HEATHKIT® MANUAL

for the

MICROPROCESSOR TRAINER

Model ET-3400A

595-2610-04

HEATH COMPANY • BENTON HARBOR, MICHIGAN

EBAMAN.COM

preserving our technical legacy

This manual is provided FREE OF CHARGE from EBAMAN.COM as a service to the technical community.

It was uploaded by someone who wanted to help you repair and maintain your equipment.

If you paid for this manual, you paid someone who is making a profit from the free labor of others without asking their permission.

Thousands of files are available without charge from: EBAMAN.COM

Visit us at http://ebaman.com

This file was provided by:
Jerry Ingordo W2JI

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 — Hardware	(616) 982-3309
Computers — Software:	
Operating Systems, Languages, Utilities	(616) 982-3860
Application Programs	(616) 982-3884
Heath Craft Wood Works	(616) 982-3423

YOUR HEATHKIT 90-DAY LIMITED WARRANTY

Consumer Protection Plan for Heathkit Consumer Products

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

Heath's Responsibility

PARTS — Replacements for factory defective parts will be supplied free for 90 days from date of purchase. Replacement parts are warranted for the remaining portion of the original warranty period. You can obtain warranty parts direct from Heath Company by writing or telephoning us at (616) 982-3571. And we will pay shipping charges to get those parts to you . . . anywhere in the world.

SERVICE LABOR — For a period of 90 days from the date of purchase, any malfunction caused by defective parts or error in design will be corrected at no charge to you. You must deliver the unit at your expense to the Heath factory, any Heathkit Electronic Center (units of Veritechnology Electronics Corporation), or any of our authorized overseas distributors.

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

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

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

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

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.

Owner's Responsibility

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

ASSEMBLY — Before seeking warranty service, you should complete the assembly by carefully following the manual instructions. Heathkit service agencies cannot complete assembly and adjustments that are customer's responsibility.

ACCESSORY EQUIPMENT — Performance malfunctions involving other non-Heath accessory equipment, (antennas, audio components, computer peripherals and software, etc.) are not covered by this warranty and are the owner's responsibility.

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

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

Heathkit® Manual

for the

MICROPROCESSOR TRAINER

Model ET-3400A

595-2610-04

HEATH COMPANY
BENTON HARBOR, MICHIGAN 49022

Copyright © 1981
Heath Company
All Rights Reserved
Printed in the United States of America

WARNING

Federal Communications Commission requirements prescribe certification of personal computers and any interconnected peripherals in Part 15 Subpart J of the rules and regulations. This computing device will meet these requirements when constructed in strict accordance with the instructions in this manual, using only components and materials supplied with the kit or the exact equivalent thereof. You will be instructed to sign and date the enclosed FCC ID label and affix the label to the equipment certifying that you have constructed this equipment in accordance with the above mentioned instructions. In order to meet legal requirements, be certain to follow the instructions exactly as they are stated in this manual.

This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

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

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

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

TABLE OF CONTENTS

Introduction 4

Assembly Notes 5

Parts List 8

Step-by-Step Assembly 11
- Main Circuit Board 11
- Keyboard Circuit Board 28
- Support Bracket Assembly 32
- Cabinet Assembly and Wiring 36

Initial Tests 43
- Voltage Tests 44
- Tests Continued 44
- Operational Tests 47

Final Assembly 52

Operation 56
- Keyboard 56
- Entering Programs 60
- Using Breakpoints 62
- Sample Programs 67
- Subroutine Flow Charts 77
- Monitor Listings 85
- Reserved Memory Bytes in RAM 100
- Memory 101
- Instruction Set* 103

In Case of Difficulty 105
- General 106
- Troubleshooting Charts 107
- Detailed Troubleshooting 108

Specifications 121

Theory of Operation 122

Semiconductor Identification Charts 124

Circuit Board X-Ray Views
..... Illustration Booklet, Page 14

Schematic ..... Fold-in

Warranty ..... Inside front cover

Customer Service ..... Inside rear cover

*Copied with permission of Motorola

INTRODUCTION

The Model ET-3400A Microcomputer Learning System is a practical, low-cost Microprocessor Trainer that is especially designed as a learning tool to teach microprocessor operation, programming, and applications. The ET-3400A 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.

