K3</td>
<td>261-34</td>
<td>4</td>
<td>Foot</td>
<td></td>
</tr>
<tr>
<td>K4</td>
<td>352-13</td>
<td>1</td>
<td>Silicone grease</td>
<td></td>
</tr>
<tr>
<td>K5</td>
<td>354-7</td>
<td>1</td>
<td>Cable tie</td>
<td></td>
</tr>
<tr>
<td>K6</td>
<td>262-8</td>
<td>2</td>
<td>Terminal pin</td>
<td></td>
</tr>
<tr>
<td>K7</td>
<td>475-12</td>
<td>1</td>
<td>Ferrite bead</td>
<td></td>
</tr>
<tr>
<td>K8</td>
<td>462-1023</td>
<td>7</td>
<td>Square knob</td>
<td></td>
</tr>
<tr>
<td>K9</td>
<td>490-111</td>
<td>1</td>
<td>IC puller</td>
<td></td>
</tr>
</tbody>
</table>

### PRINTED MATERIAL

<table>
<thead>
<tr>
<th>KEY</th>
<th>HEATH No.</th>
<th>QTY.</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>390-1255</td>
<td>1</td>
<td>Fuse label</td>
</tr>
<tr>
<td>L2</td>
<td>390-1390</td>
<td>1</td>
<td>Power label</td>
</tr>
<tr>
<td>L3</td>
<td>390-1391</td>
<td>1</td>
<td>&quot;Heathkit&quot; label</td>
</tr>
<tr>
<td>L4</td>
<td>390-1395</td>
<td>1</td>
<td>Keyboard label set</td>
</tr>
<tr>
<td></td>
<td>390-1404</td>
<td>1</td>
<td>Red label set</td>
</tr>
<tr>
<td>L5</td>
<td>597-260</td>
<td>1</td>
<td>Blue and white label</td>
</tr>
<tr>
<td></td>
<td>1 Parts Order Form</td>
<td>1</td>
<td>Assembly Manual (See Page 1 for part number.)</td>
</tr>
</tbody>
</table>
**STEP-BY-STEP ASSEMBLY**

The steps performed in this Pictorial are
in this area of the circuit board.

---

**MAIN CIRCUIT BOARD**

**CONTINUE**

( ) Solder the resistor leads to the
circuit board as follows:

1. Push the soldering iron
tip against both the lead
and the circuit board foil.
Heat both for two or three
seconds.

2. Then apply solder to the
other side of the connection.
IMPORTANT: Let
the heated lead and the
circuit board foil melt the
solder.

3. As the solder begins to
melt, allow it to flow
around the connection.
Then remove the solder
and the iron and let the
connection cool.

( ) Hold the lead with one hand
while you cut off the excess lead
length close to the connection.
This will keep you from being hit
in the eye by the flying lead.

( ) Check the connection. Compare it
to the illustrations on the next
page. After you have checked the
solder connections, proceed with
the assembly on page 12. Use the
same soldering procedure for
each connection.

---

**PICTORIAL 1-1**

---

START

In the following steps you will be given
detailed instructions on how to install
and solder the first part on the circuit
board. Read and perform each step
carefully. Then use the same proce-
dure whenever you install parts on a
circuit board.

( ) Position the circuit board as
shown in the identification draw-
ing with the printed side up.

( ) R49: Hold a 1200 Ω (brown-red-
red) resistor by the body as shown
and bend the leads straight down.

( ) Push the leads through the holes
at the indicated location on the
circuit board. The end with
color bands may be positioned
either way.

( ) Press the resistor against the cir-
cuit board. Then bend the leads
outward slightly to hold the re-
sister in place.
A GOOD SOLDER CONNECTION

When you heat the lead and the circuit board foil at the same time, the solder will flow evenly onto the lead and the foil. The solder will make a good electrical connection between the lead and the foil.

POOR SOLDER CONNECTIONS

When the lead is not heated sufficiently, the solder will not flow onto the lead as shown above. To correct, reheat the connection and, if necessary, apply a small amount of additional solder to obtain a good connection.

When the foil is not heated sufficiently the solder will blob on the circuit board as shown above. To correct, reheat the connection and, if necessary, apply a small amount of additional solder to obtain a good connection.

SOLDER BRIDGES

A solder bridge between two adjacent foils is shown in photograph A. Photograph B shows how the connection should appear. A solder bridge may occur if you accidentally touch an adjacent previously soldered connection, if you use too much solder, or if you “drag” the soldering iron across other foils as you remove it from the connection. A good rule to follow is: always take a good look at the foil area around each lead before you solder it. Then, when you solder the connection, make sure the solder remains in this area and does not bridge to another foil. This is especially important when the foils are small and close together. NOTE: It is alright for solder to bridge two connections on the same foil.

Use only enough solder to make a good connection, and lift the soldering iron straight up from the circuit board. If a solder bridge should develop, turn the circuit board foil-side-down and heat the solder between connections. The excess solder will run onto the tip of the soldering iron, and this will remove the solder bridge. NOTE: The foil side of most circuit boards has a coating on it called “solder resist.” This is a protective insulation to help prevent solder bridges.
MAIN CIRCUIT BOARD

NOTE: When you install an IC socket, use the following procedure:

1. Be sure all the pins are straight.
2. Insert the pins into the holes.
3. Turn the circuit board over and be sure the correct number of pins extend from the board. If not, one or more pins may be bent under the socket. Remove the socket, straighten the pins, and reinstall the socket.
4. Solder the pins to the foil as you install each socket. NOTE: Some socket pins will have no foil pads; do not solder these pins.

16-pin IC sockets at the seven following locations:

( ) IC14. ( ) IC19.
( ) IC15. ( ) IC20.
( ) IC16. ( ) IC22.
( ) IC17.

NOTE: Be sure you install the first resistor (Page 10).

The steps performed in this Pictorial are in this area of the circuit board.

START

CONTINUE

( ) R10: 8200 Ω (gray-red-red).
( ) R7: 220 kΩ (red-red-yellow).
( ) R9: 150 kΩ (brown-green-yellow).
( ) R2: 150 Ω (brown-green-brown).
( ) R14: 100 kΩ (brown-black-yellow).
( ) R12: 100 kΩ (brown-black-yellow).
( ) R13: 820 kΩ (gray-red-yellow).
( ) R11: 100 kΩ (brown-black-yellow).
( ) R15: 8200 Ω (gray-red-red).
( ) R5: 8200 Ω (gray-red-red).
( ) R6: 6800 Ω (blue-gray-red).
( ) 14-pin IC socket at IC18.
( ) R50: 220 kΩ (red-red-yellow).
( ) R1: 27 kΩ (red-violet-orange).
( ) R8: 15 kΩ (brown-green-orange).
( ) R3: 68 Ω, 1/2-watt (blue-gray-black).
( ) R4: 68 Ω, 1/2-watt (blue-gray-black).

Solder the leads to the foil and cut off the excess lead lengths.

( ) 14-pin IC socket at IC21.
START

1. Install thirty 470 Ω (yellow-violet-brown) resistors in the area shown. After you install each group of five or six resistors, solder their leads to the foil and cut off the excess lead lengths. NOTE: See "Circuit Board X-Ray Views" in the "Illustration Booklet" for circuit component numbers.

In the next column you will install diodes. Be sure you install each diode as follows.

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

CAUTION: ALWAYS POSITION THE BANDED END AS SHOWN ON THE CIRCUIT BOARD.

If your diode has a solid body, the band is clearly defined. If your diode has a glass body, do not mistake the colored end inside the diode for the banded end. Look for a band painted on the outside of the glass.

NOTE: Hold the leads with a pair of long-nosed pliers close to the body of the diode. Then bend the leads down.

NOTE: As you install the remaining components on this Pictorial, solder the leads to the foil and cut off the excess lead lengths.

- **D10**: 1N4149 diode (#56-56).
- **R63**: 470 Ω (yellow-violet-brown).
- **D5**: 1N4002 diode (#57-65).
- **R71**: 470 Ω (yellow-violet-brown).
- **D6**: 1N4002 diode (#57-65).
- **D1**: 16-pin IC socket at IC23.
- **D4**: 1N4002 diode (#57-65).
- **R79**: 470 Ω (yellow-violet-brown).
- **D3**: 1N4002 diode (#57-65).
- **R87**: 470 Ω (yellow-violet-brown).
- **D1**: 16-pin IC socket at IC24.
- **R95**: 470 Ω (yellow-violet-brown).
- **D1**: 3A1 diode (#57-42). See Detail 1-3A.
- **R106**: 470 Ω (yellow-violet-red).
- **D2**: 3A1 diode (#57-42). See Detail 1-3A.
- **R104**: 470 Ω (yellow-violet-brown).
START

14-pin IC sockets at the six following locations:

1) H, 2) Z.
3) L, 4) V.
5) N, 6) G.

7) 24-pin IC socket at IC12.

NOTE: When you install the diodes in the next three steps, be sure to position the banded ends as shown.

CAUTION: ALWAYS POSITION THE BANDED END AS SHOWN ON THE CIRCUIT BOARD.

If your diode has a solid body, the band is clearly defined. If your diode has a glass body, do not mistake the colored end inside the diode for the banded end. Look for a band painted on the outside of the glass.

8) D7: 1N4149 diode (#56-56).
9) D8: 1N4149 diode (#56-56).
10) D9: 1N4149 diode (#56-56).
11) Solder the leads to the foil and cut off the excess lead lengths.
12) 16-pin IC socket at IC13.

CONTINUE

Install twelve 470 Ω (yellow-violet-brown) resistors.

1) R62.
2) R61.
3) R70.
4) R69.
5) R78.
6) R77.
7) R86.
8) R85.
9) R94.
10) R93.
11) R102.
12) R101.

1) Solder the leads to the foil and cut off the excess lead lengths.

NOTE: When you install a terminal pin, push the pin as far as possible into the circuit board hole. Then solder the pin to the foil and cut off the excess pin length.

1) Two terminal pins at "SEGMENT TEST."
The steps performed in this Pictorial are in this area of the circuit board.

START

( ) R51: 8200 Ω (gray-red-red).
( ) R42: 27 kΩ (red-violet-orange).
( ) R43: 8200 Ω (gray-red-red).
( ) R45: 8200 Ω (gray-red-red).

CONTINUE

( ) 14-pin IC socket at IC9.
( ) 14-pin IC socket at IC10.
( ) 14-pin IC socket at IC4.
( ) 20-pin IC socket at IC6.
( ) 40-pin IC socket at IC11.

( ) Solder the leads to the foil and cut off the excess lead lengths.
( ) 20-pin IC socket at IC7. Be sure all the pins protrude through the board before you solder any of them.
( ) 20-pin IC socket at IC8.
The steps performed in this Pictorial are in this area of the circuit board.

**CONTINUE**

Install eight 8200Ω (gray-red-red) resistors.

( ) R24.
( ) R25.
( ) R26.
( ) R27.
( ) R28.
( ) R29.
( ) R30.
( ) R31.

( ) Solder the leads to the foil and cut off the excess lead lengths.

Install three 8200Ω (gray-red-red) resistors.

( ) R44.
( ) R41.
( ) R47.

( ) Solder the leads to the foil and cut off the excess lead lengths.

( ) 1" bare wire. Remove 1" of insulation from the yellow wire. Then cut off the bare wire, install it, solder its ends to the foil, and cut off the excess wire ends.

**PICTORIAL 1-6**
NOTE: In the next step, be sure to position the switch assembly as shown. (It may have slide or rocker switches.)

1. Switch assembly (#60-621).
6. C5: .01 μF ceramic. See Detail 1-7A.
7. C20: .01 μF ceramic. See Detail 1-7A.
8. Solder the leads to the foil and cut off the excess lead lengths.

Install connector blocks in the following manner:

- **A.** 4-Pin Connector Block
  - Be sure the metal tab is straight. Then install the block.
  
- **B.**
  - Solder the metal tab to the foil.

- **C.**
  - Flatten the end posts with your soldering iron tip.

The steps performed in this Pictorial are in this area of the circuit board.

**CONTINUE**

- Install two 8-pin connector blocks.
- Install eight 4-pin connector blocks.
- Install four 4-pin connector blocks.
- Install six 8-pin connector blocks.

Position the capacitor down flat toward the edge of the circuit board as shown.

**Circuit Board Edge**

Detail 1-7A

**PICTORIAL 1-7**
START

IMPORTANT: As you install LED’s in the following step, be sure to match the flat on each LED with the outline of the flat on the circuit board as shown.

1) LED2 through LED9: Install eight 1/4" red LED’s in the shaded area. Solder the leads to the foil and cut off the excess lead lengths.

2) R32 through R39: Install eight 180 Ω (brown-gray-brown) resistors. Solder the leads to the foil and cut off the excess lead lengths.

3) Locate an 8-pin connector block. Then refer to Detail 1-8A below and check all four contacts on the bottom of the block. If you find any burrs or raised edges, press them down with a screwdriver blade or similar tool. This will prevent them from causing a short circuit on the circuit board. NOTE: Make sure you use this connector block in the next step.

CHECK FOR BURRS OR RAISED EDGES HERE.

Detail 1-8A

PICTORIAL 1-8
The steps performed in this Pictorial are in these areas of the circuit board.

NOTE: When you install a tantalum capacitor, install the lead marked with the positive (+) mark or color dot on the capacitor in the positive (+) marked hole on the board.

- C13: .68 μF tantalum.
- C11: 10 μF (10M) tantalum.
- C8: 2.2 μF tantalum.
- C12: 10 μF (10M) tantalum.
- C9: 2.2 μF tantalum.
- Solder the leads to the foil and cut off the excess lead lengths.

START

- C14: .01 μF ceramic.
- C23: 100 pF mica.
- C24: .01 μF ceramic.
- C15: .01 μF ceramic.
- C4: .01 μF ceramic.
- C16: .01 μF ceramic.

SOLDER the leads to the foil and cut off the excess lead lengths.
START

In the following steps, install IC's in the designated sockets. Be careful to match the pin 1 end of each IC to the index mark on the circuit board. See Detail 1-10A.

Before you apply downward pressure to an IC, make sure each IC pin is centered in its proper socket hole. Handle IC's with care, as their pins bend very easily.

NOTE: An IC puller has been furnished to remove an IC from its socket if necessary.

Push the shorter end of the puller in between the IC and the socket and rock the longer portion back and forth. Be very careful, as the IC pins are very easily bent.

- IC1: 74LS241 (#443-824).
- IC4: 74126N (#443-717).
- IC2: 74LS42 (#443-807). Be sure to notice the index mark on the circuit board.
- IC5: 74S00 (#443-26).
- IC7: 74LS241 (#443-824).
- IC8: 74LS241 (#443-824).
- IC3: 74LS42 (#443-807).

The steps performed in this Pictorial are in this area of the circuit board.

PICTORIAL 1-10

Detail 1-10A.
The steps performed in this Pictorial are in this area of the circuit board.

START

( ) IC9: 74LS243 (#443-839).
( ) IC10: 74LS243 (#443-839).
( ) IC18: LM3302N or LM2901N (#442-616).
( ) IC19: MC6875 (#443-840).
( ) IC20: 74LS42 (#443-807).
( ) IC21: 74S00 (#443-26).
( ) IC22: 74LS42 (#443-807).
( ) IC23: 74LS259 (#443-804).
( ) IC24: 74LS259 (#443-804).
( ) IC25: 74LS259 (#443-804).
( ) IC26: 74LS259 (#443-804).
( ) IC27: 74LS259 (#443-804).
( ) IC28: 74LS259 (#443-804).

NOTE: The remaining integrated circuits will be installed later. IC’s 16 and 17 are supplied with the educational course.

PICTORIAL 1-11
Refer to Pictorial 1-12 (Illustration Booklet, Page 3) for the following steps.

- Reposition the main circuit board as shown.

- SW1: Refer to Detail 1-12A and mount the rocker switch on the main circuit board at SW1 with two 6-32 x 3/16" screws. Install the switch so the lugs are positioned as shown in the Detail.

Detail 1-12B

- Refer to Part A of Detail 1-12B and install the connector strips (supplied with the large connector block) into the block in the manner shown. NOTE: You may have some connector strips left over.

- Turn the connector block right side up, and with a screwdriver handle or similar tool, tap on the top of the block until all the connector strips are fully seated up into the block.

- Refer to Part B of Detail 1-12B and remove the paper backing from the vinyl strip supplied with the connector block. Position the connector as shown, line up the long edge of the vinyl strip with the long edge of the connector block, and firmly press the strip onto the block.

- Refer to Detail 1-12B and remove the backing paper from the insulating paper. Then apply the insulating paper along the indicated edges of the vinyl strip. Keep the paper even with the edges of the large connector block.
CIRCUIT BOARD CHECKOUT

Carefully inspect the foil side of the circuit board for the following conditions.

( ) Unsoldered connections.

( ) Poor solder connections.

( ) Solder bridges between foil patterns. NOTE: If you are in doubt about a foil pattern, refer to the "Circuit Board X-Ray View" (Illustration Booklet, Page 18).

( ) Protruding leads which could touch together.

Carefully inspect the component side of the circuit board for the following conditions.

( ) Integrated circuits for proper type and installation.

( ) Tantalum capacitors for the correct position of the positive (+) mark or dot.

( ) Diodes for the correct position of the banded ends.

( ) LED's for the correct position of the flat sides.

NOTE: There are many unused connections on the foil side of the main circuit board, some of which will be used later. As you make further connections to the circuit board, be sure to inspect each one carefully to be sure the foils remain unbridged.

Set the main circuit board aside temporarily.

DETAIL 1-12E

3/8" RED LED
(#412-640)

0 1/8 3/8 1/2 1 (INCHES) 2 3 4 5 6 7
0 5 1 (CM) 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
KEYBOARD CIRCUIT BOARD

START

Position the keyboard circuit board as shown. Then proceed with the following steps.

NOTE: To prepare a wire, as in the following step, cut it to the indicated length and remove 1/4" of insulation from each end. If the wire is stranded, tightly twist each wire end and apply a small amount of solder to hold the fine strands together.

() Prepare the following wires:
- One 2-1/2" white stranded
- Nine 1-1/2" white stranded
- One 1-3/8" yellow

As you install a prepared wire in the following steps, solder it to the foil and cut off the excess wire length.

() 2-1/2" white wire at A.

() Nine 1-1/2" white wires at B through K.

() R52 through R57: Install six 8200 Ω (gray-red-red) resistors. Solder the leads to the foil and cut off the excess lead lengths.

() 1-3/8" yellow wire jumper.

() Remove the insulation from 1" of brown wire. Then cut off this bare wire.

() 1" bare wire at L.

() R107: 180 Ω (brown-gray-brown). Mount it vertically down on the circuit board, solder the lead to the foil, and cut off the excess lead length. The free lead will be connected later.

CONTINUE

As you install pushbutton switches in the following step, be sure each key is down against the top of the keyboard before you solder its two lugs. Your switches may look different than the one shown.

() 17 pushbutton switches.

CIRCUIT BOARD CHECKOUT

Carefully inspect the circuit board for the following conditions.

() Unsoldered connections.

() "Cold" solder connections.

() Solder bridges between foil patterns.

() Protruding leads which could touch together.
Refer to Pictorial 3-1 (Illustration Booklet, Page 3) for the following steps.

Refer to Detail 3-1A, turn the main circuit board upside down, and loosely mount spacers onto the foil side at the four locations shown in the Pictorial. Use 6-32 x 1/4" screws and #6 lockwashers.
Connect the wires coming from the keyboard circuit board to the main circuit board:

- Wire B to B.
- Wire K to K.
- Wire J to J.
- Wire H to H.
- Wire G to G.
- Wire F to F.
- Wire E to E.
- Wire A to A.
- Wire D to D.
- Wire C to C.

- Flip the keyboard circuit board over, end-for-end, (keep the wires out of the way) and position the tops of the pushbutton switches into their corresponding holes in the main circuit board. (If your switches have springs and brass washers, you may have to force them through the holes).

- Connect the wire coming from keyboard hole L to hole M on the main circuit board. Do not solder the connection.

- Connect the resistor coming from keyboard hole M to hole L on the main circuit board. Do not solder the connection.

- Loosely install four 6-32 x 1/4" screws and #6 lockwashers at the keyboard corner holes. Turn the screws into the spacers as shown.

- On the top of the main circuit board, tighten the four 6-32 screws to secure the spacers; then tighten the four keyboard mounting screws.

- Solder the wire and resistor lead to the main circuit board at L and M and cut off the excess lengths.
**SUPPORT BRACKET ASSEMBLY**

Refer to Pictorial 4-1 (Illustration Booklet, Page 4) for the following steps.

1. Position the support bracket on your work area as shown.

2. Refer to Detail 4-1A and mount a solder lug at A with a 6-32 x 1/4" screw and a 6-32 nut. Position the solder lug as shown in the Pictorial.

3. Press a rubber grommet into hole B.

4. Refer to Detail 4-1B and mount a 2-lug terminal strip at C. Use a 6-32 x 1/4" screw, #6 lockwasher, #6 solder lug, and 6-32 nut. Position the terminal strip and solder lug as shown in the Pictorial.

5. Cut the lead at the positive (+) end of a 1200 µF electrolytic capacitor (#25-241) to 1/2".

NOTE: In the following steps, (NS) means not to solder a connection because other wires or leads will be connected later. “S-” with a number, such as (S-2), means to solder the connection. The number following the “S” tells how many wires are at the connection.

6. C6: Connect the **positive** (+) lead of a 1200 µF capacitor to terminal strip C lug 1 (NS) and the negative lead to solder lug A (NS). Position the capacitor as shown in the Pictorial.

7. Cut the lead at the negative (unmarked) end of another 1200 µF electrolytic capacitor to 1/2".

8. C7: Connect the **negative** (unmarked) lead of the other 1200 µF capacitor to terminal strip C lug 2 (NS) and the positive (+) lead to solder lug A (S-2).

Refer to Detail 4-1C for the next two steps.

9. Refer to the inset drawing on Detail 4-1C and open the container of silicone grease. Apply a liberal coating of the grease to the bottom of the µA309K integrated circuit (#442-30).

10. IC31: Carefully observe the wide spacing on the IC holes in the support bracket at IC31 and place the TO-3 socket on the underside of the bracket as shown in the Detail. Be sure the shoulders of the socket are centered in the two end holes. Then push the pins of the µA309K IC into the socket, through the support bracket. Making sure the socket shoulders are still centered in their holes, secure the IC with two 6-32 x 1/2" screws.
( ) Cut a 2" piece of flat braid.

( ) Refer to Detail 4-2A and crimp and solder a spade lug onto one end of the 2" braid. Apply a liberal amount of solder to 1/4" of the free braid end.

( ) Loosely mount the support bracket to the foil side of the main circuit board at E with a 6-32 x 1/4" screw and a 6-32 nut. Secure the support bracket and the spade lug with the braid at F with a 6-32 x 1/4" screw and 6-32 nut. Be sure to position the free end of the braid as shown in the Pictorial. Tighten the support bracket mounting hardware.

( ) Refer to the Pictorial and form the center of the braid and the spade lug as shown to be sure the braid will not come in contact with any of the other circuit board foils.

( ) Refer to Detail 4-2B and add a liberal amount of solder to the indicated foil pad on the main circuit board. Be very careful not to form a bridge to other foils. Press the free end of the braid onto the top of this foil pad and heat it with the soldering iron until the solder melts into the braid. Hold the braid in place with pliers until it has cooled.

---

---
Refer to Detail 4-2C and pass the tip of the cable tie through hole G in the support bracket making sure the rough side is facing upward. Then pass the tie across the rear of the bracket and back through hole H making sure the rough side is down. Pass the cable tie around capacitor C1 and push the tip of the tie through the other end retainer as shown. Pull the tie until it is tightly secure around the capacitor; then cut off the excess tie end.

Refer to Pictorial 4-3 (Illustration Booklet, Page 5) for the following steps.

Reposition the circuit board as shown.

Prepare the following wires:

1-1/4" red  6-1/2" yellow  
2-1/2" white-brown 6-1/4" white-yellow  
2-1/2" brown  13" brown  
1-3/4" orange  13" white-brown

NOTE: As you install wires in the following steps, form each of them as shown in the Pictorial. After a wire has been soldered to the foil or to the switch lug, cut off any excess wire lengths.

Connect one end of a 1-1/4" red wire to switch SW1 lug 1 (S-1). Slide a ferrite bead (#475-12) onto the free end of this wire; then connect the free end to the main circuit board hole R (S-1).

Connect a 2-1/2" white-brown wire from hole S (S-1) to switch SW1 lug 7 (S-1).

Connect a 2-1/2" brown wire from hole T (S-1) to switch SW1 lug 4 (S-1).

Connect a 1-3/4" orange wire from hole U (S-1) to switch SW1 lug 2 (NS). Be sure this wire does not cover the large nearby hole.

Form the orange wire coming from socket IC31 lug 2 downward and across the circuit board as shown. Connect the free end of the wire to SW1 lug 2 (S-2).

Route the free end of the white-orange wire coming from socket IC31 lug 1 downward to the board, and along the board as shown. Connect the free end of the wire to circuit board hole X (S-1).

Connect one end of a 6-1/2" yellow wire to circuit board hole W (S-1). Route the wire rearward, through support bracket grommet B. Connect the wire end to terminal strip C lug 1 (S-2).

Connect one end of a 6-1/4" white-yellow wire to circuit board hole V (S-1). Route the wire rearward, through support bracket grommet B. Connect the wire end to terminal strip C lug 2 (S-2).

Connect one end of a 13" brown wire to circuit board hole P (S-1). Route the wire forward, through support bracket grommet B. Connect the free end of the wire to switch SW1 lug 5 (S-1).

Connect one end of a 13" white-brown wire to circuit board hole N (S-1). Route the wire forward and through grommet B. Connect the free end of the wire to switch SW1 lug 8 (S-1).

Set the main circuit board assembly aside temporarily.
CABINET ASSEMBLY AND WIRING.

Refer to Pictorial 5-1 (Illustration Booklet, Page 5) for the following steps.

( ) Temporarily mount a 6-lug terminal strip on cabinet post AA with a #6 × 3/8" hex head screw as shown.

( ) F1: Refer to Detail 5-1A and install two fuse clips and the 3/8-ampere fuse on terminal strip AA lugs 2 and 4. Solder the fuse clip onto lug 4 only. NOTE: Do not use excessive heat to avoid damage to the fuse.

( ) Refer to inset drawing #1 on Detail 5-1B and insert the end of the line cord through hole AB from the outside of the cabinet bottom. Tie a knot in the line cord 4-1/2" from the end as shown.

( ) Refer to inset drawing #2 on Detail 5-1B and identify the smooth lead and the ribbed lead of the line cord. Then prepare the end of the line cord as shown in the Detail.

( ) Tightly twist the bare wire ends and apply a small amount of solder to hold the fine strands together.

NOTE: As you connect the line cord leads in the following steps, be sure to make a mechanically secure connection. Wrap the lead ends securely under the terminal strip as shown in Detail 5-1C.

( ) Smooth lead to the eyelet of lug 4 (S-1).

( ) Ribbed lead to the eyelet of lug 6 (S-1).

( ) Refer to Detail 5-1D and prepare the transformer leads as shown. Measure the leads from the edge of the transformer. If necessary, twist the lead ends tightly and apply a small amount of solder.

( ) T1: Refer to Pictorial 5-1 and install the power transformer with the red and green leads up as shown. Use #6 × 1-1/8" self-tapping screws.
ALTERNATE LINE VOLTAGE WIRING

Two sets of line voltage wiring instructions are given below, one for 120 VAC and the other for 240 VAC. In the United States, 120 VAC is most common. USE ONLY THE INSTRUCTIONS THAT AGREE WITH THE LINE VOLTAGE IN YOUR AREA.

**FOR 120 VAC**

Refer to Detail 5-1E for the following steps. In these steps, make connections to terminal strip AA. Wrap the lead ends tightly at the connections. Connect four of the power transformer leads as follows:

1. Black-red and black-yellow leads to lug 2 (S-2). NOTE: Also solder the fuse clip to lug 2.
2. Black-green and black leads to lug 6 (S-2).

**FOR 240 VAC**

Refer to Detail 5-1F for the following steps. In these steps, make connections to terminal strip AA. Wrap the lead ends tightly at the connection. Connect four of the power transformer leads as follows:

1. Black-red lead to lug 2 (S-1). NOTE: Also solder the fuse clip to lug 2.
2. Black-yellow and black-green leads to lug 5 (S-2).
3. Black lead to lug 6 (S-1).
PICTORIAL 5-2

Refer to Pictorial 5-2 for the following steps.

( ) Remove the fuse from the fuse clips. Then remove the screw you used to secure the terminal strip to the cabinet post.

( ) Refer to Detail 5-2A and mount the terminal strip in the box formed in the cabinet bottom as shown. Use a 6-32 x 3/8" flat head screw, two #6 lockwashers, and a 6-32 nut. Position the terminal strip as shown in Detail 5-2B.

( ) Reinstall fuse F1 in its fuse clips.
Again, refer to Pictorial 5-1 (Illustration Booklet, Page 5) for the following steps.

NOTE: As you connect each of the remaining power transformer wires to the main circuit board, in the following steps, solder the lead to the foil and cut off the excess lead lengths.

( ) Position the main circuit board, component side up, near the power transformer as shown in Pictorial 5-1.

( ) Connect the red-yellow transformer lead to the circuit board hole labeled “RED/YEL.”

( ) Connect the green-yellow lead to the hole labeled “GRN/YEL.”

( ) Connect either red lead to one hole labeled “RED.”

( ) Connect the other red lead to the remaining “RED” hole.

( ) Connect one green lead to one of the holes labeled “GRN.”

( ) Connect the other green lead to the remaining “GRN” hole.

NOTE: The remaining yellow wire is for any experiments you may want to do.

This completes the “Step-by-Step Assembly.” Proceed to “Initial Tests.”

( ) Refer to Detail 5-2B and route the leads and wires as shown.

( ) Mount the insulator plate to the terminal strip box with two #6 × 3/8” hex head screws. Do not pinch any leads between the plate and the box.

( ) Remove the paper backing from the fuse label and press the label in place onto the insulator plate. Then write the fuse information on the label: “3/8-Amp, 3AG, slow-blow.”
INITIAL TESTS

IC30: In the same manner, install the MC79L12AC IC (#442-646) in the socket at IC30.

VOLTAGE TESTS

NOTE: If at any time during the following tests you fail to obtain the desired results, and if power is applied to the unit, immediately unplug the line cord from the outlet and refer to the "In Case of Difficulty" section on Page 76.

You will need a volt-ohmmeter to perform the following tests. If such a meter is not available, proceed to "Tests Continued."

Connect one ohmmeter lead to one prong of the line cord plug, and the other lead to the remaining prong. The ohmmeter reading should be near or at zero.

Push down on the left side of the POWER switch (SW1) to be sure it is Off.

Plug the line cord into an AC outlet. The red LED next to the power switch should come on immediately and will remain on, regardless of the power switch setting.

Prepare two 1-1/2" wires. These may be of any color.

Locate the 4-pin connector blocks near the lower left corner of the circuit board labeled "+5" and "GND." Push one end of a short wire into each of these blocks.

Set your voltmeter to read +5 volts. Connect the positive lead to the wire at "+5" and the negative lead to "GND."

Push down on the right side of the POWER switch (SW1).

You should read 4.5 to 5.5 volts on the voltmeter.
( ) Set the voltmeter to read +12 volts. Move the positive meter lead and the test wire from “+5” to “+12.” You should read 10.8 to 13.2 volts on the meter.

( ) Remove the voltmeter leads from the test wires; then move the test wire at “+12” to “−12.”

( ) Connect the positive test lead to “GND” and the negative lead to “−12.” You should read 10.8 to 13.2 volts on the meter.

This concludes the portion of the tests that require the use of the volt-ohmmeter. Set the meter and wires aside.

TESTS CONTINUED

( ) If not already done, plug in the line cord and push down on the right side of the POWER switch (SW1). The red LED next to the POWER switch should turn on. (This LED will be on no matter which position the switch is in.)

( ) At the right edge of the circuit board, locate the “SEGMENT TEST” pins. Short these two pins together and note that all seven segments on the 7-segment LED’s are lit, as well as the decimal point at the lower right of each LED. (Some LED’s may already be lit.)

( ) Push the POWER switch to Off and remove the line cord plug from the AC outlet.

Refer to Pictorial 6-2 (Illustration Booklet, Page 7) for the following steps.

( ) Refer to Detail 6-2A and place a square knob onto one of the pushbutton switches at the lower right portion of the circuit board. Push firmly on the knob to seat it onto the switch.

( ) In the same manner, install the remaining 16 square knobs on the pushbutton switches.

( ) Locate the keyboard label set. Then, one at a time, remove each of the numbered or lettered labels from the paper backing and press the label onto its correct pushbutton knob as shown in the Pictorial.

( ) Locate the red label set. One at a time, remove the red labels from the paper backing, then position the label squarely over the 7-segment LED and press it in place. (You should have two labels left over.)
Some of the IC's are packed in conductive foam. (Save this foam in case you ever remove these IC's.) These IC's are rugged, reliable components. However, normal static electricity discharged from your body through an IC pin to an object can damage the IC. Install these IC's without interruption as follows:

1. Remove the IC from its package with both hands.
2. Hold the IC with one hand and straighten any bent pins with the other hand.
3. Refer to Detail 6-2B. Position the pin 1 end of the IC over the index mark on the circuit board.
4. Be sure each IC pin is properly started into the socket. Then push the IC down.

- IC14: Install a 2112-2 IC (#443-721) in the socket at IC14.
- IC15: Install a 2112-IC (#443-721) in the socket at IC15.
- IC13: Install a 40097 IC (#443-720) in the socket at IC13.
- IC12: Install an MCM6830A IC (#444-17) in the socket at IC12.
IC11: Install an MC6800P IC (#443-827) in the socket at IC11.

Prepare a 4" yellow wire.

Plug in the line cord and turn the Trainer on.

Refer to Detail 6-2C for the following three steps.

NOTE: In the following steps, you will check out the Binary Data LED's at the lower left side of the circuit board. Each of these LED's is numbered (from right to left), directly beneath their corresponding 4-pin connector blocks, from "0" to "7." In addition, these connectors and LED's have corresponding switches on the slide switch assembly and pairs of connector pins in the two 8-pin connector blocks located immediately above the slide switch assembly.

Connect the 4" jumper wire from 4-pin connector block No. "0" to 8-pin block pair "0" (as shown on the Detail). Operate slide switch "0" and observe that the furthest right (zero) LED turns on and goes out.

Move the jumper wire to the "1" connector blocks, second from the right. Operate the slide switch and observe that the "1" LED turns on and goes out.

Progressively, and in the same manner, move the jumper wire to the "2", the "3", the "4", the "5", the "6", and the "7" connector blocks. Each time, operate the corresponding slide switch and observe that the correct LED is lit. Then remove the wire.

OPERATIONAL TESTS

This section of the Manual will check the basic Microprocessor functions to make sure they are working properly. The entries that will be made on the keyboard are not necessarily related to the actual use of the unit. Actual use of each function is explained in detail in the "Operation" section, starting on Page 45.

Refer to Pictorial 6-3 (Illustration Booklet, Page 7) to identify the function of each keyboard key.

NOTE: If you encounter any trouble in the following steps, turn the power off and remove the line cord plug from the AC outlet. Then refer to the "In Case of Difficulty" section on Page 91.

Each number step in the following charts shows which number or letter key to push, and what the resultant readout will be. Always push the keys in the sequence shown.

The following abbreviations are used on the Microprocessor keyboard:

ACCA ....................... Accumulator "A"
ACCB ....................... Accumulator "B"
PC ........................... Program Counter
INDEX ....................... Index Register
CC ........................... Condition Codes Register
SP ........................... Stack Pointer
RTI ........................... Return From Interrupt
SS ........................... Single Step
BR ........................... Break Point
AUTO .......................... Automatic Load
BACK .......................... Back
CHAN ......................... Change
DO ........................... Do
EXAM .......................... Examine
FWD .......................... Forward
NOTES:

1. In the following charts, the symbol "*" is used to denote a blank readout indication. The symbol "X" indicates a random figure.

2. When you make two-digit entries, the indicated Readout display will be shown after the second digit key has been released.

3. If you make an incorrect entry, return to step 1.

<table>
<thead>
<tr>
<th>STEP</th>
<th>FIRST PRESS</th>
<th>THEN PRESS</th>
<th>READOUT **</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RESET</td>
<td></td>
<td>CPU *UP.</td>
</tr>
<tr>
<td>2</td>
<td>EXAM E</td>
<td></td>
<td>___ Ad.</td>
</tr>
<tr>
<td>3</td>
<td>0 ACCA ACCB PC</td>
<td></td>
<td>0123XX</td>
</tr>
<tr>
<td>4</td>
<td>CHAN C</td>
<td></td>
<td>0123_</td>
</tr>
<tr>
<td>5</td>
<td>INDEX 4 CO 5</td>
<td></td>
<td>012345</td>
</tr>
<tr>
<td>6</td>
<td>EXAM E</td>
<td></td>
<td>___ Ad.</td>
</tr>
<tr>
<td>7</td>
<td>SP 6 RTI SS BR</td>
<td></td>
<td>6789XX</td>
</tr>
<tr>
<td>8</td>
<td>CHAN C</td>
<td></td>
<td>6789_</td>
</tr>
<tr>
<td>9</td>
<td>AUTO A BACK B</td>
<td></td>
<td>6789Ab</td>
</tr>
<tr>
<td>10</td>
<td>EXAM E</td>
<td></td>
<td>___ Ad.</td>
</tr>
<tr>
<td>11</td>
<td>CHAN C D E F FWD</td>
<td></td>
<td>CDEFFXX</td>
</tr>
</tbody>
</table>

** Shows readout as presented on LED's after the key is pressed.
You have now determined that the Microprocessor keys are operating properly. Continue the operational test as you enter the following simple program.

<table>
<thead>
<tr>
<th>STEP</th>
<th>FIRST PRESS:</th>
<th>THEN PRESS:</th>
<th>READOUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RESET</td>
<td></td>
<td>CPU* UP.</td>
</tr>
<tr>
<td>2</td>
<td>AUTO A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0 0 0 0</td>
<td>(Memory address)</td>
<td>0000 _ _</td>
</tr>
<tr>
<td>4</td>
<td>SP 8 6</td>
<td>(Load Accumulator A w/following two characters)</td>
<td>0001 _ _</td>
</tr>
<tr>
<td>5</td>
<td>0 ACCA 1</td>
<td>(Data for first step)</td>
<td>0002 _ _</td>
</tr>
<tr>
<td>6</td>
<td>BCK B 7</td>
<td>(Store Accumulator A at extended address of following four characters)</td>
<td>0003 _ _</td>
</tr>
<tr>
<td>7</td>
<td>CHAN C 1</td>
<td></td>
<td>0004 _ _</td>
</tr>
<tr>
<td>8</td>
<td>SP 6 FWD</td>
<td></td>
<td>0005 _ _</td>
</tr>
<tr>
<td>9</td>
<td>RTI 7 EXAM</td>
<td>(Jump to extended address of following four bytes of information)</td>
<td>0006 _ _</td>
</tr>
<tr>
<td>10</td>
<td>0 0</td>
<td></td>
<td>0007 _ _</td>
</tr>
<tr>
<td>11</td>
<td>0 0</td>
<td></td>
<td>0008 _ _</td>
</tr>
<tr>
<td>12</td>
<td>RESET</td>
<td>(To terminate AUTO addressing sequence)</td>
<td>CPU* UP.</td>
</tr>
<tr>
<td>13</td>
<td>DO D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>RESET</td>
<td></td>
<td>CPU* UP.</td>
</tr>
</tbody>
</table>
The preceding program entered information into memory storage that told the Microprocessor that you wished to turn on the decimal point of the "H" LED. The program was proved in steps 13 and 14 above.

Note in step 8 the characters "6F." This is the information that told the Microprocessor that you wished to address the "H" LED, and in particular, the decimal point; the "6" addressed the LED and the "F" addressed the decimal point. Refer to Detail 6-3A and note that each segment of an LED may be similarly addressed. Thus, to turn on each segment of the "H" (or "6") LED in turn, the terminal character must be changed to agree with the segment address.

In the following chart, the top bar of the "H" LED will be addressed and examined.

<table>
<thead>
<tr>
<th>STEP</th>
<th>FIRST PRESS:</th>
<th>THEN PRESS:</th>
<th>READOUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RESET</td>
<td></td>
<td>CPU*UP.</td>
</tr>
<tr>
<td>2</td>
<td>EXAM E</td>
<td></td>
<td>_ _ _ Ad.</td>
</tr>
<tr>
<td>3</td>
<td>0 0 0 0</td>
<td>INDEX</td>
<td>00046F</td>
</tr>
<tr>
<td>4</td>
<td>CHAN C</td>
<td></td>
<td>0004 _ _</td>
</tr>
<tr>
<td>5</td>
<td>SP 6 EXAM E</td>
<td>&quot;H&quot; LED, TOP BAR.</td>
<td>00046E</td>
</tr>
<tr>
<td>6</td>
<td>RESET</td>
<td></td>
<td>CPU*UP.</td>
</tr>
<tr>
<td>7</td>
<td>DO D</td>
<td></td>
<td>_ _ _ do.</td>
</tr>
<tr>
<td>8</td>
<td>0 0 0 0</td>
<td></td>
<td>_ * * * * *</td>
</tr>
<tr>
<td>9</td>
<td>RESET</td>
<td></td>
<td>CPU*UP.</td>
</tr>
</tbody>
</table>
As you observed in step 8, only the top-bar segment of the “H” LED lit up. You may further address and call up the remaining segments, in turn, of the same LED as follows: Repeat all nine steps in the preceding chart, with one exception. After you push the “CHAN/C” key in step 4, enter “6D,” then proceed with the remaining steps 6 through 8. The next time, at step 5, enter “6C,” and proceed. In the same manner, at step 5 of each repetition, enter “6B, 6A, 69 and 68.” Refer to Detail 6-3A to determine which LED segment should be lit.

To address the individual segments of the “I” LED, use the preceding chart and perform steps 1 through 4 as before. At step 5, enter “5F,” and proceed with steps 6 through 8. Note that the decimal on the “I” LED will light. Then, at step 5, one at a time, enter “5E, 5D, 5C, 5B, 5A, 59, and 58.” All segments of the remaining four LED’s may be called up in a like manner using, for example: “4F, 3F, 2F, 1F” at step 5 to light the respective decimal segments.

This completes the “Operational Tests” of your Microprocessor Trainer.

NOTE:

Provision has been made for you to install a 40-pin connector for system expansion. Brief instructions and a list of manufacturers are given below. If you do not wish to install a connector at this time, proceed directly to “Final Assembly” on Page 42.

Purchase and install a 40-pin connector* on the circuit board between IC2 and IC3. Then connect eight wires from the eight circuit board holes that connect the connector data pins to the eight data holes near IC9 and IC10. (See the “Schematic” and “Circuit Board X-Ray View.”) These wires connect the eight data lines [D0-D7]. Be sure you connect these wires properly so that data D0 goes to data line D0, etc.

*The connectors must have .025” square pins on .100” centers. The following manufacturers supply such connectors. Some are single strips of connectors that must be cut to length.

AP Products
929834-01 (2 strips required)
929836-01

Molex
22-04-2201 (2 strips required)

AMP
2-87215-0
2-87543-0
FINAL ASSEMBLY

PICTORIAL 7-1

Refer to Pictorial 7-1 for the following steps.

( ) Remove the paper backing from the “Heathkit” label. Carefully press the label in place on the upper portion of the cabinet top as shown.

( ) Press an LED grommet into the small round hole in the cabinet top.
Refer to Pictorial 7-2 for the following steps.

( ) Unplug the line cord.

( ) Turn the cabinet top upside down and position it near the cabinet bottom as shown in the Pictorial. As you lower the main circuit board down onto the inverted cabinet top, be sure that LED1, next to the Power switch, fits straight down into the LED grommet as shown in Detail 7-2A. NOTE: If the LED protrudes through the cabinet top too far, resolder the LED’s leads so the LED is closer to the circuit board.

( ) Secure the main circuit board to the cabinet top with eight #6 x 3/8" hex head screws.
Refer to Pictorial 7-3 for the following steps.

( ) Turn the cabinet top and main circuit board assembly right side up and fit the assembly into the cabinet bottom.

( ) Turn both cabinet halves bottom-side up as shown in the Pictorial; then secure the bottom to the top with eight #6 x 5/8" self-tapping screws.

( ) Remove the paper backing from the four feet and press them in place on the cabinet bottom in the smooth areas near the four corners as shown.

( ) Remove the paper backing from the blue and white label and press the label in place on the cabinet bottom. NOTE: Be sure to refer to the numbers on the blue and white label in any correspondence you have with the Heath Company about your kit.

( ) Remove the paper backing from the power label and press the label in place near the line cord as shown in the Pictorial.

This completes the “Final Assembly” of your kit. Proceed to “Operation.”
OPERATION

This section of the Manual describes the operation of your Trainer, explains the keyboard commands, describes how to enter programs, has several sample programs, contains the monitor listing and several subroutine flowcharts, shows the memory map, and lists the entire 6800 instruction set.

Pictorial 8-1 (Illustration Booklet, Page 8) gives a brief description of the switches, LED's, and connectors.

KEYBOARD

The keyboard allows you to quickly enter commands and data to the microprocessor. After you press the RESET key, the display will show CPU UP, and the next keyboard entry will be interpreted as a command. The following paragraphs discuss the various commands.

Display Accumulator A

1 Press this key and the contents of accumulator A will be displayed. The first four digits and decimal point identify the display, and the next two digits show the contents of the accumulator.

In the following example, the contents of accumulator A is 4A\textsubscript{16} (or binary 01001010).

Example: Acca.4A

Now you may change the contents of accumulator A if you wish. To do this, press the \textcircled{C} key. The display will now be:

Acca. _ _

With two key strokes, enter the new hexadecimal number you want in accumulator A.

Display Accumulator B

2 Press this key and the contents of accumulator B will be displayed. A typical display is:

Accb.5F

In this example, accumulator B contains 5F\textsubscript{16} (binary 01011111).

The contents of accumulator B can be changed in the same way that accumulator A is changed.

Display Program Counter

3 Press this key and the contents of the microprocessor's program counter will be displayed. The first two digits and decimal point identify the display, and the next four digits show the contents of the program counter.

Example: Pc.0040

In this example, the program counter contains 0040\textsubscript{16}. You may change the program counter by pressing the \textcircled{C} key and then entering the new hexadecimal number.

Display Index Register

4 Press this key and the contents of the index register will be displayed.

Example: In.FDF4.

You can change the register by pressing the \textcircled{C} key and then entering a new hexadecimal number.
Display Condition Codes Register

Press this key and the contents of the condition codes register (1's and 0's) will be displayed. The display letters (H, I, N, Z, V, and C) correspond to the letters assigned to the six condition codes. (See the “instruction set” on Page 89.)

Example: 001001

This register cannot be changed by pressing the C key.

Display Stack Pointer Register

Press this key and the contents of the stack pointer register will be displayed.

Example: SP.00d2

This register cannot be changed by pressing the C key.

Resume User's Program

Press this key and your program will start at the location contained in the program counter. This key is used to return to normal user program operation from breakpoints or single stepping.

Single Step User's Program

Press this key and the microprocessor will perform only one step of your program. The instruction to be performed is taken from the address contained in the program counter. After the step, the next instruction and its address are displayed. The displayed instruction may be changed by pressing the C key and then entering the new data. Also at this time, you may examine registers, memory, or use any of the other monitor functions.

Set Breakpoint

Press this key and you can then make an entry into the monitor breakpoint table. A breakpoint is a point where you want to stop the program to examine the microprocessor registers, memory, etc.

The display is __ __ br.

Enter the four digits of a hexadecimal address for the breakpoint. The address must be the address of an operational code in your program and that code must be in RAM. No breakpoints are possible in ROM. You may have up to four breakpoints in your program at any one time.

Do not press the RESET key. This clears all the breakpoints.

If you make an incorrect entry, and the entry is still displayed, press the C key as many times as necessary for the display to return to __ __ br. Then enter the correct address.

Auto Load Of Memory

Press this key and __ __ Ad will be displayed.

Enter the address you want to start at. Example: Enter 0, 0, A, and 4. The display is now:

00A4 __ .

Enter the 2-digit hexadecimal value you want entered at that address.

The display will now advance to the next address. You can continue changing memory data until you press the RESET key.

Display Previous Address

Press this key when an address and its data are displayed (you are examining memory with the E function, your program has come to a breakpoint, or you are single stepping your program), and the previous address and its data will be displayed. You may change this data by pressing the C key and then entering the new data.
Change Displayed Value

Press this key when an address and its data are displayed, and the data will be replaced with "...". Then enter the new hexadecimal value you want at this address.

You may use this function to correct a value you entered by mistake. However, if the monitor is expecting a command and the change function is not valid, the change command will be ignored.

DO User Program

Press this key and the display will become:

      - - - do.

Enter the beginning address of your program. Your program will now start at the new address instead of where the program counter was pointing. The display will become blank and the program will run until a display is called for, until it comes to a breakpoint, or until you press the reset key.

This key function combines several other functions. You could get the same result by displaying and changing the program counter and then pressing the 7 key.

Examine Memory

Press this key and the display will become:

      - - - Ad.

Enter a new address. The display will now indicate the data at this new address. You may now change the displayed value by using the 7 key or you can step backwards or forwards through memory using the B and F keys.

Display Next Address

Press this key when an address and its data are displayed, and the next address and its data will by displayed. You may change this data by pressing the 7 key and then entering the new data.
ENTERING PROGRAMS

Pictorial 8-2 shows the first two instructions of Sample Program 1 (in the following section) and indicates the various information they contain. This information is further described in the following paragraphs.

Instruction Address: This is usually called the Program Counter. In order to perform an instruction, the Program Counter must contain the address that is in this column. RTI and SS require the Program Counter to contain the address that is in this column for proper execution. The address entered after DO is pressed must be an instruction address. Breakpoints are not recognized except at instruction addresses.

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

Op code: This is a “byte of information referred to as machine code, it indicates in hexadecimal the operation to be performed.

Operand: This is additional hexadecimal information required to perform the operation. It may be zero, one, or two bytes as determined by the addressing mode.

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 three-letter indication of the source instruction. A fourth letter, A or B, is added to indicate which of two accumulators if the instruction applied to either one.

Mnemonic operand: Again, this is additional information that is required for the operation. It may be a label, address, or data. The $ sign indicates the information is a hex value. The # sign indicates the immediate addressing op code is to be used.

Comments: This is a brief description of what is happening in the program.

PICTORIAL 8-2
When you load a program into the Trainer, only the one, two, or three bytes of each instruction are entered. You may use either of two modes to enter the instructions: "Auto", or the more laborious "Examine and Change." Forward, Back, and Change are valid commands in the Examine, and Change mode and may be used to correct entry errors. However, they are not valid in Auto. If you make an error in the Auto mode, press the Reset and Auto keys. Then enter the address where the error was made and continue from there; or, remember where the error was made and then examine and change that memory location after you finish entering the entire program.

The following charts show the sequence of events to enter the first two instructions of sample program 1. The first chart shows the Examine and Change mode while the second chart shows the Auto mode.

### Examine and Change

Press the EXAM key and then enter the first instruction address, 0000, by pressing the 0 key four times. Then check the display and continue to enter the program as shown below.