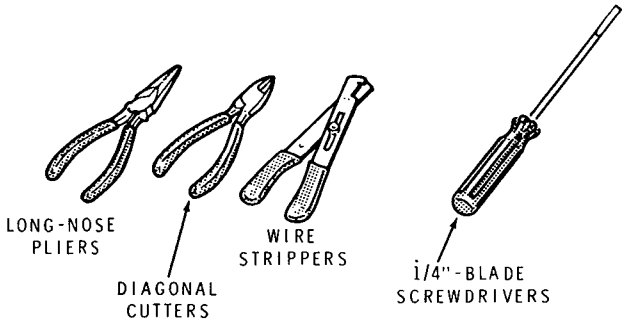
MAIN FEATURES

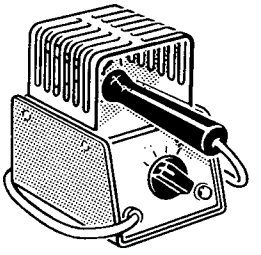
- The popular 6808 Microprocessor.
- 512 bytes of semiconductor RAM.
- A 1k ROM monitor program.
- A hexadecimal keyboard for rapid data and program entry.
- Six digits of hexadecimal display for reading out memory addresses, their contents, and register contents.
- Breadboarding sockets that permit rapid, solderless assembly of IC logic circuitry to be used with the microprocessor. Ideal for prototyping special interface circuits.
- Microprocessor address bus, data bus, control lines, and associated signals that are buffered and terminated on front panel connectors; allowing you complete freedom to experiment with the microprocessor and its associated circuitry.
- Eight individual, independent, binary LED indicators for monitoring logic states in the breadboard circuitry.
- Eight individual, independent, binary data switches you can use to supply binary words and logic levels in the breadboarding circuitry.
- Built-in power supplies that 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.
- Provision for future expansion of memory and I/O capabilities.

ASSEMBLY NOTES

TOOLS

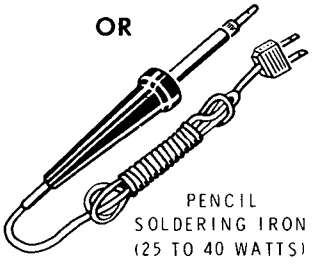
You will need these tools to assemble your kit.





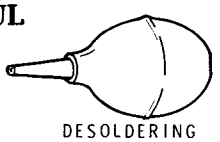
HEATHKIT SOLDERING IRON

OR



PENCIL SOLDERING IRON (25 TO 40 WATTS)

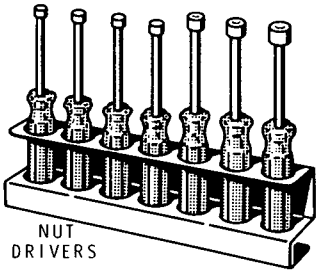
OTHER HELPFUL TOOLS



DESOLDERING BULB*

NUT STARTER (MAY BE SUPPLIED WITH KIT)

DESOLDERING BRAID*



NUT DRIVERS

*TO REMOVE SOLDER FROM CIRCUIT CONNECTIONS.

ASSEMBLY

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 excessive 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:

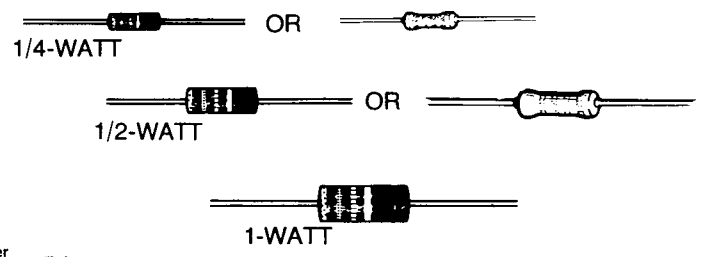
1. Use the right type of soldering iron. A 25 to 40-watt pencil soldering iron with a 1/8" or 3/16" chisel or pyramid tip works best.