<table>
<thead>
<tr>
<th>Display is</th>
<th>Press</th>
<th>Enter</th>
<th>Display is</th>
<th>Press</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000XX</td>
<td>CHAN</td>
<td>BD</td>
<td>0000BD</td>
<td>FWD</td>
</tr>
<tr>
<td>0001XX</td>
<td>CHAN</td>
<td>FC</td>
<td>0001FC</td>
<td>FWD</td>
</tr>
<tr>
<td>0002XX</td>
<td>CHAN</td>
<td>BC</td>
<td>0002BC</td>
<td>FWD</td>
</tr>
<tr>
<td>0003XX</td>
<td>CHAN</td>
<td>86</td>
<td>000386</td>
<td>FWD</td>
</tr>
<tr>
<td>0004XX</td>
<td>CHAN</td>
<td>01</td>
<td>000401</td>
<td>FWD</td>
</tr>
<tr>
<td>0005XX</td>
<td>CHAN</td>
<td>etc.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Auto

Press the AUTO key and then enter the first instruction address, 0000, by pressing the 0 key four times.

<table>
<thead>
<tr>
<th>Display is</th>
<th>Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000--</td>
<td>BD</td>
</tr>
<tr>
<td>0001--</td>
<td>FC</td>
</tr>
<tr>
<td>0002--</td>
<td>BC</td>
</tr>
<tr>
<td>0003--</td>
<td>86</td>
</tr>
<tr>
<td>0004--</td>
<td>01</td>
</tr>
<tr>
<td>0005--</td>
<td>etc.</td>
</tr>
</tbody>
</table>

Press RESET after last entry to exit the AUTO mode.
If Examine and Change is used, the last entry in sample program 1 (Page 55 and last page of Illustration Booklet) results in the display 0025 DA, and this display remains until a new command is entered through the keyboard.

If Auto is used, the last display will be the address of the next continuous memory location which is the last program instruction address plus the number of bytes in the instruction. In this program, 0024 plus two: 0026. The dash (or “prompt” characters) are displayed in the data locations.

After you enter a program, by either method, check the ending address to be sure that you have not omitted or double entered data.

Enter sample program 1 (on Page 55 and last page of Illustration Booklet) into your Trainer. Use either of the two entry methods.

If you used the Examine and Change mode to enter the program, the program can be run by pressing DO and entering the address of the instruction labeled “Start,” 0000. If you used the Auto mode, first press the RESET key to exit the Auto mode. Then press DO and enter the address of the instruction labeled “Start,” 0000.

USING BREAKPOINTS

We will now use sample program 1 to show how programs can be inadvertently changed and even “crash” when breakpoints are inserted at improper locations (at addresses other than the instruction address).

Press RESET and insert breakpoints at 0004 and 0005.

Press BR 0004.

Press BR 0005.

Start the program by pressing DO 0000.

Notice that the CPU has ignored the breakpoint at 0004 and stopped at 0005.

Examine 0003 and 0004 by pressing EXAM, 0003, and FWD. The instructions there are correct (86 and 01).

Examine accumulator A by pressing the ACCA key. Accumulator A has been loaded with the software interrupt instruction 3F that was temporarily placed at 0004 by the breakpoint at that address.

Watch the “H” display and press RTI. The 3F in accumulator A caused the first display to be incorrect. The program will stop again at 0005.

Insert a breakpoint at 0002 and then press RTI to resume execution. The program will run until it comes to 0002 which is changed by the breakpoint to 3F. The program will “crash” because of the wrong instruction and one of three things will happen: The display will be blank; all eights will be displayed; or all eights will appear, followed by CPU UP. In any case, press Reset to return control to the monitor program.

Press EXAM and enter 0000. Use FWD and CHAN to examine and correct errors introduced when the program crashed. You will always find the data at the breakpoint addresses has been changed. More often than not, the data at the breakpoint addresses will become 3F, although this may also change because the program crashed. Before you proceed, run the program to be sure all errors have been corrected.

In order to properly execute SS or RTI, the program counter must contain the instruction address where you wish to start. If single step begins at an incorrect address, the single step routine will not execute an invalid instruction and the display will not change. If the instruction at the PC address is a valid opcode, SS will execute the instruction using the following bytes as necessary and will continue unless it comes to an invalid instruction. RTI will try to execute the instruction in the same manner; except in the case of an invalid instruction, the program will probably crash. We will use SS to illustrate what happens.

Push RESET. Examine the Program Counter by pushing PC. Then change it to 0016 by pushing CHAN and 0016.

Press SS. The instruction at 0016 is not a valid instruction. In the single step mode, the machine will reject the instruction and 0016 FD will continue to be displayed and nothing happens. If RTI is pressed, the program will crash as it would when an invalid instruction is encountered. Probably only the first in-
struction will be changed, if any, in this particular circumstance. If you press RT1 to see what happens, examine the program afterwards and correct any errors introduced; then run the program to be sure it is correct before proceeding.

Examine and change the program counter to 000F by pressing PC, CHAN, and 000F. Press SS. In this case FE is a valid instruction, LDX extended, and X is loaded from non-existent memory locations 3ACE and 3ACF.

Press SS. Here again 2F is a valid instruction (a conditional branch BLE). A branch may occur to 0015 or the program may fall through to 0014. In either case, two incorrect instructions have been performed in place of two or three correct instructions introducing error in the program. This is of no great consequence in this program but may be in another. Since an invalid instruction was not encountered, placing the program counter at 000F and pressing RT1 would do exactly the same thing.

Now sample program 1 will be used to illustrate a procedure using breakpoints and single step to go through a program.

There are two important considerations pertaining to reserved memory bytes to keep in mind. First: DIGADD is used by all monitor routines. If you examine these memory locations, 00F0 and 00F1, you will always find C12F, “V” display address, there because the examine command puts it there before it outputs the data. Secondly: DIGADD is always loaded with C16F, “H” display address, when DO or RT1 are used.

Single step uses RMB TEMP, T1 and T0 in common with many of the monitor routines. Single step will replace information stored at these locations by the monitor routines. As a result, the routine may return with incorrect information or it may not be able to return at all and the program will crash.

When the program stops, at a breakpoint or after a single step, the address of the next instruction (contained in the program counter) and the instruction will be displayed. You may examine and make changes to any register (except stack pointer) or address provided you DO NOT change the program counter. The instruction displayed when the program stopped will be the next one executed when SS or RT1 is pressed, regardless of what is being displayed.

The following procedure gives instructions. The six characters on the right, on the same line, indicate what the display should be after you perform the instruction. You will be instructed to examine registers affected by the instruction that has been executed.
You may examine any other registers or memory locations if you wish. The comment after an instruction is explanatory information.

<table>
<thead>
<tr>
<th>INSTRUCTION</th>
<th>DISPLAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Press RESET.</td>
<td>CPU UP.</td>
</tr>
<tr>
<td>Press PC, CHAN, and enter 0000. The program counter now contains the start</td>
<td>Pc 0000</td>
</tr>
<tr>
<td>address.</td>
<td></td>
</tr>
<tr>
<td>Press SP. This is the next location available on the stack. The JSR</td>
<td>SP 00d2</td>
</tr>
<tr>
<td>instruction should store the address for return from the REDIS subroutine</td>
<td></td>
</tr>
<tr>
<td>(0003) at this location.</td>
<td></td>
</tr>
<tr>
<td>Press SS, Jump to REDIS.</td>
<td>FCbC dF</td>
</tr>
<tr>
<td>Press EXAM and 00D1.</td>
<td>00d1 00</td>
</tr>
<tr>
<td>Press FWD. Return address is on stack.</td>
<td>00d2 03</td>
</tr>
<tr>
<td>Press BR and enter 0003. To get past monitor routine.</td>
<td>0003 br</td>
</tr>
<tr>
<td>Press RTI. Might normally use examine to check result of routine. In</td>
<td>0003 86</td>
</tr>
<tr>
<td>this case, DIGADD RMB is loaded with C16F. Examine will just change what</td>
<td></td>
</tr>
<tr>
<td>is there.</td>
<td></td>
</tr>
<tr>
<td>Press SS.</td>
<td>0005 20</td>
</tr>
<tr>
<td>Press ACCA. (A) loaded with correct value.</td>
<td>Acca.01</td>
</tr>
<tr>
<td>Press SS. Branch to OUT-offset correct.</td>
<td>000E bd</td>
</tr>
<tr>
<td>Press SS. Jump to OUTFCH.</td>
<td>FE3A dF</td>
</tr>
<tr>
<td>Press BR and enter 0011. To get past monitor routine. Could check stack</td>
<td>0011 br</td>
</tr>
<tr>
<td>here if desired.</td>
<td></td>
</tr>
<tr>
<td>Press RTI. Exit OUTFCH address of next display in DIGADD; Do not check.</td>
<td>0011 CE</td>
</tr>
<tr>
<td>Press SS.</td>
<td>0014 09</td>
</tr>
<tr>
<td>Press INDEX. Is (X) loaded?</td>
<td>In.2F00</td>
</tr>
<tr>
<td>Press SS.</td>
<td>0015 26</td>
</tr>
<tr>
<td>Press INDEX. Is (X) decremented?</td>
<td>IN.2EFF</td>
</tr>
<tr>
<td>Press CC. Z bit clear if (X) not 0000 yet.</td>
<td>XXX0 XX (X = don’t care)</td>
</tr>
<tr>
<td>Press SS. Branches to WAIT if Z was clear.</td>
<td>0014 09</td>
</tr>
<tr>
<td>Press INDEX, CHAN, and enter 0001.</td>
<td>In.0001</td>
</tr>
<tr>
<td>Press SS.</td>
<td>0015 26</td>
</tr>
</tbody>
</table>
Press CC. (X) decremented to 0000 sets Z bit. Should drop through branch now. XXX1 XX

Press SS. It did. 0017 16

Press ACCA. Acca.01

Press SS. 0018 5d

Press ACCB. What was in (A) should be in (B). Accb.01

Press SS. 0019 26

Press CC. Z bit clear if (B) not 00. XXX0 XX

Press SS. Branches to SAME if Z is clear. 0007 d6

Press SS. 0009 Cb

Press ACCB. When the program runs normally, (B) at this point would be 5F because exit from OUTCH would be with the next display address, C15F, in DIGADD. Single step has caused DIGADD to be C10F. Accb.0F

Press SS. 000b d7

Press ACCB. Hex 10 has been added to (B). Accb.1F

Press SS. (B) has been stored at DIGADD. No reason to examine 00F1 since EXAM and SS will change what is there anyway. 000d 48

Press ACCA. Acca.01

Press SS. 000E bd

Press ACCA. ACCA was 0000 0001 binary (01 hex). It has been shifted left and is now 0000 0010 binary (02 hex). The program is back to jump to OUTCH again. The same method as used before would get you back 0019 again. The program has proven good to that point so we will use a different method. Acca.02

Press Reset. This clears the previous breakpoints. CPU UP

Press BR and enter 0018. 0018 br.

Press DO and enter 0000. You may have noticed the program ran up to the breakpoint and the counter segment in "H" was momentarily lit. Now you are in another loop. You could press RT1 seven times and go back through the loop until (B) is 00. Again, since the branch is operating properly it is easier to change (B) to 00 and continue. 0018 5d

Press ACCB, CHAN, and enter 00. Accb.00
Press SS.  

Press CC. The Z bit is set and the program should fall through the branch.

Press SS. It did.

Press SS.

Press ACCA. (A) is loaded correctly.

Press SS.

Press Index. This is DIGADD again. Although the program has just finished with the “H” display, single step has placed C10F in DIGADD. This happens to be the address that will be in DIGADD after DP goes out in the “C” display and should result in a branch back to START.

Press SS. Same conditional BRANCH.

Press CC. Z is set and the program should fall through.

Press SS. It did.

Press SS. Every instruction in the program has been run except for the conditional branch at 0022.

Press Reset. Clears the breakpoint at 0018

Press BR and enter 001F.

Press DO and enter 0000.

Press Index. This time the program runs straight through until after (X) is loaded from DIGADD (at 001D) without an intervening single step or breakpoint. All segments were turned on and off in the “H” display and “I” display address C15F is in the index register as it should be.

Press SS. Conditional branch.

Press CC. Z is clear and a branch to out should take place.

Press SS. It did.

The entire program has now been run.
SAMPLE PROGRAMS

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

SAMPLE 1
TURN ON AND OFF EACH SEGMENT IN
SEQUENCE BEGINNING AT H DISPLAY
USES MONITOR SUBROUTINES REDIS AND OUTCH
NOTE: ONE DP IN EACH DISPLAY IS ACTIVE

0000 BD FCBC START JSR REDIS SET UP FIRST DISPLAY ADDRESS
0003 86 01 LDA A #$01 FIRST SEGMENT CODE
0005 20 07 BRA OUT
0007 D6 F1 SAME LDA B DIGADD+1 FIX DISPLAY ADDRESS
0009 CB 10 ADD B #$10 FOR NEXT SEGMENT
000B D7 F1 STA B DIGADD+1
000D 48 ASL A NEXT SEGMENT CODE
000E BD FE3A OUT JSR OUTCH OUTPUT SEGMENT
0011 CE 2F00 LDX #$2F00 TIME TO WAIT
0014 09 WAIT DEX
0015 26 FD BNE WAIT TIME OUT YET?
0017 16 TAB
0018 5D TST B LAST SEGMENT THIS DISPLAY?
0019 26 EC BNE SAME NEXT SEGMENT
001B 86 C1 LDA A #$01 RESET SEGMENT CODE
001D DE FD LDX DIGADD NEXT DISPLAY
001F 8C C10F CPX #$C10F LAST DISPLAY YET?
0022 26 EA BNE OUT
0024 20 DA BRA START DO AGAIN
SAMPLE 2
TURNS ALL DISPLAYS OFF AND ON
DISPLAYS HEX VALUE AT 0044
USES MONITOR SUBROUTINES REDIS, OUTCH AND OUTHEX

0030 BD FCBC START JSR REDIS FIRST DISPLAY ADDRESS
0033 4F CLEAR CLR A
0034 BD FE3A JSR OUTCH TURN ALL SEGMENTS OFF
0037 DE F0 LDX DIGADD NEXT DISPLAY
0039 0C C10F CPX #$C10F LAST DISPLAY YET?
003C 2F F5 BNE CLEAR
003E 0D 13 BSR HOLD
0040 BD FCBC JSR REDIS FIRST DISPLAY ADDRESS
0043 86 08 LDA #$08 HEX VALUE TO DISPLAY
0045 BD FE28 OUT JSR OUTHEX OUTPUT CHARACTER
0048 DE F0 LDX DIGADD NEXT DISPLAY
004A 0C C10F CPX #$C10F LAST DISPLAY YET?
004D 2F F6 BNE OUT
004F 0D 02 BSR HOLD
0051 20 DD BRA START DO AGAIN
0053 CE FF00 HOLD LDX #$FF00 TIME TO WAIT
0056 09 WAIT DEX
0057 26 FD BNE WAIT TIME OUT YET?
0059 39 RTS
SAMPLE 3

OUTPUTS MESSAGE BY DISPLAYING UP TO SIX
CHARACTER WORD ONE WORD AT A TIME
USES MONITOR SUB ROUTINE OUTSTO
NOTE: DP MUST BE LIT TO INDICATE END OF STRING
TO EXIT OUTSTR. DP IS PLACED IN THE
SEVENTH DISPLAY POSITION TO FULFILL THIS
REQUIREMENT WITHOUT ACTUALLY BEING DISPLAYED.

0060 BD FD8D START JSR OUTSTO LEFT DISPLAY OUT WORD
0063 00 FCB $00,$3B,$7E,$3E,$05,$00,$80 YOUR
0064 3B
0065 7E
0066 3E
0067 05
0068 00
0069 80
006A 8D 3F BSR HOLD HOLD DISPLAY
006C BD FD8D JSR OUTSTO LEFT DISPLAY OUT WORD
006F 00 FCB $00,$79,$33,$7E,$7E,$00,$80 3400
0070 79
0071 33
0072 7E
0073 7E
0074 00
0075 80
0076 8D 33 BSR HOLD HOLD DISPLAY
0078 BD FD8D JSR OUTSTO LEFT DISPLAY OUT WORD
007B 00 FCB $00,$00,$30,$58,$00,$00,$80 IS
007C 00
007D 30
007E 58
007F 00
0080 00
0081 80
0082 8D 27 BSR HOLD HOLD DISPLAY
0084 BD FD8D JSR OUTSTO LEFT DISPLAY OUT WORD
0087 00 FCB $00,$00,$3E,$67,$00,$00,$80 UP
0088 00
0089 3E
008A 67
008B 00
008C 00
008D 80
008E 8D 1B BSR HOLD HOLD DISPLAY
0090 BD FD8D JSR OUTSTO LEFT DISPLAY OUT WORD
0093 00 FCB $00,$00,$7D,$15,$3D,$00,$80 AND
0094 00
0095 7D
0096 15
0097 3D
0098 00
0099 80
009A 8D 0F  BSR  HOLD  HOLD DISPLAY
009C BD FD8D JSR  OUTSTO  LEFT DISPLAY OUT WORD
009F 05 FCB  $05,$10,$15,$15,$15,$15,$15,$80  RUNNIN
00A0 1C
00A1 15
00A2 15
00A3 10
00A4 15
00A5 80
00A6 8D 03  BSR  HOLD  HOLD DISPLAY
00A8 7E 0060  JMP  START  DO AGAIN
00AB CE FF0D HOLD  LDX  #$FF0D  TIME TO WAIT
00AE 09 WAIT  DEX
00AF 26 FD  BNE  WAIT  TIME OUT YET?
00B1 39 RTS
SAMPLE 4
OUTPUTS SAME MESSAGE AS PROGRAM 3
IN TICKER TAPE FASHION
USES MONITOR SUB ROUTINES REDIS AND OUTSTR

0000 7F 0007 START CLR MORE+1 CLEAR POINTER
0003 CE 002A NEXT LDX #MESSA MESSAGE ADDRESS
0006 A6 00 MORE LDA A O.X GET CHARACTER
0008 A7 2D STA A OUT+3-MESSA.X STORE CHAR. AT OUT PLUS
000A 08 INX NEXT CHARACTER
000B 8C 0030 CPX #$30 FULL STRING YET?
000E 26 F6 BNE MORE
0010 8D 11 BSR HOLD HOLD DISPLAY
0012 BD FCBC JSR REDIS FIRST CHAR. TO "H" DISPLAY
0015 BD 0054 JSR OUT
0018 96 07 LDA A MORE+1 FIRST CHARACTER NUMBER
001A 4C INC A MOVE STRING UP ONE CHARACTER
001B 97 07 STA A MORE+1 NEW FIRST CHARACTER
001D 81 25 CMP A #$25 LAST CHARACTER TO "H" YET?
001F 26 E2 BNE NEXT BUILD NEXT STRING
0021 20 DD BRA START DO AGAIN
0023 CE 6000 HOLD LDX #$6000 TIME TO WAIT
0026 09 WAIT DEX
0027 26 PD BNE WAIT TIME OUT YET?
0029 39 RTS
002A 08 MESSA FCB $08,$08,$08,$08,$08,$08 ----
002B 08
002C 08
002D 08
002E 08
002F 08
0030 38 FCB $38,$7E,$3E,$05,$00,$00 YOUR
0031 7E
0032 3E
0033 05
0034 00
0035 00
0036 79 FCB $79,$33,$7E,$7E,$00,$00 3400
0037 33
0038 7E
0039 7E
003A 00
003B 00
003C 30 FCB $30,$5E,$00,$00,$3E,$67 IS UP
003D 5B
003E 00
003F 00
0040 3E
0041 67
0042 00 FCB $00,$00,$7D,$15,$3D,$00,$00 AND
0046 3D
0047 00
0048 00
0049 05      FCB  $05,$1C,$15,$15,$10,$15  RUNNIN
004A 1C
004B 15
004C 15
004D 10
004E 15
004F 08      FCB  $08,$08,$08,$08,$08  -----  
0050 08
0051 08
0052 08
0053 08
0054 BD FE52 OUT  JSR  OUTSTR  OUTPUT CHARACTER STRING  
  OUTPUT STRING STORED HERE
0057 00      FCB  $00,$00,$00,$00,$00,$00,$00
0058 00
0059 00
005A 00
005B 00
005C 00
005D 80
005E 39
SAMPLE 5

THIS PROGRAM CONTINUOUSLY CHANGES THE HEX VALUE STORED AT KEY+1 UNTIL ANY HEX KEY IS DEPRESSED. THE RIGHT DP IS LIT TO INDICATE A VALUE HAS BEEN SET. THE USER THEN DEPRESSES THE VARIOUS HEX KEYS TO LOOK FOR THE SELECTED VALUE. THE RELATIONSHIP OF DEPRESSED TO CORRECT KEY IS MOMENTARILY DISPLAYED AS HI OR LO. DP AGAIN LIGHTS INDICATING TRY AGAIN. DEPRESSING THE CORRECT KEY DISPLAYS YES! WHICH REMAINS UNTIL ANY KEY IS DEPRESSED SETTING A NEW VALUE TO FIND.

USES MONITOR SUB ROUTINES ENCODE, OUTSTO, INCH

0060 7F 0086 START CLR KEY+1 CLEAR KEY POINTER
0063 C6 20 ILL LDA B #$20 VIOLATION COUNT
0065 BD FDEB ILL1 JSR ENCODE WAIT FOR ILLEGAL INTERVAL
0068 25 F9 BCS ILL STILL LEGAL?
006A 5A DEC B
006B 26 F8 BNE ILL1 NOT A FELONY
006D C6 20 LEGAL LDA B #$20 TIME UNTIL PAROLE
006F 8D 3B BSR CODE CHANGE KEY TO FIND
0071 BD FDEB LEGAL1 JSR ENCODE SET KEY TO FIND
0074 24 F7 BCC LEGAL KEY TO FIND SET?
0076 5A DEC B
0077 26 F8 BNE LEGAL1 GOOD KEY?
0079 BD FD8D OUTDP JSR OUTSTO OUTPUT STRING
007C 00 FCB $00,$00,$00,$00,$00,$00,$00,$00 DP TO "C"

* DP LIT FIND SELECTED KEY

0082 BD FDF4 JSR INCH LOOK FOR KEY
0085 C6 86 KEY LDA B #KEY+1 GET KEY VALUE
0087 11 CBA IS IT RIGHT KEY?
0088 27 14 BEQ YES IF CORRECT
008A 22 2A BHI HIGH IF GREATER THAN KEY+1 VALUE
008C BD FD8D JSR OUTSTO OUTPUT STRING
008F 00 FCB $00,$00,$00,$00,$00,$0E,$0E,$80 LO
0090 00
0091 00
0092 00
0093 0E
0094 7E
0095 80

0096 CE 6000 HOLD LDX #$6000 TIME TO HOLD DISPLAY
0099 09 WAIT DEX
009A 26 FD BNE WAIT LONG ENOUGH YET?
009C 20 DB BRA OUTDP TRY AGAIN
009E BD FD8D YES JSR OUTSTO OUTPUT STRING
OOA1 00 FCB $00,$00,$3B,$4F,$5B,$80 YES!
OOA2 00
OOA3 3B
OOA4 4F
OOA5 5B
OOA6 70
OOA7 20 B7 BRA START DO AGAIN
OOA9 96 86 CODE LDA A KEY+1 CURRENT KEY VALUE
OOAB 4C INC A NEXT KEY
OOAC 97 86 STA A KEY+1 KEY TO FIND
OOAE 81 10 CMP A #$10 CAN'T BE GREATER THAN F
OOB0 26 03 BNE GOOD
OOB2 7F 0086 CLR KEY+1 MAKE IT 0
OOB5 39 GOOD RTS
OOB6 BD FD8D HIGH JSR OUTST0 OUTPUT STRING
OOB9 37 FCB $37,$30,$00,$00,$00,$00,$80 HI
OOBA 30
OOBB 00
OOBC 00
OOBD 00
OOBE 00
OOBF 86
OOCD 7E 0096 JMP HOLD
SAMPLE 6
THIS IS A TWELVE HOUR CLOCK PROGRAM
THE ACCURACY IS DEPENDENT UPON THE MPU CLOCK
FREQUENCY AND THE TIMING LOOP AT START.
CHANGING THE VALUE AT 0005/6 BY HEX 100
CHANGES THE ACCURACY APPROXIMATELY 1 SEC/MIN.
HOURS, MINUTE, SECOND RMB 0001/2/3 ARE LOADED
WITH THE STARTING TIME. THE FIRST DISPLAY
IS ONE SECOND AFTER START OF THE PROGRAM.
SECONDS WILL BE CONTENT OF SECOND RMB +1.
USES MONITOR SUB ROUTINES REDIS, DISPLAY.
NOTE: START THE PROGRAM AT 0004.

0001 00 HOURS RMB 1
0002 00 MINUTE RMB 1
0003 00 SECOND RMB 1
0004 CE E500 START LDX #$B500 ADJUST FOR ACCURACY
0007 09 DELAY DEX
0008 26 FD BNE DELAY WAIT ONE SECOND
000A CE 60 LDA B #$60 SIXTY SECONDS, SIXTY MINUTES
000C CD SEC ALWAYS INCREMENT SECONDS
000D 8D 0F BSR INCS INCREMENT SECONDS
000F 8D 10 BSR INCMH INCREMENT MINUTES IF NEEDED
0011 CE 13 LDA B #$13 TWELVE HOUR CLOCK
0013 8D 0C BSR INCMH INCREMENT HOURS IS NEEDED
0015 BD FCBC JSR REDIS RESET DISPLAY ADDRESS
0018 CE 03 LDA B #$3 NUMBER OF BYTES TO DISPLAY
001A 8D 16 BSR PRINT DISPLAY HOURS, MINUTES, SECONDS
001C 20 6E BRA START DO AGAIN
001E CE 0003 INCS LDX #$SECOND POINT X AT TIME RMB
0021 A6 00 INCMH LDA A 0.X GET CURRENT TIME
0023 89 00 ADC A #$0 INCREMENT IF NECESSARY
0025 19 DAA FIX TO DECIMAL
0026 11 CBA TIME TO CLEAR?
0027 25 01 EBC STORE NO
0029 4F CLR A
002A A7 00 STORE STA A 0.X STORE NEW TIME
002C 09 DEX NEXT TIME RMB
002D 07 TPA
002E 88 01 EOR A #1 COMPLEMENT CARRY BIT
0030 06 TAP
0031 39 RTS
0032 9E 01 PRINT LDA A $D1 WHAT'S IN HOURS?
0034 2E 03 BNE CONTIN IF NOT ZERO
0036 7C 0001 INC HOURS MAKE HOURS ONE
0039 08 CONTIN INX POINT X AT HOURS
003A 7E FD7B JMP DISPLAY OUTPUT TO DISPLAYS
SAMPLE 7

This program calculates the op code value for branch instructions using the last two digits of the branch and destination addresses. The branch address is entered first and displayed at "H" and "I". The destination address is then entered and displayed at "N" and "Z". The op code is then calculated and displayed at "V" and "C". The display is held until new information is entered. Since only two digits are entered, it is necessary to make an adjustment if the hundreds digit in the two addresses is not the same. For example to calculate the offset of a branch from 00CD to 011B, subtract a number from both addresses that will make the greater address less than 100. For ease of calculation in this case, subtract 0C from both addresses and enter the results 0D and 5B in the program. Since the difference between the addresses is unchanged the correct op code (4C) will be displayed. If the distance is too great for branching no. will appear at "V" and "C".

Uses monitor sub routines

REDS IHB OUTBYT OUTSTR

0000 BD FCBC START
0003 BD FEO9 JSR REDIS FIRST DISPLAY AT "H"
0006 16 TAB IHB INPUT BRANCH ADDRESS
0007 BD FEO9 JSR IHB INPUT DESTINATION ADDRESS
000A 11 CBA FORWARD OR BACK?
000B 25 0C BCS BACK IF BACK
000D CB 02 FRWD ADD B #$02 ADJUST 2 BYTES
000F 10 SBA FIND DISTANCE
0010 81 80 CMP A #$80 IS IT LEGAL?
0012 24 12 BCC NO IF NOT
0014 BD FE20 OUT JSR OUTBYT OUTPUT BRANCH OP CODE
0017 20 E7 BRA START LOOK FOR NEW ENTRY
0019 40 BACK NEG A MAKE A MINUS
001A 1B ABA ADD A #$02 ADJUST 2 BYTES
001B 8B 02 ADD A #$02 ADD A AND B
001D 43 COM A GET COMPLIMENT
001E 8B 01 ADD A #$01 MAKE IT TWO'S
0020 81 80 CMP A #$80 IS IT LEGAL?
0022 25 02 BCS NO IF NOT
0024 20 EE BRA OUT OUTPUT BRANCH OP CODE
0026 BD FE52 NO JSR OUTSTR OUTPUT STRING
0029 15 FCB $15,$9D NO.
002A 9D
002B 20 D3 BRA START LOOK FOR NEW ENTRY
SUBROUTINE FLOW CHARTS

Following, are flow charts of several subroutines. These are helpful when you write your own programs. The entry requirements necessary to call these subroutines and their exit conditions are also shown.

RESET/MAIN Routine

When the Reset key is released, the CPU outputs FFFE and FFFF to get a starting address. This is the address of the top two locations in the monitor ROM which in turn outputs FC00, the beginning address of the reset routine.

Reset first initializes the stack pointer to 00EB and outputs CPU UP, to the displays. The index register is set to 00CB (the start of the user's stack) and this value is stored in the user's stack pointer at location 00F2.

Breakpoints are cleared by placing FF in the eight RAM locations, 00E4-00EB. The program then goes into the main monitor loop. The contents of accumulator A, which is FF at this point, is stored at T1 and the address to return from command handler subroutines (FC19) is placed on the stack.

The program next calls INCH to scan and encode the keyboard. The program stays in INCH until a key is found closed.

The FORWARD and BACK commands are legal only after execution of the EXAM or SINGLE STEP commands. RAM location 00EE (T1) is cleared if FORWARD and BACK are legal commands. When INCH returns a key closure, T1 is tested to see if FORWARD and BACK are legal. If they are legal, a branch is made to MAIN 2 to obtain the subroutine address to handle the command and then goes to that handler. If FORWARD and BACK are not legal commands, tests are made to see that they are not the key closed before going to MAIN 2. If FORWARD or BACK is found to be the key closed, a branch back to MAIN 1 occurs and INCH is again called to look for a legal key closure.
INCH Routine

INCH guards against the entry of a false output from the keyboard due to contact bounce or pressing more than one key. ACCB is loaded with hex 20 and ENCODE is called to scan the keyboard. If C is set (key closed), a branch occurs back to the beginning. If C is clear (no closure), ACCB is decremented and ENCODE is called again. ENCODE must return C clear 32 consecutive times (approx. 9 ms) to exit this loop. The second half of the routine is then entered. This half is identical to that described above, except C must be set 32 consecutive times before exit with the hex value of the key closed in ACCA.
ENCODE Routine

ENCODE is the keyboard scanning routine. If a key is closed, the value is found in the hex table and loaded in ACCA. The C bit in the condition code register is set to indicate a valid key. If no key is closed or if the value is not in HEX-TAB, the C bit is cleared.

ENTRY: None.
EXIT: \( [A] \) contains hex value of key closed.
\( "C" \) set for valid condition.

ENCODE ROUTINE
OUTCH Routine

OUTCH outputs a character to the display whose address is contained at memory location DIGADD (00F0-00F1). This routine may be entered at OUTO if the index register does not need to be saved. The code for the character to be displayed must be in accumulator A when the routine is entered. The following drawing shows the segment identification and the corresponding positions in the eight bits of accumulator A. A logic one in a bit will cause that segment to light, whereas a logic zero will keep it off. The hex and corresponding bit codes are shown for two characters used in the monitor program. The most significant bit is DP and the least significant bit is segment g.

Segment codes used by the monitor program are shown at the end of the monitor listing.

<table>
<thead>
<tr>
<th>SEGMENT</th>
<th>DP</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEX C 4E</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>LTR c OD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

OUTCH Routines

ENTRY: ACCA contains segment code. DIGADD contains address of desired digit. Entry at OUTO if index register is to be saved.

EXIT: DIGADD contains address of next digit to right.
OUTHEX Routine

OUTHEX determines the segment code for a hex value contained in the four least-significant bits (LSB) of accumulator A. Subroutine OUTO is then called to output the hex value to the display whose address is obtained from DIGADD.

**OUTHEX ROUTINE**

ENTRY: ACCA contains hex value.

EXIT: Address of next digit to right contained in DIGADD.

OUTBYT Routine

OUTBYT outputs two hex values contained in accumulator A to two adjacent displays. The value contained in the four most-significant bits (MSB) are moved to the LSB positions. OUTHEX is called to determine the segment code and in turn calls OUTO to output the character to the display addressed at DIGADD. Accumulator A is restored, and OUTHEX and OUTO are called again to output the LSB to the next display to the right.

**OUTBYT ROUTINE**

ENTRY: ACCA contains two hex values.

EXIT: Digit address for next digit to right contained in DIGADD.
OUTST1 Routine

OUTST1 outputs a string of characters from left to right on the displays. The first character is output to the display whose address is contained in the index register upon entry to the routine. The last character must have the decimal point lit to indicate the end of the string. Adding hex 80 to the desired segment code causes the decimal point to be lit. For example, if the last character is to be LTR P, hex 67 (the last character code) would be hex 67 plus hex 80, or hex E7.

The routine may be entered at OUTSTJ or at OUTSTO (with the first character appearing in the left-most display) or at OUTSTA (with the first character appearing in the V display). Entry at OUTSTR requires the address for the first character to be in DIGADD. Exit from the routine is to the next instruction, which is one plus the address of the last character.

OUTST1 ROUTINE

ENTRY: Calling convention must be JSR to entry point. Segment codes for output characters from left to right at consecutive addresses immediately following jump instruction. Entry at OUTSTJ or OUTSTA set up for left-most character at H or V display respectively. Entry at OUTST1 requires (X) contain left-most digit address. Entry at OUTSTR requires left-most digit address at DIGADD. Decimal point must be lit on last character.

EXIT: To next instruction at 1 + address of last character.
ACCA is clear.
DIGADD contains address of display to right of last digit lit.
DSPLAY Routine

DSPLAY is called to output two or three bytes to the displays. The number of bytes to output is indicated by the contents of accumulator B. This routine could be called to output one byte, although OUTBYT would normally be called for this purpose.

Accumulator A is loaded with a byte value from an address contained in the index register and OUTBYT is called to output the byte to the displays. Then the index register is incremented to get the next byte, accumulator B is decremented, and OUTBYT is called again. When accumulator B is zero, all bytes have been output and the index register and accumulator B are restored before returning from the routine.

DSPLAY ROUTINE

ENTRY: [X] contains address of first byte.
[B] contains number of byte to output.
DICARD contains address of digit.

EXIT: [X] and [B] unchanged.
Address of next digit to right contained in DICARD.

![Diagram of DSPLAY Routine](image)
**IHB Routine**

IHB outputs two hex characters to the displays corresponding to two consecutive key closures and returns to the calling routine with the byte value of the two closures in accumulator A.

INCH is called to get the value of the first key closure. OUTHEX is called to display the value on the display whose address is contained at DIGADD. The value contained in the four LSB of accumulator A is moved to the four MSB of accumulator A and then saved in accumulator B.

INCH is called again to get the value of the second key closure. OUTHEX is then called again and this value is displayed on the next display to the right.

The contents of accumulators A and B are combined and placed in accumulator A. Accumulator A now contains the byte value of the two closures. The MSB contains the first closure value and the LSB contains the second value. Accumulator B is restored, accumulator A is pushed onto the stack, and ENCODE is called to wait for the release of the second key. When the key is released, the byte is pulled from the stack and the program returns to the calling routine with the byte contained in accumulator A.

**REDIS Routine**

REDIS is a short routine to reset the address at DIGADD to the left-most display digit.

**REDIS ROUTINE**

**ENTRY:** None.

**EXIT:** DIGADD contains address of left-most digit.

**IHB ROUTINE**

**ENTRY:** None.

**EXIT:** ACCA contains byte value.

Digit address for next digit to right contained in DIGADD.
MONITOR LISTING

Your Trainer is controlled by IC12, the "read only memory" (ROM). The following is a listing of the program stored in this IC.

```
FC00 ORG $FC00

** RESET - CLEAR BREAKPOINT TABLE AND INITIALIZE STACK

FC00 8E 00 EB RESET LDS $>2*NBR+BKTBL-1
FC03 BD FD 8D JSR OUTSTO
FC06 4E 67 3E FCC HEXC+LTRP+LTRU+O+LTRUE+LTRP+*80
FC0C CE 00 CB LDX @USRSTK
FC0F DF F2 STX USERS SET UP FOR USER
FC11 86 FF LDA A $FF
FC13 C6 08 LDA B $2*NBR
FC15 36 RESEL PSH A
FC16 5A DEX B
FC17 26 FC BNE RESEL

** MAIN - MAIN MONITOR LOOP
*
* HANDLERS RETURN:
* (B) = NUMBER BYTES SUBJECT TO 'CHANGE'
* (X) = ADDRESS BYTES SUBJECT TO 'CHANGE'
* (A) = 0 ENABLES 'FORWARD' AND 'BACK'

FC19 97 EE MAIN STA A T1
FC1B 86 19 LDA A $MAIN/256+MAIN LO ORDER RET.ADDR.
FC1D 36 PSH A
FC1E 86 FC LDA A $MAIN/256 HI ORDER BYTE RET. ADDR.
FC20 36 PSH A
FC21 BD FD F4 MAIN1 JSR INCH GET COMMAND
FC24 7D 00 EE TST T1
FC27 27 08 BEQ MAIN2 FORWARD OR BACK OK
FC29 81 0F CMP A $FF
FC2B 27 F4 BEQ MAIN1 ILLEGAL NOW
FC2D 81 08 CMP A $FB
FC2F 27 F0 BEQ MAIN1 ALSO ILLEGAL NOW
FC31 DF EC MAIN2 STX T0
FC33 CE FF B4 LDX @CMDTAB-2
FC36 08 MAIN3 INX
FC37 08 INX GET HANDLER ADDRESS
FC38 4A DECA
FC39 2A FB BPL MAIN3
FC3B A6 01 LDA A 1+X TARGET ADDRESS ONTO STACK
FC3D 36 PSH A
FC3E A6 00 LDA A 0+X
FC40 36 PSH A
FC41 DF EC LDX T0 RESTORE X
FC43 96 EE LDA A T1
FC45 39 ZERO RTS JUMP TO HANDLER

** ADDR - ACCEPT ADDRESS VALUE WITH 'AD' PROMPT
*
* ENTRY, EXIT -- SEE 'DOPMT'

FC7E DF EE ADDR STX T1
FC80 BD 04 BSR OUTSTA
FC82 77 BD FCC HEXA+LTRD++80
FCB4 20 EF BRA DOPMT
```
** OUTSTA - OUTPUT STRING FOR ADDRESS PROMPT *

FC86  CE  C1  2F  OUTSTA  LDX  #DG2ADD
FC89  7E  FE  50  JMP  OUTST1

** DO - RESET USER FC AND RESUME *

ENTRY:  NONE
EXIT:  TO 'RESUME'
USERS:  ALL

FC8C  DE  F2  DO  LDX  USERS
FC8E  0B  INX
FC8F  0B  INX
FC90  0B  INX
FC91  0B  INX
FC92  0B  INX
FC93  0B  INX  X TO USER FC
FC94  8D  D9  BSR  DIGPNT

** RESUME - RESUME USER PROGRAM *

1) BLANKS ALL DISPLAYS
2) INITIALIZES (DIGADD)
3) STEPS USER CODE PAST BREAKPOINT
4) INSERTS BREAKPOINTS
5) PRINTS INSTRUCTION UPON RETURN
ENTRY: NONE
EXIT: (B) = 1
(X) = USERPC
USERS: ALL, TO, T1

FC96  BD  24  RESUME  BSR  REDIS  RESET DISPLAY
FC98  4F  CLR A
FC99  C6  06  LDA B  #6
FC9B  BD  FE  3A  RES1  JSR  OUTCH  CLEAR DISPLAYS
FC9E  5A  DEC B
FC9F  26  FA  BNE  RESI
FCA1  8D  19  BSR  REDIS  RESET DISPLAY
FCA3  BD  FE  6B  RES2  JSR  SSTEP  STEP PAST BREAKPOINT
FCA6  C6  04  LDA B  #NBR  SET BREAKPOINTS
FCA8  30  RES3  TSX
FCA9  EE  08  LDX  2*NBR,X  GET BREAKPOINT ADDRESS
FCAB  A6  00  LDA A  0,X
FCAE  36  PSH A
FCAF  36  PSH A
FCAF  86  3F  LDA A  #3F  REPLACE WITH SWI
FCB1  A7  00  STA A  0,X
FCB3  5A  DEC B
FCB4  26  F2  BNE  RESJ
FCB6  CE  FC  CE  LDX  #BKPT
FCB9  7E  FE  FC  JMP  SWIVE1  GO TO USER CODE

** REDIS - RESET DISPLAYS *

ENTRY: NONE
EXIT: DIGADD SET TO LEFTMOST DIGIT
USERS: TO

FCBC  DF  EC  REDIS  STX  TO
FCBE  CE  C1  6F  LDX  #DG6ADD
FCC1  DF  F0  STX  DIGADD
FCC3  DE  EC  LDX  TO
FCC5  39  RTS
** BADDR - BUILD ADDRESS
* 
* ENTRY: NONE
* EXIT (X) = ADDRESS

** RKPT - BREAK POINT RETURN
* 1) REMOVE BKPTS FROM USER CODE
* 2) CHECK FOR BREAKPOINT HIT AND EITHER
* A) RESUME IF NO HIT
* B) PRINT INSTRUCTION AND RETURN IF HIT

** NOW CLEAR BREAKPOINTS

** MEM - DISPLAY ADDRESS AND DATA
* 
* ENTRY: (X) = ADDRESS
* EXIT: (B) = 1
* USES: A,B,C,T0,T1

** AUTO - AUTO LOAD OF MEMORY
* 
* ENTRY: NONE
* EXIT: NO EXIT POSSIBLE
* USES: ALL,T0,T1
** EXAM - EXAMINE MEMORY

* ENTRY: NONE
* EXIT: (X) = ADDRESS
* (B) = 0
* (A) = 0
* USES: ALL TO T1

** FOWD - DISPLAY NEXT BYTE

* ENTRY: (X) = OLD ADDRESS
* EXIT: (X) = (XOLD) + 1
* (B) = 1
* (A) = 0
* USES: ALL TO

** BACK - DISPLAY PREVIOUS BYTE

* ENTRY: (X) = ADDRESS
* EXIT: (X) = (XOLD) + 1
* (B) = 1
* (A) = 0
* USES: ALL TO

** REPLAC - REPLACE DISPLAYED VALUE

* 'REPLAC' 1) BACKSPACES DISPLAY TO CANCEL DISPLAYED VALUE
* 2) SENDS PROMPT FOR REPLACEMENT VALUE
* 3) ACCEPTS AND REPLACES DESIGNATED BYTE(S)
* ENTRY: (X) = ADDRESS OF BYTE(S) TO REPLACE
* (B) = NUMBER OF BYTES
* (DIGADD) = ADDRESS OF DIGIT TO RIGHT OF DISPLAYED
* EXIT: B×X, DIGADD UNCHANGED
* USES: T0×A×C

** PROMPT - PROMPT AND INPUT BYTES

* ENTRY: (X) = ADDRESS TO STORE VALUE
* (B) = NUMBER OF BYTES
* (DIGADD) = ADDRESS OF FIRST ECHO CHARACTER
* EXIT: B×X UNCHANGED
* DIGADD UPDATED
* USES: T0, DIGADD
FD25 37  PROMPT  PSH B
FD26 86 08  LDA A  #DASH  PROMPT CHARACTER
FD28 5B  ASL B
FD29 BD FE 3A  PROM1  JSR OUTCH  SEND PROMPT
FD2C 5A  DEC B
FD2D 26 FA  BNE PROM1
FD2F 33  PUL B
FD30 BD 11  BSR BKSP  BACKSPACE DISPLAYS
FD32 37  PSH B  **ALTERNATE ENTRY**
FD33 BD FE 09  PROM2  JSR IHB  GET BYTE VALUE
FD36 A7 00  STA A  0,X  PLACE INTO MEMORY
FD38 08  INX  RUMP POINTER
FD39 5A  DEC B
FD3A 26 F7  BNE PROM2  MORE TO GO
FD3C 33  PUL B
FD3D 17  TBA  DUPLICATE
FD3F 09  DEX  FIX X
FD40 26 FC  BNE PROM3
FD42 39  RTS  EXIT

** BKSP - BACKSPACE DISPLAYS
**
* ENTRY: (B) = NUMBER DIGIT PAIRS TO BACKSPACE
* EXIT: (DIGADD) = (DIGADD) + 20 * (B)
* USES: A,C

FD43 37  BKSP  PSH B
FD44 96 F1  LDA A  DIGADD+1  L.S. BYTE
FD46 80 20  BKSP1  ADD A  #$20  BACKSPACE TWO PLACES
FD48 5A  DEC B
FD49 26 FB  BNE BKSP1
FD4A 97 F1  STA A  DIGADD+1
FD4C 33  PUL B
FD4E 39  RTS

** REGISTER DISPLAY FUNCTIONS
**
* ENTRY: NONE
* EXIT: (B) = NUMBER BYTES THIS REGISTER
* (X) = REGISTER ADDRESS ON STACK
* (DIGADD) INITIALIZE TO DIGIT 6
* USES: ALL,T0

FD4F BD 3B  REGX  BSR OUTSTJ  PRINT 'REGX'
FD51 30 95  FCC  LTRI,LTRN+$0
FD53 20 16  BRA REGXI
FD55 BD 35  REGA  BSR OUTSTJ  PRINT 'ACCA'
FD57 77 0B 0D  FCC  HEXA,LTRC,LTRC,LTR+$0
FD5B 20 16  BRA REGA1
FD5D BD 2B  REGB  BSR OUTSTJ  PRINT 'ACCB'
FD5F 77 0D 0D  FCC  HEXA,LTRC,LTRC,LTRB+$0
FD63 20 09  BRA REGB1
FD65 BD 25  REGP  BSR OUTSTJ  PRINT 'PC'
FD67 67 0D  FCC  LTRP,LTRC+$0
FD69 4C  INC A  (A) = OFFSET INTO STACK
FD6A 4C  INC A
FD6B 5C  REGX1 INC B
FD6C 4C  INC A
FD6D 4C  REGA1 INC A
FD6E 5C  REGB1 INC B
FD6F 88 02  ADD A  #$2
** DISPLAY - DISPLAY Indexed Bytes

ENTRY: \( (X) = \text{Address of Bytes to Output} \)

\( (B) = \text{Number of Bytes to Display} \)

EXIT: \( X, R \) Unchanged

\( (\text{DIGADD}) \) Updated

USES: All, To

** CONDX - DISPLAY Condition Codes

ENTRY: \( \text{DIGADD} \) Initialized

EXIT: \( (B) = 0 \)

USES: All, To

** STKPTR - OUTPUT User Stack Pointer

ENTRY: \( \text{(DIGADD) Initialized} \)

EXIT: \( (B) = 0 \)

USES: All, To
** ENCODE - SCAN AND ENCODE KEYBOARD **

ENTRY: NONE
EXIT: (A) = HEX VALUE OF KEY PRESSED
'C' SET FOR VALID CONDITION

USES: A,C,T0

FDBB 37 ENCODE PSH B
FDBC F& C0 03 LDA B COL1 GET KEYBOARD DATA
FDBD 86 C0 06 LDA A COL3
FDBE 48 ASL A LEFT JUSTIFY DATA
FDBF 48 ASL A
FDBG 48 ASL A
FDBH 48 ASL A
FDBA 48 ROL B
FDBB 48 ROL B
FDBD 48 ASL A
FDBE 48 ROL B
FDBF 48 ROL B
FDBG 37 PSH B
FDBH F& C0 05 LDA B COL2 GET LAST DATA
FDBA C4 1F AND B ***1F MASK ANY GARBAGE
FDBB 18 ADA
FDBC 33 PUL B
FDBC 43 COM A
FDBD 53 COM B

* (BA) IS NOW KEYBOARD PATTERN *

FDBA DF EC STX T0 TABLE OF POSSIBLE OUTPUTS
FDBD CE FF AC LDX #HEXTAB-1 FIND ACTIVE ACCUMULATOR
FDBF 0F 11 BRA ENC3 ILLEGAL OR NO KEY
FDBD 27 11 BEQ ENC1 A ACTIVE
FDBE 24 06 BCC ENC3 B ACTIVE
FDBF 36 PSH A INTERCHANGE B,A
FDBA 17 TRA
FDBD 33 PUL B
FDB1 CE FF AD LDX #HEXTAB17
FDB2 5D ENC1 TST B B SHOULD BE ZERO
FDB3 26 06 BNE ENC3 ILLEGAL
FDB4 08 ENC2 INX SCAN FOR ACTIVE BIT
FDB5 40 ASL A
FDB6 22 FC BHI ENC2 NOT ACTIVE BIT
FDB7 2B 01 BEQ ENC4 LEGAL CHARACTER
FDB8 0C ENC3 CLC ILLEGAL RETURNS 'C' CLEAR
FDB9 A6 00 ENC4 LDA A 0,X GET HEX FROM TABLE
FDBA DE EC LDX TO
FDBB 33 PUL B CLEAN UP
FDBC 39 RTS AND RETURN

** INCH - INPUT CHARACTER FROM KEYBOARD **

'C' INCH WAITS FOR A TRANSITION BETWEEN ILLEGAL AND
LEGAL KEYBOARD CONDITIONS, AND RETURNS HEX VALUE
OF KEY DEPRESSED

ENTRY: NONE
EXIT: (A) = HEX VALUE
USES: A,C,T0

FDF4 37 INCH PSH B
FDF5 C6 7F INC1 LDA B #TIME VIOLATION COUNT
FD67 8D C2  INC2  BSR  ENCODE  WAIT FOR ILLEGAL INTERVAL
FD69 25 FA  BCS  INC1  STILL LEGAL
FD6B 5A  DEC B
FD6C 26 F9  BNE  INC2  NOT A FELONY

*  NOW WE'RE SURE WE HAVE AN ILLEGAL CONDITION AND  
*  NOT JUST A RELEASE CONSTANT BOUNCE

FD6E C6 7F  INC3  LDA B  ##TIME  TIME UNTIL PAROLE
FE00 8D B9  INC4  BSR  ENCODE  BAD BEHAVIOR
FE02 24 FA  BCC  INC3  BACK IN THE SLAMMER
FE04 5A  DEC B
FE05 26 F9  BNE  INC4  BACK IN THE SLAMMER
FE07 33  PUL B
FE08 39  RTS

**  INH -- INPUT HEX BYTE AND DISPLAY ON LEDS  

*  ENTRY: NONE
*  EXIT: (A) = BYTE VALUE
*  (DIGADD) UPDATED
*  USES: A, T0, C

FE09 8D E9  INH  BSR  INCH  GET FIRST HALF
FE0B 8D 1B  BSR  OUTHEX  ECHO TO DISPLAYS
FE0D 48  ASL A
FE0E 48  ASL A
FE0F 48  ASL A
FE10 48  ASL A
FE11 37  PSH B
FE12 16  TAB
FE13 8D DF  BSR  INCH  GET NEXT HALF
FE15 8D 11  BSR  OUTHEX  ECHO
FE17 1B  ABA
FE18 33  PUL B
FE19 36  PSH A
FE1A 8D 9F  INH1  BSR  ENCODE  WAIT FOR KEY RELEASE
FE1C 25 FC  BCS  INH1
FE1E 32  PUL A  RESTORE LEGAL ENTRY
FE1F 39  RTS

**  OUTBYT -- OUTPUT TWO HEX DIGITS  

*  ENTRY: (A) = BYTE VALUE TO OUTPUT
*  EXIT: (DIGADD) UPDATED
*  USES: C, T0

FE20 36  OUTBYT  PSH A
FE21 44  LSR A
FE22 44  LSR A
FE23 44  LSR A
FE24 44  LSR A
FE25 8D 01  BSR  OUTHEX  OUTPUT M.S. FOUR BITS
FE27 32  PUL A

**  OUTHEX -- OUTPUT HEX DIGIT  

*  ENTRY: (A) = HEX VALUE
*  EXIT: (DIGADD) UPDATED
*  USES: C, T0

FE28 36  OUTHEX  PSH A
FE29 84 0F  AND A  ##F  MASK GARBAGE
FE2B DF EC  STX  TO
FE2D CE FF 95  LDX  #DISTAB-1  DISPLAY CODE TABLE
FE30 08  OUTH1  INX
FE31 4A  DEC A
FE32 2A FC  BPL  OUTH1
FE34 A6 00  LDA A  0*X  DISPLAY CODE FOR HEX
FE36 8D 04  BSR  OUT0  ALTERNATE ENTRY FOR 'OUTCH'
OUTCH - OUTPUT CHARACTER TO DISPLAY

** ENTRY: (A) = SEGMENT CODE
** (DIGADD) = ADDRESS OF DIGIT TO OUTPUT
** EXIT: (DIGADD) UPDATED
** USES: C, TO

OUTCH
STX TO
LDX DIGADD
**ALTERNATE ENTRY** FROM 'OUTCHEX'

DF EC
OUT0
LDX DIGADD

FE3A
FE3C
FE3E
FE3F
FE40
FE41
FE43
FE44
FE46
FE47
DEC B
FE48
BNE OUT1
FE4A
STX DIGADD
LDX TO
PUL B
FE4C
FE4E
RTS

OUTCH1
STX DIGADD
**ALTERNATE ENTRY** SETS UP DIGI

DF F0
OUTST1
TSX
FE50
FE52
FE53
FE55
FE56
FE57
FE59
FE5B
FE5C
FE5D
FE5F
JMP
FE60
FE62
FE64
FE66
FE6B

OUTSTR - OUTPUT EMBEDDED CHARACTER STRING

CALLING CONVENTION:
** JSR OUTSTR
** FIRST CHARACTER
**
** LAST CHARACTER (HAS D.F. LIK)
** NEXT INSTRUCTION
** ENTRY: NONE
** EXIT: TO 'NEXT INSTRUCTION'
** (A) = 0
** USES: A, *X, TO

OUTSTR
STX
FE50
FE52
FE53
FE55
FE56
FE57
FE59
FE5B
FE5C
FE5D
FE5F
JMP
FE60

OUTST2
TSX
LUX
INS
INS
LUX
LUX
OUTST3
LUX
INX
TST A
MPL OUTST3
CLR A
FE5D
FE5F
JMP
FE60

** STEP - STEP USER CODE
**
** ENTRY: NONE
** EXIT: (B) = 1
** (X) = USER P.C.
** (A) = 0
** USES: ALL, TO, T1

STEP
LUX
LUX
JMP
FE62
FE64
FE66
FE68

** SSTEP - PERFORM SINGLE STEP.
FE69 9F EE  SSTEP  STS  TEMP  WE'LL USE THIS WHEN WE RETURN
FE6D DE F2  LDX  USERS
FE6F A6 07  LDA A  7*X  PUSHING USER PC INTO MONITOR
FE71 36  PSH A  STACK
FE72 A6 06  LDA A  6*X
FE74 36  PSH A
FE75 EE 06  LDX  6*X  NOW GET USER PC INTO X
FE77 B6 3F  LDA A  #$3F  SWI'S ARE NORMAL EXIT FROM
FE79 36  PSH A  SCRATCHPAD EXECUTION
FE7A 36  PSH A
FE7B A6 02  LDA A  2*X  NOW WE ARE COPYING THREE BYTES
FE7D 36  PSH A  OF INSTRUCTION
FE7E A6 01  LDA A  1*X
FE80 36  PSH A
FE81 A6 00  LDA A  0*X  THIS IS THE OP CODE SO
FE83 36  BYTCNT  PSH A  SCRUTINIZE CAREFULLY
FE84 16  TAB
FE85 CE FF 75  LDX  $OPTAB-1
FE88 08  BYT1  INX
FE89 C0 08  SUB B  $8
FE8B 24 FB  BCC  BYT1
FE8D A6 00  LDA A  0*X
FE8F 46  BYT2  ROR A
FE90 5C  INC B
FE91 26 FC  BNE  BYT2
FE93 32  PUL A
FE94 36  PSH A
FE95 25 1E  BCS  BYT7
FE97 B1 30  CMP A  #$30  CHECK FOR BRANCH
FE99 24 04  BCC  BYT3
FE9B B1 20  CMP A  #$20
FE9D 24 14  BCC  BYT5  IT IS A BRANCH
FE9F B1 60  CMP A  #$60
FEAF B1 30  CMP A  #$30
FER1 C2 FF  BYT4  SBC B  #$FF
FER3 SC  BYT5  INC B
FER4 SC  BYT6  INC B
FER5 27 70  BYT7  BEO  BSTRD
FER7 30  TSX
FER8 25 02  BCS  STEP1
FERA E7 01  STA B  1*X  BRANCH OFFSET TO 2
FERB 36 01  STEP1  LDA A  $1
FERC C1 02  CMP B  $2
FERD 2E 06  BGT  STEP3
FERF 27 02  BEQ  STEP2  TWO BYTES
FEC4 A7 01  STA A  1*X  FOR ONE BYTES
FEC6 A7 02  STEP2  STA A  2*X  NOT FOR THREE BYTES
FEC8 4F  STEP3  CLR A  NOW ADD BYTE COUNT TO PC
FEC9 EB 06  ADD B  6*X
FECB A9 05  ADC A  5*X
FECF A7 05  STA A  5*X
FECF E7 06  STA B  6*X

*  DOES THE INSTRUCTION INVOLVE THE PC?  IF SO THEN IT
*  MUST BE INTERPRETED

FEF1 DE F2  SRCHOP  LDX  USERS
FEF3 A7 06  STA A  6*X
FEF5 E7 07  STA B  7*X  UPDATE PC ON USER STACK
FEF7 C6 06  LDA B  $6
FEF9 32  PUL A
FEFA 36  PSH A
FE8C 84 CF  AND A $04CF  IS THIS A SUBROUTINE CALL?
FE8D 81 8D  CMP A $068D
FE8E 32  PUL A
FE8F 27 48  BEQ BSRH
FE90 27 6E  CMP A $066E
FE91 27 5B  BEQ JPXH  IT IS INDEXED JUMP
FE92 81 7E  CMP A $076E
FE93 27 5E  BEQ JMPH  IT IS EXTENDED JUMP
FE94 81 39  CMP A $039E
FE95 27 62  BEQ RTSH  IT IS RTS
FE96 81 3B  CMP A $03BE
FE97 27 6C  BEQ RTIH  IT IS RTI
FE98 81 3F  CMP A $03EF
FE99 27 6E  BEQ SWIH  IT IS SWI
FE9A AF 06  STS $06,X  AIM USER PC AT SCRATCH AREA
FE9B 36  PSH A  REPLACE OPCODE
FE9C CE FF 05  LDX #$SRRET

** SWIVE1 - SET UP BREAKPOINT RETURN AND JUMP TO USER CODE
* 
* ENTRY: (X) = SWI VECTOR
* EXIT: TO USER PROGRAM

FEFE 86 7E  LDA A $07E  JUMP OF CODE
FEEF 97 F4  STA A SYSSWI
FEF0 DF 05  STX SYSSWI+1
FEF2 9E F2  LDS USERS
FEF4 3B  RTI

* THE FOLLOWING CODE IS EXECUTED AFTER A SINGLE STEP
* OF AN OUT-OF-PLACE INSTRUCTION. NOW CHECK TO SEE
* IF BRANCH OCCURRED, MODIFY THE USER PC ACCORDINGLY

FEF5 30  LDX $5,X  GET SWI HIT LOCATION INTO X
FEF6 EE 05  LDS $5,X
FEF8 08  INX
FEF9 4F  CLR A
FEF0 5F  CLR B
FEF1 9C EE  CPX TEMP
FEF2 26 0C  BNE BCHNTK

* ADD THE BRANCH OFFSET TO THE USER PC

FEF3 09  DEX  X WILL NOW POINT AT USERPC
FEF4 EE 00  LDS $0,X  SAVED VALUE OF PC INTO X
FEF5 09  DEX  PREPARE TO FETCH BRANCH OFFSET
FEF6 E6 00  LDA B $0,X
FEF7 2A 01  BPL PLUS
FEF8 43  COM A  A IS SIGN EXTENSION OF B
FEF9 30  PLUS TSX  LO COST WAY TO POINT TO USERPC
FEFA EE 05  LDS $5,X
FEFB EB 01  BCHNTK ADD B $1,X  ADD BRANCH OFFSET OR ZERO TO PC
FEFC A9 60  ADD A $0,X
FEFD 30  TSX
FEFE A7 05  STA A $5,X
FEFF 27 64  STA B $6,X
FEF0 3F  DEX  PLACE NEW USERPC ONTO STACK
FEF1 DF 02  STOX STX USERS
FEF2 9E EE  BSTRD LDS TEMP
FEF3 39  RTS  RETURN TO CALLING ROUTINE

** SPECIAL HANDLERS
** JSR HANDLER **

| FF30 00 3F | JSRH | SRR A | F3F | JSR'S TO JUMPS |
| FF32 36 | FSH A | DEX | CORRECTED OPCODE ONTO STACK |
| FF33 09 | DEX | |
| FF35 DF F2 | STX | USERS |
| FF37 A6 03 | JSRH1 | LDA A | 3|X |
| FF39 A7 01 | STA A | 1|X |
| FF3B 08 | INX | |
| FF3C 5A | DEC B | |
| FF3D 2A F8 | BPL | JSRH1 |
| FF3F 20 90 | BRA | SRCHOP |

** JFXH - INDEXED JUMP HANDLER **

| FF41 33 | JPXH | PUL B | GET OFFSET |
| FF42 4F | | CLR A |
| FF43 EB 05 | ADD B | 5|X |
| FF45 A9 04 | ADC A | 4|X |
| FF47 8C | FCB | 8C |

** JMP HANDLER **

| FF48 32 | JMPH | PUL A |
| FF49 33 | PUL B |
| FF4A A7 06 | NEWFC | STA A | 6|X |
| FF4C E7 07 | STA B | 7|X |
| FF4E 20 05 | BRA | STOX |

** RTS HANDLER **

| FF50 08 | RTSH | INX |
| FF51 09 | INX |
| FF52 DF F2 | STX | USERS |
| FF54 A6 03 | RTS1 | LDA A | 3|X |
| FF56 A7 05 | STA A | 5|X |
| FF58 09 | DEX |
| FF5A 2E F8 | BGT | RTS1 |
| FF5C 20 C9 | BRA | BSTD |

** RTI HANDLER **

| FF5E 08 | RTIH | INX |
| FF5F 5A | DEC B |
| FF60 2A FC | BPL | RTIH |
| FF62 20 C1 | BRA | STOX |

** SWI HANDLER **

| FF64 A6 07 | SWIH | LDA A | 7|X |
| FF66 A7 00 | STA A | 0|X |
| FF68 09 | DEX |
| FF69 5A | DEC B |
| FF6A 2A F8 | BPL | SWIH |
| FF6C 8A 10 | ORA A | $00010000 |
| FF6E A7 01 | STA A | 1|X |
| FF70 C6 FA | LDA B | $USWI/256|USWI |
| FF72 B6 00 | LDA A | $USWI/256 |
| FF74 20 D4 | BRA | NEWFC |

PATCH IN UIRG
** OPTAB - LEGAL OP-CODE LOOKUP TABLE

FF76 9C 00 3C OPTAB FDR $9C00,$3CAF,$4000,$00AC,$6412,$6412,$6410,$6410
FF86 11 01 10 FDR $1101,$1004,$1000,$1000,$110D,$100C,$100C,$100C

** HEX DISPLAY CODE TABLE

FF96 7E 30 6D DISTAB FCC HEX0,HEX1,HEX2,HEX3,HEX4,HEX5,HEX6,HEX7
FF9E 7F 7B 77 FCC HEX8,HEX9,HEXA,HEXB,HEXC,HEXD,HEXE,HEXF

** KEY VALUE TABLE

FFA6 07 0A 0D HEXTAB FCC 7,10,13,2,5,8,11,14
FFAE 03 06 09 FCC 3,6,9,12,15,0,1,4

** COMMAND HANDLER ENTRY POINT TABLE

FFB6 FC 45 FD CMDBL FDB ZERO,REAG,REGB,REGP,REGX,CONDX,REGS,RESUME,STEP
FFC8 FC 46 FD FDB BKSET,AUTO,BACK,REPLAC,B0,EXAM,F0WD

FFF8 ORG $FFF8

** INTERRUPT VECTORS.

FFF8 00 F7 FDR UIRO USER IRQ HANDLER
FFFA 00 F4 FDB SYSSWI SYSTEM SWI HANDLER
FFFC 00 FD FDB UNMI USER NMI HANDLER
FFFE FC 00 FDR RESET
0000 END

STATEMENTS =970
FREE BYTES =24298
NO ERRORS DETECTED
**Symbolic Reference Table**

<table>
<thead>
<tr>
<th>ADDR</th>
<th>FCTE</th>
<th>DIGCCD</th>
<th>G070</th>
<th>WM</th>
<th>F09</th>
<th>RD01</th>
<th>F073</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO</td>
<td>F00A</td>
<td>DIGSAB</td>
<td>F096</td>
<td>W012</td>
<td>F027</td>
<td>REPLAC</td>
<td>F018</td>
</tr>
<tr>
<td>AUT1</td>
<td>F0CC</td>
<td>DIS1</td>
<td>F07C</td>
<td>W01N</td>
<td>F01E</td>
<td>REPLI</td>
<td>F024</td>
</tr>
<tr>
<td>BACK</td>
<td>F01B</td>
<td>DIS2</td>
<td>F097</td>
<td>NEWPC</td>
<td>F04A</td>
<td>RESSET</td>
<td>F00D</td>
</tr>
<tr>
<td>BADDR</td>
<td>F0C1</td>
<td>DU</td>
<td>F0C1</td>
<td>UPTAB</td>
<td>F076</td>
<td>RESEL</td>
<td>F010</td>
</tr>
<tr>
<td>BHEMTK</td>
<td>F1B</td>
<td>DOWMT</td>
<td>F0EF</td>
<td>OUTST3</td>
<td>F025</td>
<td>RESUME</td>
<td>F006</td>
</tr>
<tr>
<td>BKPT</td>
<td>F0CE</td>
<td>DOWMI</td>
<td>F075</td>
<td>OUTCH</td>
<td>F03A</td>
<td>RES1</td>
<td>F09B</td>
</tr>
<tr>
<td>BKP1</td>
<td>F0D7</td>
<td>DISPLAY</td>
<td>F078</td>
<td>OUTHEX</td>
<td>F013</td>
<td>RES2</td>
<td>F0A3</td>
</tr>
<tr>
<td>BKP2</td>
<td>F0E1</td>
<td>ENCOD</td>
<td>F088</td>
<td>OUTHI</td>
<td>F020</td>
<td>RES3</td>
<td>F085</td>
</tr>
<tr>
<td>BKP3</td>
<td>F0E6</td>
<td>ENC1</td>
<td>F064</td>
<td>OUTSTA</td>
<td>F06E</td>
<td>RT1H</td>
<td>F096</td>
</tr>
<tr>
<td>BKP4</td>
<td>F0C0</td>
<td>ENC2</td>
<td>F067</td>
<td>OUTSTJ</td>
<td>F06C</td>
<td>RTSH</td>
<td>F09D</td>
</tr>
<tr>
<td>BKB5</td>
<td>F046</td>
<td>ENC3</td>
<td>F0ED</td>
<td>OUTSTR</td>
<td>F052</td>
<td>RT51</td>
<td>F054</td>
</tr>
<tr>
<td>BKB6</td>
<td>F040</td>
<td>ENC4</td>
<td>F0EE</td>
<td>OUTSTO</td>
<td>F06D</td>
<td>SRHOE</td>
<td>F061</td>
</tr>
<tr>
<td>BKB7</td>
<td>F055</td>
<td>EXAM</td>
<td>F013</td>
<td>OUTST1</td>
<td>F050</td>
<td>STYSET</td>
<td>F025</td>
</tr>
<tr>
<td>BKB8</td>
<td>F055</td>
<td>EXAM1</td>
<td>F016</td>
<td>OUTST3</td>
<td>F037</td>
<td>SSTEP</td>
<td>F068</td>
</tr>
<tr>
<td>BKB9</td>
<td>F043</td>
<td>HEXTAB</td>
<td>F0A6</td>
<td>OUTO</td>
<td>F02C</td>
<td>STEP</td>
<td>F062</td>
</tr>
<tr>
<td>BKB1</td>
<td>F046</td>
<td>IHB</td>
<td>F099</td>
<td>OUT1</td>
<td>F043</td>
<td>STEP1</td>
<td>F02C</td>
</tr>
<tr>
<td>BKB2</td>
<td>F02A</td>
<td>INB1</td>
<td>F0A1</td>
<td>F00S</td>
<td>F018</td>
<td>STEP2</td>
<td>F006</td>
</tr>
<tr>
<td>BSTR</td>
<td>F027</td>
<td>INCH</td>
<td>F074</td>
<td>PROMPT</td>
<td>F025</td>
<td>STEP3</td>
<td>F00B</td>
</tr>
<tr>
<td>BYNTN</td>
<td>F05</td>
<td>INC1</td>
<td>F075</td>
<td>PROM1</td>
<td>F029</td>
<td>STRPTR</td>
<td>F0A5</td>
</tr>
<tr>
<td>BYT1</td>
<td>F0E8</td>
<td>INC2</td>
<td>F077</td>
<td>PROM2</td>
<td>F033</td>
<td>STGRK</td>
<td>F023</td>
</tr>
<tr>
<td>BYT2</td>
<td>F08</td>
<td>INC3</td>
<td>F07E</td>
<td>PROM3</td>
<td>F03E</td>
<td>SWK</td>
<td>F064</td>
</tr>
<tr>
<td>BYT3</td>
<td>F08F</td>
<td>INC4</td>
<td>F0ED</td>
<td>REDIS</td>
<td>F0BC</td>
<td>SWYVEI</td>
<td>F0FC</td>
</tr>
<tr>
<td>BYT4</td>
<td>F0BE</td>
<td>JWFIH</td>
<td>F04B</td>
<td>REGA</td>
<td>F055</td>
<td>SYSSWI</td>
<td>F0F4</td>
</tr>
<tr>
<td>BYT5</td>
<td>F0F1</td>
<td>JFHF</td>
<td>F041</td>
<td>REGA1</td>
<td>F06D</td>
<td>ZERO</td>
<td>F0C5</td>
</tr>
<tr>
<td>BYT6</td>
<td>F044</td>
<td>JSRH</td>
<td>F030</td>
<td>REGB</td>
<td>F05D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BYT7</td>
<td>F085</td>
<td>JSRH1</td>
<td>F037</td>
<td>REGB1</td>
<td>F05E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMDTAB</td>
<td>F0E6</td>
<td>MAIN</td>
<td>F019</td>
<td>REGB</td>
<td>F065</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONV</td>
<td>F053</td>
<td>MAINI</td>
<td>F021</td>
<td>REGS</td>
<td>F048</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONO</td>
<td>F059</td>
<td>MAIN2</td>
<td>F051</td>
<td>REDX</td>
<td>F0DF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CON1</td>
<td>F0E3</td>
<td>MAIN3</td>
<td>F036</td>
<td>REDX1</td>
<td>F06B</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Assembly Constant Table**

**Keyboard Locations**

**Display Addresses**

<table>
<thead>
<tr>
<th>COL1</th>
<th>$0003</th>
<th>Rightmost Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>COL2</td>
<td>$0005</td>
<td>Leftmost Column</td>
</tr>
<tr>
<td>COL3</td>
<td>$0006</td>
<td>Misc. Constants</td>
</tr>
</tbody>
</table>

**TIME**

| NBR | 32 | Number Breakpoints |

**Displayed Character Segment Codes**

<table>
<thead>
<tr>
<th>HEX0</th>
<th>$78</th>
<th>HEX8</th>
<th>$7F</th>
<th>LTRA</th>
<th>$7D</th>
<th>LTRH</th>
<th>$37</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEX1</td>
<td>$30</td>
<td>HEX9</td>
<td>$7B</td>
<td>LTRB</td>
<td>$1F</td>
<td>LTRC</td>
<td>$1D</td>
</tr>
<tr>
<td>HEX2</td>
<td>$60</td>
<td>HEXA</td>
<td>$77</td>
<td>LTRD</td>
<td>$0D</td>
<td>LTRD</td>
<td>$3D</td>
</tr>
<tr>
<td>HEX3</td>
<td>$79</td>
<td>HEXB</td>
<td>$1F</td>
<td>LTRF</td>
<td>$47</td>
<td>LTRG</td>
<td>$05</td>
</tr>
<tr>
<td>HEX4</td>
<td>$33</td>
<td>HEXC</td>
<td>$4E</td>
<td>LTRH</td>
<td>$1B</td>
<td>LTRI</td>
<td>$3E</td>
</tr>
<tr>
<td>HEX5</td>
<td>$65</td>
<td>HEXD</td>
<td>$30</td>
<td>LTRJ</td>
<td>$30</td>
<td>LTRJ</td>
<td>$38</td>
</tr>
<tr>
<td>HEX6</td>
<td>$5F</td>
<td>HEXE</td>
<td>$4F</td>
<td>LTRK</td>
<td>$57</td>
<td>LTRK</td>
<td>$5B</td>
</tr>
<tr>
<td>HEX7</td>
<td>$7C</td>
<td>HEXF</td>
<td>$47</td>
<td>LTRL</td>
<td>$0E</td>
<td>DASH</td>
<td>$0B</td>
</tr>
</tbody>
</table>

**Reserved Memory Bytes in RAM**

<table>
<thead>
<tr>
<th>OOD9</th>
<th>USRSTK</th>
<th>EQU</th>
<th>*-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>OOD1</td>
<td>RMB 19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OOD3</td>
<td>MONSTK</td>
<td>EQU</td>
<td>*-1</td>
</tr>
<tr>
<td>OOD4</td>
<td>RMTASL</td>
<td>RMB 2*NBR</td>
<td></td>
</tr>
<tr>
<td>OOD5</td>
<td>TD</td>
<td>RMB 2</td>
<td></td>
</tr>
<tr>
<td>OOD6</td>
<td>TEMP</td>
<td>RMB 2</td>
<td></td>
</tr>
<tr>
<td>OOD7</td>
<td>DIGADD</td>
<td>RMB 2</td>
<td></td>
</tr>
<tr>
<td>OOD8</td>
<td>USERS</td>
<td>RMB 2</td>
<td></td>
</tr>
<tr>
<td>OOD9</td>
<td>TI</td>
<td>EQU TEMP</td>
<td></td>
</tr>
<tr>
<td>OODA</td>
<td>USWLI</td>
<td>RMB 3</td>
<td></td>
</tr>
<tr>
<td>OODB</td>
<td>USWI</td>
<td>RMB 3</td>
<td></td>
</tr>
<tr>
<td>OODC</td>
<td>USWI</td>
<td>RMB 3</td>
<td></td>
</tr>
</tbody>
</table>

**Temporary**

- JUDC by single stepper
- USER STACK POINTER
- SYSTEM SWI VECTOR
- USER IRQ VECTOR
- USER SWI VECTOR
- USER IRQ VECTOR
MEMORY

Memory Map

The memory is organized as shown below.

<table>
<thead>
<tr>
<th>Monitor ROM</th>
<th>FFFF FC00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not usable</td>
<td>C1FF C170</td>
</tr>
<tr>
<td>Displays</td>
<td>C16F C110</td>
</tr>
<tr>
<td>Not usable</td>
<td>C10F C00F</td>
</tr>
<tr>
<td>Keyboard</td>
<td>C00E C003</td>
</tr>
<tr>
<td>Not usable</td>
<td>C002 C000</td>
</tr>
</tbody>
</table>

Optional 256 bytes of user RAM 01FF 0100
59 bytes RAM (reserved for monitor) 00FF 00C5
197 bytes of user RAM 00C4 0000

Memory Decoding

<table>
<thead>
<tr>
<th>A15</th>
<th>A14</th>
<th>A13</th>
<th>A12</th>
<th>A11</th>
<th>A10</th>
<th>A9</th>
<th>A8</th>
<th>A7</th>
<th>A6</th>
<th>A5</th>
<th>A4</th>
<th>A3</th>
<th>A2</th>
<th>A1</th>
<th>A0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM IC12</td>
<td>FFXX FCxx</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RAM (Optional) IC16, IC17</td>
<td>01XX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RAM IC14, IC15</td>
<td>00XX</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>KEYBOARD C0 - X</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DISPLAYS C1XX</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>—</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>—</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

1 = LOGIC 1, 0 = LOGIC 0, — = DOES NOT CARE, X = FUNCTIONING ADDRESS

* Not fully decoded.
Keyboard And Display Functioning Addresses

**KEYBOARD**

<table>
<thead>
<tr>
<th>Keys</th>
<th>BINARY</th>
<th></th>
<th></th>
<th></th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>3, 6, 9, C, F</td>
<td>—</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2, 5, 8, B, E</td>
<td>—</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>0, 1, 4, 7, A, D</td>
<td>—</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>

**DISPLAY**

<table>
<thead>
<tr>
<th>LED</th>
<th>BINARY</th>
<th></th>
<th></th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>I</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>N</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Z</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>V</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**LED SEGMENTS**

<table>
<thead>
<tr>
<th>LED SEGMENT</th>
<th>BINARY</th>
<th></th>
<th></th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>—</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>—</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>c</td>
<td>—</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d</td>
<td>—</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>e</td>
<td>—</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>f</td>
<td>—</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>g</td>
<td>—</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DP</td>
<td>—</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

— = DOES NOT CARE
### INSTRUCTION SET

#### ACCUMULATOR AND MEMORY

<table>
<thead>
<tr>
<th>OPERATIONS</th>
<th>ADDRESSES</th>
<th>IMMED</th>
<th>DIRECT</th>
<th>INDEX</th>
<th>EXTND</th>
<th>INNER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>ADDA</td>
<td>B9 2 2</td>
<td>98 2 2</td>
<td>A8 5 2</td>
<td>B8 4 3</td>
<td>A + M - A</td>
</tr>
<tr>
<td>Add</td>
<td>ADDB</td>
<td>C8 2 2</td>
<td>08 3 2</td>
<td>E8 5 2</td>
<td>F8 4 3</td>
<td>B + M - B</td>
</tr>
<tr>
<td>Add Acmtrs</td>
<td>ABA</td>
<td>18 2 2</td>
<td>B8 2 2</td>
<td>A - B - A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add with Carry</td>
<td>ADCA</td>
<td>B9 2 2</td>
<td>95 2 3</td>
<td>A5 5 2</td>
<td>B5 4 3</td>
<td>A + M + C - A</td>
</tr>
<tr>
<td>And</td>
<td>ANDA</td>
<td>B4 2 2</td>
<td>94 3 2</td>
<td>A4 5 2</td>
<td>B4 4 3</td>
<td>A + M - A</td>
</tr>
<tr>
<td>Bit Test</td>
<td>BITA</td>
<td>B5 2 2</td>
<td>95 3 2</td>
<td>A5 5 2</td>
<td>B5 4 3</td>
<td>A + M</td>
</tr>
<tr>
<td>Clear</td>
<td>CLR</td>
<td>6F 7 2</td>
<td>1F 6 3</td>
<td>00 - A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compare</td>
<td>CMPA</td>
<td>B1 2 2</td>
<td>91 3 2</td>
<td>A1 5 2</td>
<td>B1 4 3</td>
<td>A - M</td>
</tr>
<tr>
<td>Compare Acmtrs</td>
<td>CMPE</td>
<td>C1 2 2</td>
<td>D1 3 2</td>
<td>E1 5 2</td>
<td>F1 4 3</td>
<td>B - M</td>
</tr>
<tr>
<td>Complement, 1's</td>
<td>COM</td>
<td>63 7 2</td>
<td>73 6 3</td>
<td>M - M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complement, 2's</td>
<td>COMA</td>
<td>63 2 2</td>
<td>1 - B - B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load Acmtrs</td>
<td>LOA</td>
<td>B8 2 2</td>
<td>96 3 2</td>
<td>A6 5 2</td>
<td>B6 4 3</td>
<td>M - A</td>
</tr>
<tr>
<td>Or, Inclusive</td>
<td>ORA</td>
<td>B8 2 2</td>
<td>9A 3 2</td>
<td>A2 5 2</td>
<td>B8 4 3</td>
<td>A + M - A</td>
</tr>
<tr>
<td>Push Data</td>
<td>PSHA</td>
<td>35 4 1</td>
<td>A - Msp, SP - 1 - SP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pull Data</td>
<td>PUL</td>
<td>32 4 1</td>
<td>SP - Msp, SP + A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rotate Left</td>
<td>ROL</td>
<td>69 7 2</td>
<td>79 6 3</td>
<td>M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rotate Right</td>
<td>ROR</td>
<td>66 7 2</td>
<td>76 6 3</td>
<td>M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shift Left, Arithmetic</td>
<td>ASL</td>
<td>68 7 2</td>
<td>78 6 3</td>
<td>M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shift Right, Arithmetic</td>
<td>ASR</td>
<td>67 7 2</td>
<td>77 6 3</td>
<td>M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shift Right, Logic</td>
<td>LSR</td>
<td>64 7 2</td>
<td>74 6 3</td>
<td>M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Store Acmtrs</td>
<td>STA</td>
<td>97 4 2</td>
<td>A7 6 2</td>
<td>B7 5 3</td>
<td>A + M</td>
<td></td>
</tr>
<tr>
<td>Subtract</td>
<td>SUBA</td>
<td>A0 2 2</td>
<td>E7 6 2</td>
<td>F7 5 3</td>
<td>B - M</td>
<td></td>
</tr>
<tr>
<td>Subtract Acmtrs</td>
<td>SBA</td>
<td>10 2 2</td>
<td>A - B - A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subtract with Carry</td>
<td>SBCA</td>
<td>A2 2 2</td>
<td>92 3 2</td>
<td>A2 5 2</td>
<td>B2 4 3</td>
<td>A - M + C - A</td>
</tr>
<tr>
<td>Transfer Acmtrs</td>
<td>TSB</td>
<td>6A 2 2</td>
<td>02 3 2</td>
<td>E2 5 2</td>
<td>F2 4 3</td>
<td>B - M + C - B</td>
</tr>
<tr>
<td>Test, Zero or Minus</td>
<td>TST</td>
<td>60 7 2</td>
<td>70 6 3</td>
<td>00 - M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSTA</td>
<td>4D 2 2</td>
<td>A - 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### BOOLEAN/ARITHMETIC OPERATION

<table>
<thead>
<tr>
<th>Processor Code Reg.</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>H</td>
<td>N</td>
<td>Z</td>
<td>V</td>
<td>C</td>
</tr>
</tbody>
</table>

*Copied with permission of Motorola.*
### INDEX REGISTER AND STACK

<table>
<thead>
<tr>
<th>Pointer Operations</th>
<th>Mnemonic</th>
<th>OP</th>
<th>#</th>
<th>OP</th>
<th>#</th>
<th>OP</th>
<th>#</th>
<th>OP</th>
<th>#</th>
<th>Boolean/Aithmetic Operation</th>
<th>M</th>
<th>N</th>
<th>Z</th>
<th>V</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compare Index Reg</td>
<td>CPX</td>
<td>BC</td>
<td>3</td>
<td>3</td>
<td>BC</td>
<td>4</td>
<td>2</td>
<td>AC</td>
<td>6</td>
<td>2</td>
<td>BC</td>
<td>5</td>
<td>3</td>
<td>INX*H: (L + M + 1)</td>
<td>0</td>
</tr>
<tr>
<td>Decrement Index Reg</td>
<td>DEX</td>
<td>DE</td>
<td>3</td>
<td>3</td>
<td>DE</td>
<td>4</td>
<td>2</td>
<td>EE</td>
<td>6</td>
<td>2</td>
<td>FE</td>
<td>5</td>
<td>3</td>
<td>X ← X</td>
<td>0</td>
</tr>
<tr>
<td>Decrement Stack Ptr</td>
<td>DES</td>
<td>CE</td>
<td>3</td>
<td>3</td>
<td>CE</td>
<td>4</td>
<td>2</td>
<td>EE</td>
<td>6</td>
<td>2</td>
<td>FE</td>
<td>5</td>
<td>3</td>
<td>SP ← SP</td>
<td>0</td>
</tr>
<tr>
<td>Increment Index Reg</td>
<td>INX</td>
<td>BE</td>
<td>3</td>
<td>3</td>
<td>BE</td>
<td>4</td>
<td>2</td>
<td>AE</td>
<td>6</td>
<td>2</td>
<td>BE</td>
<td>5</td>
<td>3</td>
<td>M ← M + 1</td>
<td>0</td>
</tr>
<tr>
<td>Increment Stack Ptr</td>
<td>INS</td>
<td>BX</td>
<td>3</td>
<td>3</td>
<td>BX</td>
<td>4</td>
<td>2</td>
<td>BF</td>
<td>6</td>
<td>2</td>
<td>FF</td>
<td>6</td>
<td>3</td>
<td>SP ← SP</td>
<td>0</td>
</tr>
<tr>
<td>Load Index Reg</td>
<td>LDX</td>
<td>BX</td>
<td>3</td>
<td>3</td>
<td>BX</td>
<td>4</td>
<td>2</td>
<td>AF</td>
<td>7</td>
<td>2</td>
<td>BF</td>
<td>6</td>
<td>3</td>
<td>X ← X</td>
<td>0</td>
</tr>
<tr>
<td>Load Stack Pointer</td>
<td>LDS</td>
<td>BX</td>
<td>3</td>
<td>3</td>
<td>BX</td>
<td>4</td>
<td>2</td>
<td>AF</td>
<td>7</td>
<td>2</td>
<td>BF</td>
<td>6</td>
<td>3</td>
<td>SP ← SP</td>
<td>0</td>
</tr>
<tr>
<td>Store Index Reg</td>
<td>STX</td>
<td>OF</td>
<td>5</td>
<td>2</td>
<td>EF</td>
<td>7</td>
<td>2</td>
<td>FF</td>
<td>6</td>
<td>3</td>
<td>OF</td>
<td>5</td>
<td>2</td>
<td>X ← X</td>
<td>0</td>
</tr>
<tr>
<td>Store Stack Pointer</td>
<td>STS</td>
<td>OF</td>
<td>5</td>
<td>2</td>
<td>EF</td>
<td>7</td>
<td>2</td>
<td>FF</td>
<td>6</td>
<td>3</td>
<td>OF</td>
<td>5</td>
<td>2</td>
<td>SP ← SP</td>
<td>0</td>
</tr>
<tr>
<td>Index Reg ← Stack Pointer</td>
<td>TXS</td>
<td>34</td>
<td>4</td>
<td>1</td>
<td>X ← X</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stack Pointer ← Index Reg</td>
<td>TSX</td>
<td>34</td>
<td>4</td>
<td>1</td>
<td>X ← X</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### JUMP AND BRANCH

<table>
<thead>
<tr>
<th>Operations</th>
<th>Mnemonic</th>
<th>OP</th>
<th>#</th>
<th>OP</th>
<th>#</th>
<th>OP</th>
<th>#</th>
<th>OP</th>
<th>#</th>
<th>BRANCH TEST</th>
<th>M</th>
<th>N</th>
<th>Z</th>
<th>V</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch Always</td>
<td>BRA</td>
<td>20</td>
<td>4</td>
<td>2</td>
<td>None</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Branch if Carry Cleared</td>
<td>BCC</td>
<td>24</td>
<td>4</td>
<td>2</td>
<td>C ← 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Branch if Carry Set</td>
<td>BCS</td>
<td>25</td>
<td>4</td>
<td>2</td>
<td>C ← 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Branch if Zero</td>
<td>BEO</td>
<td>27</td>
<td>4</td>
<td>2</td>
<td>Z ← 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Branch if Zero, Carry Cleared</td>
<td>BEQ</td>
<td>28</td>
<td>4</td>
<td>2</td>
<td>C ← 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Branch if Zero, Carry Cleared</td>
<td>BEQ</td>
<td>28</td>
<td>4</td>
<td>2</td>
<td>C ← 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Branch if Higher</td>
<td>BHI</td>
<td>32</td>
<td>4</td>
<td>2</td>
<td>C ← Z</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Branch if Higher, Carry Cleared</td>
<td>BHL</td>
<td>32</td>
<td>4</td>
<td>2</td>
<td>C ← Z</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Branch if Lower or Same</td>
<td>BLS</td>
<td>33</td>
<td>4</td>
<td>2</td>
<td>C ← Z</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Branch if Lower or Same</td>
<td>BLO</td>
<td>33</td>
<td>4</td>
<td>2</td>
<td>C ← Z</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Branch if Lower or Same</td>
<td>BLS</td>
<td>33</td>
<td>4</td>
<td>2</td>
<td>C ← Z</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Branch if Lower or Same</td>
<td>BLS</td>
<td>33</td>
<td>4</td>
<td>2</td>
<td>C ← Z</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### CONDITIONS CODE REGISTER

<table>
<thead>
<tr>
<th>Operations</th>
<th>Mnemonic</th>
<th>OP</th>
<th>#</th>
<th>OP</th>
<th>#</th>
<th>Boolean Operation</th>
<th>M</th>
<th>N</th>
<th>Z</th>
<th>V</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear Carry</td>
<td>CLC</td>
<td>BC</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>C ← 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Clear Interrupt Mask</td>
<td>CLI</td>
<td>OE</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>C ← 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Clear Overflow</td>
<td>CLV</td>
<td>OA</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>V ← 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Set Carry</td>
<td>SEC</td>
<td>SE</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>C ← 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Set Interrupt Mask</td>
<td>SEI</td>
<td>OF</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>S ← 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Set Overflow</td>
<td>SEV</td>
<td>OF</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>V ← 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Acctv A: CCR</td>
<td>TAP</td>
<td>OE</td>
<td>2</td>
<td>1</td>
<td>A ← CCR</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CCR: Acctv A</td>
<td>FPA</td>
<td>OE</td>
<td>2</td>
<td>1</td>
<td>CCR ← A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

### CONDITION CODE REGISTER NOTES

<table>
<thead>
<tr>
<th>Bit Set</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Bit set if test is true and cleared otherwise</td>
</tr>
<tr>
<td>1</td>
<td>Bit set if test is false and cleared otherwise</td>
</tr>
<tr>
<td>2</td>
<td>Bit set if test is true and cleared otherwise</td>
</tr>
<tr>
<td>3</td>
<td>Bit set if test is false and cleared otherwise</td>
</tr>
<tr>
<td>4</td>
<td>Bit set if test is true and cleared otherwise</td>
</tr>
<tr>
<td>5</td>
<td>Bit set if test is false and cleared otherwise</td>
</tr>
<tr>
<td>6</td>
<td>Bit set if test is true and cleared otherwise</td>
</tr>
<tr>
<td>7</td>
<td>Bit set if test is false and cleared otherwise</td>
</tr>
<tr>
<td>8</td>
<td>Bit set if test is true and cleared otherwise</td>
</tr>
<tr>
<td>9</td>
<td>Bit set if test is false and cleared otherwise</td>
</tr>
<tr>
<td>10</td>
<td>Bit set if test is true and cleared otherwise</td>
</tr>
<tr>
<td>11</td>
<td>Bit set if test is false and cleared otherwise</td>
</tr>
<tr>
<td>12</td>
<td>Bit set if test is true and cleared otherwise</td>
</tr>
<tr>
<td>13</td>
<td>Bit set if test is false and cleared otherwise</td>
</tr>
<tr>
<td>14</td>
<td>Bit set if test is true and cleared otherwise</td>
</tr>
<tr>
<td>15</td>
<td>Bit set if test is false and cleared otherwise</td>
</tr>
</tbody>
</table>

### LEGEND

- **OP**: Operation Code (Hexadecimal)
- **H**: Height of cell from bit 3
- **N**: Negative (sign bit)
- **Z**: Zero (byte)
- **V**: Overflow (2's complement)
- **M**: Carry from bit 1
- **MS**: Most Significant
- **MS**: Most Significant
- **LS**: Least Significant
IN CASE OF DIFFICULTY

**WARNING:** Dangerous AC voltage is present inside the cabinet (where the fuse is located) when the line cord is plugged in. See Pictorial 9-1.

This section of the Manual is divided into three parts: “Visual Checks,” “Troubleshooting Chart,” and “Detailed Troubleshooting.” Use the “Visual Checks” first to find a difficulty that shows up right after your kit is assembled. You can also use the other two sections right after your kit is assembled, or at some future time — if your Trainer ever stops working.

If the trouble is still not located after you complete the “Visual Checks,” check voltage readings against those shown in the “Schematic.” NOTE: All voltage readings were taken with a high impedance voltmeter (10 MΩ or greater).

In the extreme case where you are unable to resolve a difficulty, refer to the “Customer Service” information inside the rear cover of your Manual. Your Warranty is located inside the front cover.

**VISUAL TESTS**

1. Recheck the wiring. Trace each lead in colored pencil on the Pictorial as it is checked. It is frequently helpful to have a friend check your work. Someone who is not familiar with the unit may notice something you consistently overlook.

2. About 90 percent of the kits that are returned to the Heath Company for repair do not function properly due to poor connections and soldering. Therefore, you can eliminate many troubles by reheating all connections to make sure they are soldered.

3. Check to be sure that all the integrated circuits are in their proper location and that each IC pin is properly installed in its connector, and not bent or under the IC.

4. Check the values of the parts. Be sure in each step that the proper part has been wired into the circuit, as shown in the Pictorial diagrams. It would be easy, for example, to install a 470 Ω (yellow-violet-brown) resistor where a 4700 Ω resistor (yellow-violet-red) resistor should have been installed.

5. Check for bits of solder, wire ends, or other foreign matter which may be lodged in the wiring.

6. A review of the “Theory of Operation” may also help you determine the trouble.
Precautions

1. Be cautious when you test IC's. Although they have almost unlimited life when used properly, they are much more vulnerable to damage from excessive voltage or current than some other components.

2. Be sure you do not short any terminals to ground when making voltage measurements. If the probe slips, for example, and shorts out a bias or supply point, it is very likely to damage one or more IC or diode.

3. Do not remove an IC while the line cord is plugged in.

Substitution

Corresponding display components can be interchanged; IC's 23 through 28 can be interchanged, for example. If one display unit shows a digit incorrectly, interchange it with one of the other units to determine if the display or the circuit is faulty. If the circuit is faulty and there are no solder bridges on the associated foil, interchange the decoder/driver IC with one of the others. This troubleshooting method can also be used with other problems.
TROUBLESHOOTING CHARTS

NOTES:

1. The following chart lists parts to check. These parts indicate areas of the circuits where problems could exist. Check the circuitry and look for an assembly error or solder bridge. Parts are rugged and reliable. Consider a part to be bad last.

2. If you make a repair, make sure you eliminate the cause as well as the effect of the trouble. If, for example, you find a damaged part, be sure you find out what damaged the part. If the cause is not eliminated, the replacement part may also become damaged when you put the unit back into operation.

3. In several areas of the circuit boards, the foil patterns are quite narrow. When you unsolder a part to check or replace it, avoid excessive heat while you remove the part. A suction-type desoldering tool makes part removal easier.

POWER SUPPLIES

<table>
<thead>
<tr>
<th>DIFFICULTY</th>
<th>POSSIBLE CAUSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>No +5 V, +12 V, and −12 V supplies. LED1 not lit.</td>
<td>1. Fuse F1.</td>
</tr>
<tr>
<td></td>
<td>2. Transformer T1 primary wiring.</td>
</tr>
<tr>
<td></td>
<td>3. Line cord wiring.</td>
</tr>
<tr>
<td>No +5 V supply, in Standby or On position. LED1 not lit.</td>
<td>1. Transformer T1 secondary wiring (green and green-yellow leads).</td>
</tr>
<tr>
<td></td>
<td>2*. Regulator IC31.</td>
</tr>
<tr>
<td></td>
<td>3. Short circuit on 5 V line.</td>
</tr>
<tr>
<td>No +5 V supply in On position.</td>
<td>1. Switch SW1.</td>
</tr>
<tr>
<td></td>
<td>2. Short circuit on main 5 V line.</td>
</tr>
<tr>
<td>No +12 V supply in On position.</td>
<td>1.* Regulator IC29.</td>
</tr>
<tr>
<td>No −12 V supply in On position.</td>
<td>1.* Regulator IC30.</td>
</tr>
<tr>
<td>Have +5 V in Standby position. No +5 V in On position.</td>
<td>1. Shorted main +5 V line.</td>
</tr>
<tr>
<td>Have +5 V in Standby position. No +5 V to LED’s in On position.</td>
<td>1. Open main +5 V line.</td>
</tr>
<tr>
<td></td>
<td>2. Switch SW1.</td>
</tr>
</tbody>
</table>

*The voltage regulator IC’s have built-in short circuit protection. Therefore, the lack of voltage at an output connector may indicate a short or open circuit on the circuit board or in the wiring.
Troubleshooting Chart (cont’d.)

7-SEGMENT LED’s

<table>
<thead>
<tr>
<th>DIFFICULTY</th>
<th>POSSIBLE CAUSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>No LED’s light when “Segment Test” is shorted.</td>
<td>1. +5 V not supplied to LED’s.</td>
</tr>
<tr>
<td>All seven segments of one LED do not light when “Segment test” is shorted.</td>
<td>1. +5 V not supplied to this LED.</td>
</tr>
</tbody>
</table>
| One segment of an LED does not light when “Segment Test” is shorted. | 1. LED segment.  
2. Decoder driver.  
3. LED not properly installed. |
| All segments of all LED’s lit. | 1. Clear line of decoder driver IC’s shorted.  
| All LED’s light dimly when “Segment Test” is shorted. | 1. +5 V not supplied to IC’s 23 through 28. |
| All LED’s light when “Segment Test” is shorted, but one LED is dim. | 1. +5 V not supplied to associated decoder driver IC.  
2. Defective decoder driver IC. |
| All LED’s light, except one segment, when “Segment Test” is shorted. | 1. Associated LED.  
2. Associated decoder driver IC. |
| One LED stays lit. | 1. Associated LED.  
2. Associated decoder driver IC. |
| LED’s light dimly when experiments are connected and the Power switch is turned off. | 1. This is normal. |

DETAILED TROUBLESHOOTING

The microprocessor is very complex, such that any error in the system results in a complete breakdown of the system. Open or shorted address, data, or control lines; their associated IC’s; or a non-operating clock will all essentially show the same symptom (that is, when the unit is turned on, some or all of the LED segments will light, but nothing else happens). The following material gives you a systematic check of the Trainer circuitry to help you locate the problem. The material is divided into sections (which are listed below). If you know the section that the trouble is in, proceed to that section and start there. Otherwise, start at “Binary Data LED’s.”

- Binary Data LED’s
- Clock
- Reset
- Address Lines
- Data Lines
- Control Lines
- Decoding

Binary Data LED’s

If the +5-volt supply is operating, indicated by the LED1 next to the Power switch being lit, you can troubleshoot your Trainer without using test equipment.

Set the Power switch to On.

Cut a 14" length of yellow wire and remove 3/8" of insulation from each end. Refer to Pictorial 9-2 and insert one end of the wire into the LED connector block labeled 7, or to the block of an LED that you know does not work. Insert the other wire end into the +5 connector block. The LED directly above the connector block should light.

If the LED does not light:

A. Visually inspect the LED’s. The flat at the base of each LED should face the top of the circuit board.
B. Unplug the line cord, remove IC1 from its socket, and plug the line cord back in.

With the indicated end of the yellow wire, one at a time, touch the eight circuit board plated-through holes shown. The eight LED's should light one at a time. If they do not, replace the LED that does not light.

C. With pliers, flatten one end of the yellow wire.

Carefully insert the wire into pin 20 of socket IC1 and touch the other wire end to the indicated plated-through hole. The 0 LED should light. If it does not light, check the IC socket pins and the circuit board foils to find out why +5 volts is not at pin 20 of the socket. Then remove the yellow wire.

D. Unplug the line cord.

Be sure the pins of IC1 are straight and then properly reinstall the IC in its socket.

Reconnect the yellow wire to LED connector block 7 (or to the connector block of an LED that you know was not working) and the +5 connector block. The data LED should light. If it does not light, replace IC1.

Clock

The simplest test to determine if the clock (IC19) is operating is to place a portable radio near the clock and tune the radio across the broadcast band. If the clock is operating, you will hear several "beat" signals. Unplug the Trainer's line cord and the beat signal will disappear if it is caused by the microprocessor clock.

There are four different clock outputs used in the system (pins 7, 9, 13, and 15). Usually, the outputs of a clock that has failed will assume a logic 0 state. To test the clock, use a wire and connect a data LED (LED 2 through LED 9) to the clock's four outputs. If the clock is working, the LED will light but it will be noticeably dimmer than the same LED connected to +5 volts. This is due to the 50% duty cycle of the clock. See Pictorial 9-3.
PICTORIAL 9-4

**Reset**

Refer to Pictorial 9-4 and connect a wire from a data LED connector block to IC11 pin 40. The LED should light. While you hold the test lead on pin 40, press the Reset key. The LED should go out while the Reset key is pressed and come back on again when it is released. Then remove the wire.

To test the reset input, connect a wire from the indicated Binary Data connector block to LED connector block 0.

**Set data switch 0 to logic 1.**

Connect another wire from the indicated Binary Data connector block to the circuit board soldered connection just left of the 0 key. The lamp should stay on until you push the Reset key; then it will go out. It will come back on when you release the key.

Other effects of pushing the Reset key will be covered later after you check the address and data lines. Remove the two wires.
Test Wires

The following paragraphs instruct you how to make indicators for testing tri-state* devices. These are necessary for testing address and data lines in the following sections.

Unplug the line cord.

Refer to Pictorial 9-5 (Illustration Booklet, Page 9) and unsolder and disconnect the indicated lead of resistor R24 from the circuit board as shown.

Prepare a 2" yellow wire. Temporarily solder one end of the wire to the free lead of the resistor and plug the other end of the wire into connector block 6.

Prepare two 12" yellow wires. Remove 3/8" of insulation from both ends of each wire.

Insert one end of one wire into LED connector block 7.

Insert one end of the other wire into LED connector block 6.

In the following sections, these two wires will be referred to as test wire 7 and test wire 6. Be sure you reconnect and resolder the loose resistor lead after you locate and repair the problem.

Plug in the line cord. LED 7 should be on and the other data LED's should be off.

Address Lines

In checking the buffered address lines, you will look for two basic problems:

1. Lines that are shorted.

2. Lines that are not connected properly.

To perform these tests, you will tri-state the CPU. In this state, the address lines from the CPU and from the buffers are in a high impedance state. Therefore, any logic level can be put on these lines. Data input switches will apply test logic levels to the address lines, and data LED displays will serve as logic level indicators.

Prepare the following yellow wires. Cut them to the lengths shown and remove 3/8" of insulation from each end:

<table>
<thead>
<tr>
<th>Wires</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4&quot;</td>
</tr>
<tr>
<td>2</td>
<td>8&quot;</td>
</tr>
</tbody>
</table>

Refer to Pictorial 9-5 and connect a 4" wire from the ground connector block to TSC.

Touch test wire 6 to the +5 connector block to test the LED. LED 6 should light. Touch test wire 6 to IC11 pin 39. LED 6 should again light; this indicates proper TSC voltage. If the LED does not light, proceed to “Control Lines” tests on Page 103.

Touch test wire 6 to IC7 pin 1 and then to IC8 pin 1. The LED should light both times. This indicates proper tri-state voltage. If the LED does not light, proceed to “Control Lines” tests on Page 103.

Touch test wire 7 to IC7 pin 19 and then to IC8 pin 19. LED 7 should go out both times; this indicates the correct voltage to tri-state the address buffers.

One after another, touch test wire 7 to each address output connector (A<sub>v</sub>-A<sub>12</sub>). LED 7 will remain lit unless the line touched is shorted to logic 0. If the LED goes out, trace the foil pattern and look for a solder bridge. If this does not solve the problem, then remove the IC's connected to that line, one at a time, to check for a defective IC. CAUTION: Do not remove or install IC's with the line cord plugged in. If you remove a MOS IC, place it in the protective foam in which you received it. This will prevent possible damage from a static charge. (See Page 36 for instructions on how to handle MOS IC's.)

One after another, touch test wire 6 to each address output connector. The LED will remain unlit. If the LED should light, it indicates a short to logic 1. Use the same procedure as above to check for the cause of the problem.

* Registered Trademark, National Semiconductor
The next test is to make sure that none of the address lines are shorted together. To do this, you will put a logic 1 on one line and a logic 0 on the line beside it. If the two lines are shorted together, the logic 0 will cause both LED's to be off. You will also check to see that the address line is indeed connected to all the IC's where it should be.

Connect a 4" wire from the connector block of data switch 0 to the connector of data LED 0.

Connect a 4" wire from data switch 1 to LED 1.

Place the data 0 switch to logic 1 and the data 1 switch to logic 0.

Connect an 8" wire from data LED 0 to A₀, and an 8" wire from data LED 1 to A₁. LED 0 should be on and LED 1 should be out. If there is a short between lines A₀ and A₁, both LED's will be out. If the LED's are out, check for solder bridges or defective IC's.

Connect test wire 6 to all the IC pins indicated in the following chart as being connected to A₀ (IC7 pin 12, IC12 pin 24, etc.). The LED should light as each pin is touched. If it does not light, an open circuit exists between the address terminal and the pin being tested.

Move the wire that is at line A₁ to A₂. Then move the wire at A₀ to A₁.

As before, LED 0 should be on and LED 1 should be out. If both LED's are out, this time check for a short between lines A₁ and A₂.

Connect test wire 6 to all the IC pins indicated in Test Chart A as being connected to A₁. The LED should light as each pin is touched.

Continue moving the two wires towards A₁₅, one position at a time, and make the tests in the above three steps until all the address lines have been checked.
# TEST CHART A

<table>
<thead>
<tr>
<th>BUFFERED ADDRESS LINES</th>
<th>BUFFERS</th>
<th>ROM</th>
<th>RAM</th>
<th>DISPLAY LATTICES</th>
<th>ADDRESS DECODING</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IC7 PIN</td>
<td>IC8 PIN</td>
<td>PIN</td>
<td>IC14</td>
<td>IC15</td>
</tr>
<tr>
<td>A0</td>
<td>12</td>
<td>24</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>14</td>
<td>23</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>16</td>
<td>22</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td>18</td>
<td>21</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td>9</td>
<td>20</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td>7</td>
<td>19</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A6</td>
<td>5</td>
<td>18</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A7</td>
<td>3</td>
<td>17</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A8</td>
<td>12</td>
<td>16</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A9</td>
<td>14</td>
<td>15</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A10</td>
<td>16</td>
<td>14</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A11</td>
<td>18</td>
<td>13</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A12</td>
<td>3</td>
<td>10</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A13</td>
<td>5</td>
<td></td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A14</td>
<td>7</td>
<td></td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A15</td>
<td>9</td>
<td></td>
<td>13</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To check the address lines between the CPU and the address connectors, it is necessary to remove the tri-state condition from the buffers and the CPU.

Be sure the line cord is unplugged.

Remove IC11 from its socket.

Remove IC5 and bend pin 11 out slightly. Then reinstall the IC so that pin 11 is not in the socket.

Remove all the previously installed wires except test wires 6 and 7.

Follow chart below and reconnect the wires.

<table>
<thead>
<tr>
<th>WIRE</th>
<th>FROM</th>
<th>TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>4&quot;</td>
<td>Data switch 0</td>
<td>LED 0</td>
</tr>
<tr>
<td>4&quot;</td>
<td>Data switch 1</td>
<td>LED 1</td>
</tr>
<tr>
<td>4&quot;</td>
<td>+5</td>
<td>BA</td>
</tr>
<tr>
<td>8&quot;</td>
<td>LED 0</td>
<td>A_0</td>
</tr>
<tr>
<td>8&quot;</td>
<td>LED 1</td>
<td>A_1</td>
</tr>
</tbody>
</table>

Set data input switches 0 and 1 to logic 1.

Plug in the line cord.

Use test wire 6 and check for correct logic levels at IC7 and IC8. Pin 1 is logic 0 (LED 6 off) and pin 19 is logic 1 (LED 6 on).

Address lines A_0 and A_1 should be logic 1, indicated by LED 0 and LED 1 being lit. Remove the 12" test wire from LED 7 and insert one end in the GND connector.

Touch the free wire end to IC7 pin 8. The LED connected to A_0 should go out, while the LED connected to A_1 will remain lit. If both LED's go out, there is a short circuit between the A_0 and A_1 lines, between the CPU and the buffer inputs.

Follow Test Chart B to check all the address lines.
## TEST CHART B

<table>
<thead>
<tr>
<th>8&quot; WIRE FROM LED 0 TO:</th>
<th>8&quot; WIRE FROM LED 1 TO:</th>
<th>GND IC</th>
<th>GND PIN</th>
<th>TURNS OFF LED</th>
</tr>
</thead>
<tbody>
<tr>
<td>A₀</td>
<td>A₁</td>
<td>7</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>A₂</td>
<td>A₁</td>
<td>7</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>A₂</td>
<td>A₃</td>
<td>7</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>A₄</td>
<td>A₃</td>
<td>7</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>A₄</td>
<td>A₅</td>
<td>7</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>A₆</td>
<td>A₅</td>
<td>7</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>A₆</td>
<td>A₇</td>
<td>7</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>A₇</td>
<td>A₇</td>
<td>7</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>A₈</td>
<td>A₉</td>
<td>8</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>A₉</td>
<td>A₉</td>
<td>8</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>A₁₀</td>
<td>A₁₁</td>
<td>8</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>A₁₀</td>
<td>A₁₁</td>
<td>8</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>A₁²</td>
<td>A₁₃</td>
<td>8</td>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>A₁₄</td>
<td>A₁₃</td>
<td>8</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>A₁₄</td>
<td>A₁₅</td>
<td>8</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>A₁₄</td>
<td>A₁₅</td>
<td>8</td>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>

Properly replace IC11 and IC5 into their sockets.

Remove all the wires except the test wires.
**Data Lines**

To check data lines for opens and shorts, you will input data through the data buffers, alternate logic 0 and logic 1 on adjacent data lines, and then look for the correct data at the affected IC pins. To do this, you will need the following yellow wires. Cut them to the lengths specified and remove 3/8" of insulation from each end.

<table>
<thead>
<tr>
<th>WIRES</th>
<th>LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>8&quot;</td>
</tr>
<tr>
<td>3</td>
<td>4&quot;</td>
</tr>
</tbody>
</table>

If not already done, refer to Page 97 and prepare the two test wires as instructed there.

Refer to Pictorial 9-6 (Illustration Booklet, Page 9) and and install a 4" wire between GND and TSC to tri-state the CPU.

The data I/O buffers are bi-directional transceivers with the enable line to provide data to the output connectors.

Touch test wire 6 to pins 1 and 13 of IC9 and IC10. The LED should light, indicating that the buffers are in the right state.

Touch the test lead to each of the data connectors (D₀-D₇). The lamp should light at each terminal, indicating that the data lines are tri-stated and none of the data lines are shorted to ground. If the LED does not light, check both the terminal and the CPU sides of the data lines involved.

Install the following jumper wires.

<table>
<thead>
<tr>
<th>WIRE</th>
<th>FROM</th>
<th>TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>4&quot;</td>
<td>Data switch 0</td>
<td>LED 0</td>
</tr>
<tr>
<td>4&quot;</td>
<td>Data switch 1</td>
<td>LED 1</td>
</tr>
<tr>
<td>8&quot;</td>
<td>LED 0</td>
<td>D₀</td>
</tr>
<tr>
<td>8&quot;</td>
<td>LED 1</td>
<td>D₁</td>
</tr>
<tr>
<td>8&quot;</td>
<td>GND</td>
<td>RE</td>
</tr>
</tbody>
</table>

Set data switch 0 to logic 1.

Set data switch 1 to logic 0.

Refer to Test Chart C and touch test wire 6 to any IC pin to which line D₀ is connected. The LED should light. If the LED does not light, there is a short between the D₀ and D₁ lines. Visually check for shorts. Remove the IC's connected to the D₀ and D₁ lines, one at a time, to determine if a short exists in an IC. CAUTION: Do not remove or install IC's with the line cord plugged in. If a short is not indicated, test all pins to which line D₀ is connected by moving the data switch from logic 1 to logic 0 while you touch each pin with the test wire. If you do not obtain the correct results at all pins, check for an open circuit to the pin not showing the proper response. (NOTE: Line D₀ also goes to the display latches and is inserted at IC21 pins 9 and 10.)

Move the leads from LED 0 and LED 1 to buffer data connectors D₁ and D₂, and repeat the test for D₁. Continue this procedure until you have checked all the data lines.
TEST CHART C

<table>
<thead>
<tr>
<th>CONNECTOR SIDE</th>
<th>CPU SIDE</th>
<th>CPU</th>
<th>ROM</th>
<th>RAM</th>
<th>KEYBOARD BUFFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC9 PIN</td>
<td>IC10 PIN</td>
<td>IC9 PIN</td>
<td>IC10 PIN</td>
<td>IC11 PIN</td>
<td>IC12 PIN</td>
</tr>
<tr>
<td>D₀</td>
<td>8</td>
<td>6</td>
<td>33</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>D₁</td>
<td>9</td>
<td>5</td>
<td>32</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>D₂</td>
<td>10</td>
<td>4</td>
<td>31</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>D₃</td>
<td>11</td>
<td>3</td>
<td>30</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>D₄</td>
<td>8</td>
<td>6</td>
<td>29</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>D₅</td>
<td>9</td>
<td>5</td>
<td>28</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>D₆</td>
<td>10</td>
<td>4</td>
<td>27</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>D₇</td>
<td>11</td>
<td>3</td>
<td>26</td>
<td>9</td>
<td>12</td>
</tr>
</tbody>
</table>

Line D₀ is also applied to IC21 pins 9 and 10. The D₀ output, IC21 pin 8 and IC6 pin 2, is connected from the output of IC6 (pin 18) to pin 13 of IC23 through IC28.

Remove all the wires except the test wires.

Control Lines

If not already done, refer to Page 97 and prepare the two test wires as instructed there.

There are nine lines other than data, clock, and address that affect the operation of the CPU. Four lines are always logic 1, unless they are pulled low by an external connection. These are RESET, HALT, IRQ, and NMI. Reset has been checked earlier in this section and will be covered again later in greater detail. The three other lines are connected through noninverting buffers to the CPU. The connector and the associated CPU pin are therefore at the same logic level. To test these three lines, touch test wire 6 to the connector and then to the corresponding CPU pin. The LED should light at both locations. Then repeat this procedure with a wire installed from GND to the connector block associated with the line being checked. The LED should not light at either the connector or the CPU pin.

In the above test, if you fail to get the correct indication, check for open or shorted lines. Also, IC6 may be defective.
To check the five remaining control lines (R/W, TSC, BA, VMA, and DBE) plus VMAφ2, you will use Halt and TSC, which forces a given logic level to appear on these control lines. Refer to the following chart and connect a wire from ground to the designated connector, and check for the desired result by touching test wire 6 to the indicated connector or IC pin.

<table>
<thead>
<tr>
<th>TOUCH GROUND WIRE TO CONNECTOR</th>
<th>DBE IC11 PIN 36</th>
<th>TSC IC11 PIN 39</th>
<th>R/W IC11 PIN 34</th>
<th>VMA IC11 PIN 5</th>
<th>BA IC11 PIN 7</th>
<th>IC4 PINS 1, 4, 10</th>
<th>IC4 PIN 13</th>
<th>VMAφ2</th>
<th>BA</th>
<th>TSC</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>———</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>„1“</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>HALT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TSC</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

1 The VMAφ2 state does not appear to change. However, the LED will not be as brightly lit when the CPU is running as it is when the CPU is in the Halt or TSC states.

2 Although the R/W line changes, the output connector does not change because the buffer is tri-stated when R/W is low.

**RESET**

Previous tests indicated that the logic level on this pin is correct.

When the Reset key is closed, reset goes low, VMA and BA are low, and R/W is high. In addition, the CPU puts the first address of the reset sequence on the address line. This address is FFFE. Therefore, test all the address lines with test wire 6. They will all be logic 1 except for A9, which is logic 0.

**Decoding**

In this section, you will put various addresses on the lines and then refer to the "Decoding Chart" and look at logic levels at the decoding IC's to check their operation. In each case, VMAφ2 must be logic 0 in order to provide the proper addressing.

If not already done, refer to Page 97 and prepare the two test wires as instructed there.

VMAφ2, for internal operation, is taken from the line connecting IC5 pin 6 to IC4 pin 9. To perform tests on the decoding section, you will need to pull this line to logic 0. To avoid damaging IC5 when you do this, refer to Pictorial 9-7, carefully remove IC5 from its socket, bend pin 6 out just far enough to clear the edge of the socket, and then reinstall the IC so that pin 6 is not in the socket.

Install one end of a 4" wire in LED 4. Insert the free end of this wire in the plated-through hole just below pin 9 of IC4. Temporarily solder this point on the bottom side of the circuit board.

Install a 4" wire from LED 4 to data switch 4. Data switch 4 will now determine the logic level of VMAφ2, and LED4 will display the level. 1 is ON, 0 is OFF.

Install a 4" wire from GND to TSC to tri-state the address lines so you can place an address on the lines.
PICTORIAL 9-7

In the following steps, refer to Pages 10 and 11 in the Illustration Booklet and use test wire 6 to check for proper address decoding. It is not necessary to go through the entire “Decoding Chart” unless the “End Result” is not correct. After you place an address on the address lines, check all the end results to make sure a problem does not exist, which results in more than the desired function being addressed. The logic level on the pins listed in the End Result column should be opposite of that indicated when the IC or function is not being addressed.

To address RAM 00 — Install a wire from A_{15} to GND. Then install wires from A_{15} to A_{14}, A_{14} to A_{13} etc., until lines A_8 through A_{15} are all connected together. To make sure VMA\phi 2 is doing its job, switch D4 between logic 1 and logic 0. The chart “End Result” should only be obtained when VMA\phi 2 is logic 0.

To address RAM 01 — Remove the wire installed between A_8 and A_9 for RAM 00 — Install a wire from A_8 to +5.

To address ROM FC — Remove the wire installed between A_8 and A_{15}, and install it between A_8 and A_9. Move the wire from A_{15} to +5, and the wire from A_8 to GND instead of +5.

To address the keyboard C0-X — Remove the wire from between A_{14} and A_{15}, and install it between A_9 and A_{10}. Address lines A_9 through A_7 are “don’t care” lines, so let them float. Install 4” wires from data switch 0, 1, and 2 to the corresponding LED terminals and 8” wires from the connectors to the corresponding address lines.

With the keyboard address on the lines, first look for the proper end result in the “Main Decoding Chart.” If it is correct, proceed to the “Keyboard Column Address Decoding Chart.”

In order to determine if a key is depressed, the monitor program causes the CPU to put the keyboard address on the line. Then it looks at the data lines to determine if a key is closed, which is indicated by the presence of a logic 0 on one of the affected data lines (D_7 through D_3). The eight high-order bits (C0 hex) are decoded and enable the keyboard buffer.

The three lower-order bits (3, 5, or 6 hex) place a logic 0 on one of the key columns. If a key is closed in the column address with a logic 0, a logic 0 will appear on the corresponding data line. Then you can tell which key is closed.
Place data switches 0, 1, and 2 in their logic 1 positions. The address lines to the key columns are all logic 1 and no key is actually addressed. Depress keys F, E, and D. All data lines should remain logic 1. If a line is logic 0, it indicates a shorted address line to the column of keys containing the depressed key.

Put the address for the right-hand column of keys (hex 3) on the three low bit address lines using the data switches. The LED will indicate that the address is correct. Connect the logic probe to each of the data output connectors, D₈ through D₁₅. All the connections should be logic 1. If one of the data lines should be logic 0, a short to GND is indicated in the keyboard circuit. This could be caused by the key associated with the data line or it could be the row of three keys. For example, with the hex 3 address on the line, we find D₈ to be logic 0. The problem could be a short that only affects key F, or it could be a short affecting the row of keys D, E, and F. If you change the hex 3 portion of the address to either hex 5 or hex 6, and D₈ changes to logic 1, the short is only associated with key F. However, if the logic 0 remains, the problem is associated with the line to the entire row.

If the data lines are all logic 1 with no key depressed and a hex 3 address, depress key F. Look at all the data lines while the key is depressed. Only D₀ should be logic 0. If, for example, lines D₀ and D₁ indicate logic 0, a short exists between keys F and C, or between the rows of keys D, E, F, and A, B, C. Again, to determine individual key versus rows of keys, change the column address to hex 5 and depress key E. If only D₀ is now logic 0, the problem exists between keys F and C. If D₀ and D₁ are logic 0, the problem is a short between the D, E, F and A, B, C rows or keys.

With the hex 3 address on the line, depress keys D and E. If data line D₀ goes to logic 0, a short is indicated between the column of key associated with the key depressed and the column containing the F key.

To address the display LED's CIX —. Remove the wire at A₀-A₉. Move the GND wire from A₈ to A₉. Then connect a wire from A₉ to +5. Check for the proper end result indicated in the Main Decoding Chart. Move the 8” wires installed at A₀, A₁, and A₂ to A₄, A₅, and A₆. Use data switches 0, 1, and 2 to apply the LED address as shown in the Display LED Chart. Test for the proper logic level at pin 14 of the addressed IC.

To address an LED segment CIXX, move the 8” wires from A₄, A₅, and A₆ to A₀, A₁, and A₂ respectively. Now use data switch 0, 1, and 2 to address the desired segment. Move the wire soldered to VMAF₂ from LED4 to GND. Install 4” wires from data switches 3, 4, and 5 to LED connectors 3, 4, and 5. Install 8” wires from LED 3, 4, and 5 to address lines A₄, A₅, and A₆. (NOTE: The data switches are one number from the corresponding address line so LED’s 6 and 7 can still be used as logic level indicators.) Data switches 3, 4, and 5 can now be used to address the desired display LED.

The D₀ data line controls the state of the LED segment when the segment is addressed and VMAF₂ is logic 0. If D₀ is logic 1, the segment will light and if D₀ is logic 0, the segment will be off. The D₀ data line is connected through IC21 and IC6 to the D data input (pin 13) of decoder latch IC’s.

The “D₀ Logic Level Chart” shows the levels at the various connections on the D₀ segment control line. To control the logic level on the D₀ data line, connect an 8” wire from R₁ to GND. Connect another 8” wire to the D₀ connector. The free end of this wire need not be connected to provide a logic 1, but it must be connected to GND to provide a logic 0 level on D₀. To test this area, place the address for an LED and a segment on the low-order address lines, touch the output pin that should be affected with test wire 6, and then watch both the probe and the selected LED segment. If D₀ is logic 1, the segment should bright and the logic probe should indicate logic 0. The reverse is true if D₀ is logic 0.

If you wish to check different LED’s or segments, insert the D₀ input lead into the 1 Hz square wave connector. The address segment will turn on or off approximately every 1/2 second.

To test the latch action of the decoder latches, move the lead soldered to VMAF₂ from GND to LED 6 and add a wire from LED connector 6 to data switch 6. If data switch 6 is logic 0, the addressed LED segment will follow the D₀ logic level. To check the latching action, move data switch 6 from 0 to 1 while the addressed LED segment is either on or off. The segment should remain in the state it is in when VMAF₂ is moved to logic 1.

Remove the wires from your Trainer, properly replace IC5, and then reconnect and resolder the free lead of resistor R24.
# SPECIFICATIONS

<table>
<thead>
<tr>
<th>Component</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU (Central Processing Unit)</td>
<td>8-bit parallel, NMOS, bus oriented 6800.</td>
</tr>
<tr>
<td>ROM (Read Only Memory)</td>
<td>NMOS, 1024 bytes.</td>
</tr>
<tr>
<td>RAM (Random Access Memory)</td>
<td>NMOS, 256 bytes (plus sockets for additional 256 bytes).</td>
</tr>
<tr>
<td>Clock Frequency</td>
<td>500 kHz (approximately).</td>
</tr>
<tr>
<td>Display</td>
<td>Six 7-segment LED digits.</td>
</tr>
<tr>
<td>Keyboard</td>
<td>Hexadecimal (0-F and Reset). 1 through F are dual-function keys and also enter commands.</td>
</tr>
<tr>
<td>Input Switches</td>
<td>Eight miniature switches in a dual-in-line package.</td>
</tr>
<tr>
<td>(For experiments.)</td>
<td></td>
</tr>
<tr>
<td>LED Monitor Lights</td>
<td>Eight red LED's with separate input terminals. (For experiments.) +5 volts at 1.5 amperes (.5A available for breadboard at output terminal.)</td>
</tr>
<tr>
<td>Power Supplies</td>
<td>+12 volts, and −12 volts at 50 milliamperes at output terminals.</td>
</tr>
<tr>
<td>Power Requirements</td>
<td>105-130 volts or 210-260 volts rms, 50-60 Hz. 30 watts maximum.</td>
</tr>
<tr>
<td>Fuse</td>
<td>3/8-ampere, slow-blow.</td>
</tr>
<tr>
<td>Dimensions</td>
<td>12-1/8&quot; wide × 11-3/4&quot; deep × 3-1/2&quot; high. (30.8 × 29.8 × 8.9 cm.)</td>
</tr>
<tr>
<td>Net Weight</td>
<td>4 lbs (1.8 kg).</td>
</tr>
</tbody>
</table>

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.
THEORY OF OPERATION

As you read this section, refer to the Block Diagram (Illustration Booklet, Page 12) and the Schematic (fold-in).

The operation of the CPU (microprocessor, IC11) is very complex. Therefore, this section of the Manual will not discuss the internal operation of the CPU, but will discuss how the sections of circuitry in your Trainer operate together. For information concerning the CPU, refer to Motorola's M6800 application manual.

Many lines are connected to front panel connectors, as shown on the Schematic. Some are buffered and some are not. Most of these connections and their buffers will not be mentioned in the following paragraphs.

The Reset key is connected to the clock (IC19) which produces a proper reset pulse. This pulse is applied through tri-state buffer IC6 to the reset input (pin 40) of the CPU.

Two non-overlapping outputs are connected from the clock to the CPU. The memory φ2 output is used for internal timing and is connected through IC4 to the DBE input (pin 36) of the CPU.

The VMA line from the CPU is buffered by IC6 and then NANDed by IC5B with memory φ2 to produce VMAφ2. This signal is then applied to the address decoding circuits.

The CPU R/W line is coupled through IC4 to the R/W inputs of RAM.

The input signal to TSC is applied through inverter IC5C. TSC is normally logic 0 and is connected through IC4 to the input of IC5A and to C of the address buffers, IC7 and IC8. Line BA is normally connected through IC5D to the control line of the TSC portion of IC4. The output of IC5A is logic 1 and is connected to the control lines in IC4 for R/W, DBE, and VMAφ2; keeping these sections enabled. The output of IC5A is also connected to an enable input of the address buffers.

If TSC is pulled to logic 1, the input to IC5A and C on the address buffers also become logic 1. The output of IC5A and, therefore, the inputs to address enable and the control lines for the other three sections of IC4 become logic 0. The address, R/W, DBE, and VMAφ2 buffers are all tri-stated. In this state, DBE is held at logic 0 by a pull-down resistor and the other three lines are held at logic 1 by pull-up resistors. When BA goes to logic 0, the TSC section of IC4 is tri-stated, TSC does not control the output, and the output is held at logic 1 by a pull-up resistor which tri-states the address buffers R/W, DBE, and VMAφ2 as described above.

The address lines are buffered by IC7 and IC8. The buffers have active outputs or are tri-stated as previously described.

The eight high-order address lines are connected to the address decoding IC's; VMAφ2 is also applied to the decoding section. This line must be logic 0 to obtain proper decoding. With the high-order byte 00 on the lines, a logic 0 is placed on CE for IC14 and IC15, and its 256 bytes of RAM memory may be addressed by lines A0-A7. R/W from the CPU determines if information is to be stored into or read from the RAM.

High-order byte 01 does the same thing for the optional 256 bytes of RAM at that address.

With the high-order byte FC, FD, FE, FF; the address decoder places a logic 0 in CS1 of the ROM. Address lines A8, A9, and A12 place logic 1 on CS0, CS2, and CS3; and lines A9 through A7 can address the 1024 bytes of read only memory.

Buffer IC13 is normally in its tri-state condition. When the high-order address byte C0 is decoded, a logic 0 is placed on its control lines to enable it. Address lines A0, A1, and A2 apply a logic 0 to one of the key columns and logic 1 to the other two columns. If a key is closed in the column with Logic 0 on it, a logic 0 is placed on the data line for the row of keys. Which key is closed is determined by the monitor program 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 accidentally changing the address due to the lines being shorted together.
When high-order address byte C1 is decoded, the output of the decoder places a logic 0 on the D input of IC22. IC22 is a 4 to 10 line decoder. If a BCD number from 0 through 10 is placed on the inputs, the output line corresponding to that number will be logic 0. Output lines 1 through 6 are connected to the enable inputs of the six display latch drivers, IC23 through IC28. If the D input to IC22 (which is BCD equivalent of 8) is high, the BCD input will always be greater than 8 and the output lines actually in use cannot be decoded. With the D input held low, the BCD information supplied to the other three inputs will be 0 through 7. These three inputs are connected to address lines A4, A5, and A6 and will determine which output line will be logic 0 by their logic levels. A hex 6 or BCD 110 on lines A4, A5, and A6 will cause the enable line for the left-most latch driver (IC23 and DISPLAY LED H) to be logic 0. Hex 1 or BCD 001 enables IC28 and DISPLAY LED C.

Address lines A8, A9, and A10 are connected to the latch select inputs of all six latch drivers. The BCD code on these lines (hex 0 through 7), is decoded in the enabled IC and results in the corresponding output line following the logic level on the D input of that IC. Each of the output lines is connected through one of seven segments of display LED or decimal point, and a current-limiting resistor, to +5 V. If the D input is logic 0 the addressed output line will be 0 also, and a corresponding segment will be lit. If D is logic 1, the output line is 1 and the segment will be out. The D8 data line is inverted by IC21C and applied to the latch driver D inputs through IC6. Therefore, if line D0 is logic 1, the D input is logic 0 and the addressed segment will be lit. If D0 is logic 0, the addressed segment is off. The status of the output lines and LED segment, as determined by the address and D0 logic level, is then latched when the enable line returns to logic 1.

The data lines are connected directly to the various devices in the system. Data buffers IC9 and IC10 are bus transceivers. They are wired to normally provide output from the data lines to the data terminals. Connecting RE to logic 0 reverses the input output pins so you can input data from the connectors.

**Binary Data Section**

The eight section data switch has one side of all switches connected to ground. The other side of each section has a 4700 ohm pull-up resistor to the switched 5 V power supply. The connectors above the switch provide convenient connection for two wires to each switch section. With a switch in the lower (closed) position, the associated terminal will provide a logic 0 level (ground). In the up (open) position the level will be logic 1. The switch sections are numbered 0 through 7 from right to left. The eight connectors numbered 0 through 7 are inputs to the non-inverting buffer IC1. An 8200 ohm pull-down resistor is connected through each input terminal to ground to hold the input at logic 0 when no connection is made to the terminals. Each buffer output is connected through an LED and a 180 ohm current-limiting resistor to ground. When the inputs to the buffer are logic 0, the outputs are also 0 and the LED is off. When the input rises to logic 1, the output also rises to logic 1 and lights the LED.

**Power Supplies**

The voltage from one of the center-tapped secondary windings (green leads) of power transformer T1 is rectified by diodes D1 and D2, filtered by capacitor C1, and regulated by IC31 to produce the +5-volt DC supply. With switch SW1 in the On position, +5 volt is supplied throughout the system. When SW1 is in the Standby position, +5 volt is not supplied to the display LED's, data switches, or the +5 V connector block.

The other center-tapped secondary winding (red) is rectified by diodes D3 and D5, filtered by C7, and regulated by IC30 to provide a -12-volt supply. This same winding is rectified by diodes D4 and D6, filtered by C6, and regulated by IC29 to provide a +12-volt supply. These two supplies are provided for bread-boarding and are not connected in the system. They are available at the appropriate connector blocks only when switch SW1 is in the On position.

**Square Wave Outputs**

The AC voltage at the anode of diode D6 is coupled by R5 and R6 to a section of voltage comparator IC18. Diode D10 keeps the AC voltage from driving the input negative with respect to ground. This section of the comparator is a zero-crossing detector to provide a symmetrical TTL compatible square wave that is in sync with the AC line.

A second section of IC18 is used as an oscillator to produce a TTL compatible square wave at approximately 1 Hz. The symmetry and frequency of the square wave are determined by C13, R13, and R14.
# SEMICONDUCTOR IDENTIFICATION CHARTS

## DIODES

<table>
<thead>
<tr>
<th>COMPONENT</th>
<th>HEATH PART NUMBER</th>
<th>MAY BE REPLACED WITH</th>
<th>IDENTIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1, D2</td>
<td>57-42</td>
<td>3A1</td>
<td></td>
</tr>
<tr>
<td>D3, D4, D5, D6</td>
<td>57-65</td>
<td>1N4002</td>
<td></td>
</tr>
<tr>
<td>D7, D8, D9, D10</td>
<td>56-56</td>
<td>1N4149</td>
<td></td>
</tr>
<tr>
<td>LED1</td>
<td>412-611</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LED2, LED3, LED4, LED5, LED6, LED7, LED8, LED9</td>
<td>412-616</td>
<td>FLV117</td>
<td></td>
</tr>
<tr>
<td>H, I, N, Z, V, C</td>
<td>411-831</td>
<td>TIL312</td>
<td></td>
</tr>
</tbody>
</table>

**Important:** The banded end of diodes can be marked in a number of ways.

- **Anode**: FLAT
- **Cathode**: ANODE

---

**TOP VIEW**

1. SEGMENT A
2. SEGMENT B
3. COMMON ANODE
4. NOT USED
5. NOT USED
6. NOT USED
7. SEGMENT C
8. SEGMENT D
9. RIGHT DECIMAL
10. SEGMENT E
11. SEGMENT F
12. NOT USED
13. SEGMENT G
14. COMMON ANODE

Note: Pins 3 and 24 are internally connected together.
### INTEGRATED CIRCUITS

<table>
<thead>
<tr>
<th>COMPONENT</th>
<th>HEATH PART NUMBER</th>
<th>MAY BE REPLACED WITH</th>
<th>IDENTIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC1, IC6, IC7, IC8</td>
<td>443-824</td>
<td>74LS241</td>
<td><img src="image1.png" alt="Image" /></td>
</tr>
<tr>
<td>IC2, IC3 IC20, IC22</td>
<td>443-807</td>
<td>74LS42</td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>IC4</td>
<td>443-717</td>
<td>74126</td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>IC5, IC21</td>
<td>443-26</td>
<td>74S00</td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>IC9, IC10</td>
<td>443-839</td>
<td>74LS243</td>
<td><img src="image5.png" alt="Image" /></td>
</tr>
</tbody>
</table>
## Integrated Circuits, Cont’d.

<table>
<thead>
<tr>
<th>COMPONENT</th>
<th>HEATH PART NUMBER</th>
<th>MAY BE REPLACED WITH</th>
<th>IDENTIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC11</td>
<td>443-827</td>
<td>MC6800P</td>
<td><img src="image1.png" alt="Diagram" /></td>
</tr>
<tr>
<td>IC12</td>
<td>444-17</td>
<td>MCM6830A*</td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
<tr>
<td>IC13</td>
<td>443-720</td>
<td>40097</td>
<td><img src="image3.png" alt="Diagram" /></td>
</tr>
<tr>
<td>IC14, IC15, IC16, IC17</td>
<td>443-721</td>
<td>2112-2</td>
<td><img src="image4.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

* Must be mask programmed from the listing in this Manual.
### Integrated Circuits Cont'd.

<table>
<thead>
<tr>
<th>COMPONENT</th>
<th>HEATH PART NUMBER</th>
<th>MAY BE REPLACED WITH</th>
<th>IDENTIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC18</td>
<td>442-616</td>
<td>LM3302N, LM2901N, or µA775 (selected)</td>
<td><img src="image1" alt="IC18 Diagram" /></td>
</tr>
<tr>
<td>IC19</td>
<td>443-840</td>
<td>MC6875</td>
<td><img src="image2" alt="IC19 Diagram" /></td>
</tr>
<tr>
<td>IC23, IC24, IC25, IC26, IC27, IC28</td>
<td>443-804</td>
<td>74LS259</td>
<td><img src="image3" alt="IC23-28 Diagram" /></td>
</tr>
<tr>
<td>IC29</td>
<td>442-644</td>
<td>LM78L12</td>
<td><img src="image4" alt="IC29 Diagram" /></td>
</tr>
<tr>
<td>IC30</td>
<td>442-646</td>
<td>LM79L12AC</td>
<td><img src="image5" alt="IC30 Diagram" /></td>
</tr>
<tr>
<td>IC31</td>
<td>442-30</td>
<td>µA309K</td>
<td><img src="image6" alt="IC31 Diagram" /></td>
</tr>
</tbody>
</table>
FOR PARTS REQUESTS ONLY

- Be sure to follow instructions carefully.
- Use a separate letter for all correspondence.
- Please allow 10 - 14 days for mail delivery time.

DO NOT WRITE IN THIS SPACE

INSTRUCTIONS

- Please print all information requested.
- Be sure you list the correct HEATH part number exactly as it appears in the parts list.
- If you wish to prepay your order, mail this card and your payment in an envelope. Be sure to include 10% (25¢ minimum, $3.50 maximum) for insurance, shipping and handling. Michigan residents add 4% tax.

  Total enclosed $________

- If you prefer COD shipment, check the COD box and mail this form.

  COD □

NAME ____________________________________________
ADDRESS ____________________________________________
CITY ____________________________ ZIP __________
STATE ____________________________

The information requested in the next two lines is not required when purchasing nonwarranty replacement parts, but it can help us provide you with better products in the future.

Model # ____________________________ Invoice # __________
Date ____________________________ Location ____________________________
Purchased ____________________________

LIST HEATH PART NUMBER QTY. PRICE EACH TOTAL PRICE

TOTAL FOR PARTS

HANDLING AND SHIPPING

MICHIGAN RESIDENTS ADD 4% TAX

TOTAL AMOUNT OF ORDER

SEND TO: HEATH COMPANY
BENTON HARBOR
MICHIGAN 49022
ATTN: PARTS REPLACEMENT

Phone (Replacement parts only): 616 982-3571

THIS FORM IS FOR U.S. CUSTOMERS ONLY
OVERSEAS CUSTOMERS SEE YOUR DISTRIBUTOR
CUSTOMER SERVICE

REPLACEMENT PARTS

Please provide complete information when you request replacements from either the factory or Heath Electronic Centers. Be certain to include the HEATH part number exactly as it appears in the parts list.

ORDERING FROM THE FACTORY

Print all of the information requested on the parts order form furnished with this product and mail it to Heath. For telephone orders (parts only) dial 616 982-3571. If you are unable to locate an order form, write us a letter or card including:

- Heath part number.
- Model number.
- Date of purchase.
- Location purchased or invoice number.
- Nature of the defect.
- Your payment or authorization for COD shipment of parts not covered by warranty.

Mail letters to: Heath Company
Benton Harbor
MI 49022
Attn: Parts Replacement

Retain original parts until you receive replacements. Parts that should be returned to the factory will be listed on your packing slip.

OBTAINING REPLACEMENTS FROM HEATH ELECTRONIC CENTERS

For your convenience, “over the counter” replacement parts are available from the Heath Electronic Centers listed in your catalog. Be sure to bring in the original part and purchase invoice when you request a warranty replacement from a Heath Electronic Center.

TECHNICAL CONSULTATION

Need help with your kit? — Self-Service? — Construction? — Operation? — Call or write for assistance. you’ll find our Technical Consultants eager to help with just about any technical problem except “customizing” for unique applications.

The effectiveness of our consultation service depends on the information you furnish. Be sure to tell us:

- The Model number and Series number from the blue and white label.
- The date of purchase.
- An exact description of the difficulty.
- Everything you have done in attempting to correct the problem.

Also include switch positions, connections to other units, operating procedures, voltage readings, and any other information you think might be helpful.

Please do not send parts for testing, unless this is specifically requested by our Consultants.

Hints: Telephone traffic is lightest at midweek — please be sure your Manual and notes are on hand when you call.

Heathkit Electronic Center facilities are also available for telephone or “walk-in” personal assistance.

REPAIR SERVICE

Service facilities are available, if they are needed, to repair your completed kit. (Kits that have been modified, soldered with paste flux or acid core solder, cannot be accepted for repair.)

If it is convenient, personally deliver your kit to a Heathkit Electronic Center. For warranty parts replacement, supply a copy of the invoice or sales slip.

If you prefer to ship your kit to the factory, attach a letter containing the following information directly to the unit:

- Your name and address.
- Date of purchase and invoice number.
- Copies of all correspondence relevant to the service of the kit.
- A brief description of the difficulty.
- Authorization to return your kit COD for the service and shipping charges. (This will reduce the possibility of delay.)

Check the equipment to see that all screws and parts are secured. (Do not include any wooden cabinets or color television picture tubes, as these are easily damaged in shipment. Do not include the kit Manual.) Place the equipment in a strong carton with at least THREE INCHES of resilient packing material (shredded paper, excelsior, etc.) on all sides. Use additional packing material where there are protrusions (control sticks, large knobs, etc.). If the unit weighs over 15 lbs., place this carton in another one with 3/4” of packing material between the two.

Seal the carton with reinforced gummed tape, tie it with a strong cord, and mark it “Fragile” on at least two sides. Remember, the carrier will not accept liability for shipping damage if the unit is insufficiently packed. Ship by prepaid express, United Parcel Service, or insured Parcel Post to:

Heath Company
Service Department
Benton Harbor, Michigan 49022
HEATH ZENITH

HEATH COMPANY • BENTON HARBOR, MICHIGAN
THE WORLD'S FINEST ELECTRONIC EQUIPMENT IN KIT FORM

LITHO IN U.S.A.
<table>
<thead>
<tr>
<th>PART NUMBER</th>
<th>PRICE</th>
<th>PART NUMBER</th>
<th>PRICE</th>
<th>PART NUMBER</th>
<th>PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-106-12</td>
<td>.30</td>
<td>6-255-11</td>
<td>.20</td>
<td>6-443-824</td>
<td>24.50</td>
</tr>
<tr>
<td>6-32-10</td>
<td>.30</td>
<td>6-259-22</td>
<td>.05</td>
<td>6-443-839</td>
<td>4.45</td>
</tr>
<tr>
<td>6-153-10</td>
<td>.30</td>
<td>6-260-66</td>
<td>.15</td>
<td>6-443-840</td>
<td>9.75</td>
</tr>
<tr>
<td>6-181-18</td>
<td>.30</td>
<td>6-265-70</td>
<td>.50</td>
<td>6-443-17</td>
<td>14.55</td>
</tr>
<tr>
<td>6-122-1</td>
<td>.30</td>
<td>6-264-34</td>
<td>.90</td>
<td>6-462-1024</td>
<td>1.45</td>
</tr>
<tr>
<td>6-271-1</td>
<td>.30</td>
<td>6-262-8</td>
<td>.05</td>
<td>6-475-12</td>
<td>8.10</td>
</tr>
<tr>
<td>6-272-1</td>
<td>.30</td>
<td>6-344-91</td>
<td>.05</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>6-472-12</td>
<td>.30</td>
<td>6-344-53</td>
<td>.50</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>6-482-1</td>
<td>.25</td>
<td>6-344-54</td>
<td>.50</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>6-282-1</td>
<td>.30</td>
<td>6-344-73</td>
<td>.05</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>6-282-1</td>
<td>.30</td>
<td>6-344-73</td>
<td>.05</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>20-196</td>
<td>.39</td>
<td>6-347-97</td>
<td>.05</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>20-700</td>
<td>.95</td>
<td>6-347-11</td>
<td>.10</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>25-221</td>
<td>1.00</td>
<td>6-347-17</td>
<td>.15</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>25-241</td>
<td>1.55</td>
<td>39-1255</td>
<td>.15</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>25-272</td>
<td>2.00</td>
<td>39-1309</td>
<td>1.45</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>25-272</td>
<td>2.00</td>
<td>39-1309</td>
<td>1.45</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>25-272</td>
<td>2.00</td>
<td>39-1309</td>
<td>1.45</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>55-490</td>
<td>11.50</td>
<td>39-1404</td>
<td>1.85</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>55-490</td>
<td>11.50</td>
<td>39-1404</td>
<td>1.85</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>55-490</td>
<td>11.50</td>
<td>39-1404</td>
<td>1.85</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>55-490</td>
<td>11.50</td>
<td>39-1404</td>
<td>1.85</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>55-490</td>
<td>11.50</td>
<td>39-1404</td>
<td>1.85</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>60-490</td>
<td>2.00</td>
<td>43-1-42</td>
<td>.15</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>60-490</td>
<td>2.00</td>
<td>43-1-42</td>
<td>.15</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>60-490</td>
<td>2.00</td>
<td>43-1-42</td>
<td>.15</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>60-490</td>
<td>2.00</td>
<td>43-1-42</td>
<td>.15</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
<tr>
<td>60-490</td>
<td>2.00</td>
<td>43-1-42</td>
<td>.15</td>
<td>6-490-111</td>
<td>.15</td>
</tr>
</tbody>
</table>

* Additional 3 ft rolls of solder, #331-6, can be ordered for 25 cents each.

---

**Write Heath Company for price information.**

*Price per foot.*
PICTORIAL 1-7

PARTIAL SCHEMATIC

Model ETA-3400/EWA-3400

Page 1
PICTORIAL 4-1
KEY NAMES/FUNCTIONS

D-DO: Enter letter "D" or do program at address to be entered.

B-Back: Enter letter "B," or decrement displayed memory address.

A-Auto: Enter letter "A," or enable automatic program loading.

7-RTI: Enter numeral "7," or resume user's program.

5-CC: Enter numeral "5," or display condition code register.

4-INDEX: Enter numeral "4," or display index register.

1-ACCA: Enter numeral "1," or display Accumulator A.

0: Enter numeral "0," (zero).

2-ACCB: Enter numeral "2," or displays Accumulator B.

E-EXAM: Enter letter "E," or request address to be examined.

F-FWD: Enter letter "F," or increment displayed memory address.

C-CHAN: Enter letter "C," or request change of address or data.

8-SS: Enter numeral "8," or single step user's program.

9-BR: Enter numeral "9," or permit entry of break points.

6-SP: Enter numeral "6," or display stack pointer.

3-PC: Enter numeral "3," or display program counter.

RESET: Reset system for new operation.
POWER switch (SW1) - Selects either the STANDBY or ON position. Memory never dumps while the line cord is connected to AC power.

LED1 - Indicates when the line cord is plugged into AC power.

Provides input and output control of data transceivers.

Provides connections to system data lines.

Provides outputs from 02 clock, VMA02 clock, line, and 1Hz square wave sources.

LED's Display status of logic inputs.

Provides inputs for LED's.

Provides connections to microprocessor control lines.

Provides outputs for INPUT SWITCHES.

INPUT SWITCHES - Provide logic 1's and 0's to 8-pin connectors.

Supplies connections to +12 volts, -12 volts, +5 volts, and ground.

Connectors are internally connected together.
microcomputer learning system

LED DISPLAY - Displays information as directed by the microprocessor.

SEGMENT TEST - When shorted together, the LED DISPLAY will show all eights.

Provides outputs from buffered address lines.

KEYBOARD - Allows you to enter data or commands.

TERMINAL BLOCK - Use this to make solderless connections. Do not insert wires or leads larger than #20 (0.032").

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

PICTORIAL 8-1
<table>
<thead>
<tr>
<th>RAM 00XX</th>
<th>RAM 01XX</th>
<th>FFXX</th>
<th>ROM FFXX</th>
<th>KEYBOARD C0-X</th>
<th>DISPLAY LED's C1XX</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
<td>0 1 1</td>
<td>1 1 0 1 1</td>
<td>0 1 1 0 1 0 1 0 1</td>
<td>0 1 1 0 1 0 1 0 1</td>
</tr>
</tbody>
</table>

### KEYBOARD COLUMN ADDRESS DECODER

<table>
<thead>
<tr>
<th>COLUMN ADDRESS</th>
<th>X</th>
<th>PRESS KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO - 3</td>
<td>0 1 1</td>
<td>F  C 9  6 3</td>
</tr>
<tr>
<td>CO - 5</td>
<td>1 0 1</td>
<td>E  B 8  5 2</td>
</tr>
<tr>
<td>CO - 6</td>
<td>1 1 0</td>
<td>D  A 7  4 1 0</td>
</tr>
</tbody>
</table>

0 = LOGIC 0  
1 = LOGIC 1  
- = DOES NOT CARE  
X = FUNCTIONING ADDRESS
**IC DECODING CHART**

<table>
<thead>
<tr>
<th>IC21</th>
<th>1, 3, 12, 11</th>
<th>IC20</th>
<th>INPUT</th>
<th>12 13 14 15</th>
<th>OUTPUT</th>
<th>5 6 10 11</th>
<th>END RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 1 0 1 0 1 0</td>
<td>1</td>
<td>0 1 0 0</td>
<td>0 1 1 1</td>
<td>1</td>
<td>IC14 and IC15, Pin 13 is 0.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0 1 0 1 0 1 0</td>
<td>1</td>
<td>0 1 0 1</td>
<td>1 0 1 1</td>
<td>1</td>
<td>IC16 and IC17, Pin 13 is 0.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0 1 0 1 0 1 0</td>
<td>1</td>
<td>1 1 0 0</td>
<td>1 1 1 1</td>
<td>1</td>
<td>IC12 pins 10, 13, and 14 are 1. Pin 11 is 0.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0 1 0 1 0 1 0</td>
<td>0</td>
<td>1 0 0 0</td>
<td>1 1 0 1</td>
<td>1</td>
<td>IC13 pins 1 and 15 are 0.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0 1 0 1 0 1 0</td>
<td>0</td>
<td>1 0 0 1</td>
<td>1 1 1 0</td>
<td>1</td>
<td>IC22 pin 12 is 0.</td>
<td></td>
</tr>
</tbody>
</table>

**ADDRESS DECODING CHART**

<table>
<thead>
<tr>
<th>PRESS KEY</th>
<th>LOGIC 0 ON DATA LINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>D₀</td>
</tr>
<tr>
<td>C</td>
<td>D₁</td>
</tr>
<tr>
<td>9</td>
<td>D₂</td>
</tr>
<tr>
<td>6</td>
<td>D₃</td>
</tr>
<tr>
<td>3</td>
<td>D₄</td>
</tr>
<tr>
<td>E</td>
<td>D₀</td>
</tr>
<tr>
<td>B</td>
<td>D₁</td>
</tr>
<tr>
<td>8</td>
<td>D₂</td>
</tr>
<tr>
<td>5</td>
<td>D₃</td>
</tr>
<tr>
<td>2</td>
<td>D₄</td>
</tr>
<tr>
<td>D</td>
<td>D₀</td>
</tr>
<tr>
<td>A</td>
<td>D₁</td>
</tr>
<tr>
<td>7</td>
<td>D₂</td>
</tr>
<tr>
<td>4</td>
<td>D₃</td>
</tr>
<tr>
<td>1</td>
<td>D₄</td>
</tr>
<tr>
<td>0</td>
<td>D₅</td>
</tr>
</tbody>
</table>
**DISPLAY LED C1**

<table>
<thead>
<tr>
<th>X</th>
<th>Input Pin</th>
<th>Logic output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A7 A6 A5 A4</td>
<td>12 13 14 15</td>
</tr>
<tr>
<td>LED C1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 0</td>
<td>0 1 1 0</td>
<td>7</td>
</tr>
<tr>
<td>1 0 1</td>
<td>0 1 0 1</td>
<td>6</td>
</tr>
<tr>
<td>1 0 0</td>
<td>0 1 0 0</td>
<td>5</td>
</tr>
<tr>
<td>0 1 1</td>
<td>0 0 1 1</td>
<td>4</td>
</tr>
<tr>
<td>0 1 0</td>
<td>0 0 1 0</td>
<td>3</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 0 0 1</td>
<td>2</td>
</tr>
</tbody>
</table>

*With a given output pin addressed, the logic level on that addressed IC.*

**LED segment C1X**

<table>
<thead>
<tr>
<th>X</th>
<th>IC23 through IC28, input pins 3, 2, 1</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A3 A2 A1 A0</td>
<td>1 1 0</td>
<td>1 1 0</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>1 1 0</td>
<td>1 0 1</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>1 1 0</td>
<td>1 0 0</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>1 1 0</td>
<td>0 1 1</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>1 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>1 1 0</td>
<td>0 0 1</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>1 1 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>1 1 0</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>

**D9 LOGIC LEVEL CHART**

<table>
<thead>
<tr>
<th>D9 logic levels</th>
<th>IC21</th>
<th>IC6</th>
</tr>
</thead>
<tbody>
<tr>
<td>D9 Pins 9 and 10</td>
<td>Pin 2</td>
<td>Pin 2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
# SPLAY LED CHART

<table>
<thead>
<tr>
<th>Pin</th>
<th>Logic 0 at output pin</th>
<th>Logic 0 on enable pin 14 of IC</th>
<th>Display LED addressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>14  15</td>
<td>1 0</td>
<td>7</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>0 1</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>0 0</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>1 1</td>
<td>4</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>1 0</td>
<td>3</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>0 1</td>
<td>2</td>
<td>28</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>through input 2, 1</th>
<th>IC23 through IC28* Output pin</th>
<th>LED pin</th>
<th>Segment*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0</td>
<td>11</td>
<td>1</td>
<td>a</td>
</tr>
<tr>
<td>0 1</td>
<td>10</td>
<td>13</td>
<td>b</td>
</tr>
<tr>
<td>0 0</td>
<td>9</td>
<td>10</td>
<td>c</td>
</tr>
<tr>
<td>1 1</td>
<td>7</td>
<td>8</td>
<td>d</td>
</tr>
<tr>
<td>1 0</td>
<td>6</td>
<td>7</td>
<td>e</td>
</tr>
<tr>
<td>0 1</td>
<td>5</td>
<td>2</td>
<td>f</td>
</tr>
<tr>
<td>0 0</td>
<td>4</td>
<td>11</td>
<td>g</td>
</tr>
<tr>
<td>1 1</td>
<td>12</td>
<td>9</td>
<td>DP</td>
</tr>
</tbody>
</table>

Logic level on that pin will follow the level on the D input of the

# IC LEVEL CHART

<table>
<thead>
<tr>
<th>IC6</th>
<th>IC23 through IC28</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin 2</td>
<td>Pin 18</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Page 11
BLOCK DIAGRAM
CIRCUIT BOARD X-RAY VIEW

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

A. Find the circuit component number (R5, C3, etc.) on the “X-Ray View.”

B. Locate this same number in the “Circuit Component Number” column of the “Parts List.”

C. Adjacent to the circuit component number, you will find the PART NUMBER and DESCRIPTION which must be supplied when you order a replacement part.

Top foil in red

(Shown from bottom side)