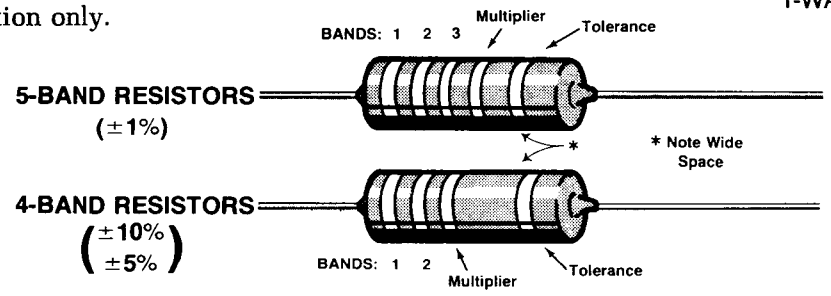
2. Keep the soldering iron tip clean. Wipe it often on a wet sponge or cloth; then apply solder to the tip to give the entire tip a wet look. This process is called tinning, and it will protect the tip and enable you to make good connections. When solder tends to "ball" or does not stick to the tip,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

NOTE: Always use rosin core, radio-type solder (60:40 or 50:50 tin-lead content) for all of the soldering in this kit. This is the type we have supplied with the parts. The Warranty will be void and we will not service any kit in which acid core solder or paste has been used.

PARTS

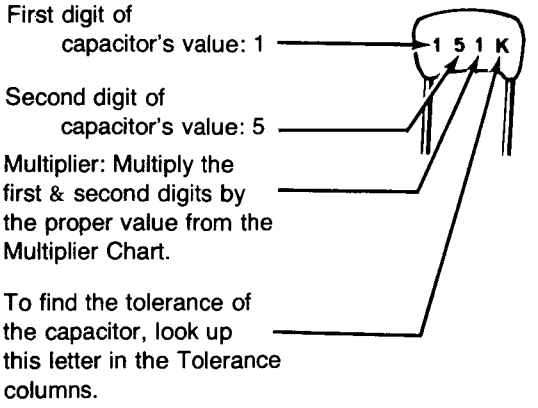
Resistors will be called out by their resistance value in Ω (ohms), kΩ (kilohms), or MΩ (megohms). Certain types of resistors will have the value printed on the body, while others will be identified by a color code. The colors of the bands and the value will be given in the steps, therefore the following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





Band 1 1st Digit		Band 2 2nd Digit		Band 3 (if used) 3rd Digit		Multiplier		Resistance Tolerance	
Color	Digit	Color	Digit	Color	Digit	Color	Multiplier	Color	Tolerance
Black	0	Black	0	Black	0	Black	1	Silver	±10%
Brown	1	Brown	1	Brown	1	Brown	10	Gold	± 5%
Red	2	Red	2	Red	2	Red	100	Brown	± 1%
Orange	3	Orange	3	Orange	3	Orange	1,000		
Yellow	4	Yellow	4	Yellow	4	Yellow	10,000		
Green	5	Green	5	Green	5	Green	100,000		
Blue	6	Blue	6	Blue	6	Blue	1,000,000		
Violet	7	Violet	7	Violet	7	Silver	0.01		
Gray	8	Gray	8	Gray	8	Gold	0.1		
White	9	White	9	White	9				

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:



EXAMPLES:

151K = 15 × 10 = 150 pF
759 = 75 ×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).

MULTIPLIER		TOLERANCE OF CAPACITOR		
FOR THE NUMBER:	MULTIPLY BY:	10 pF OR LESS	LETTER	OVER 10 pF
0	1	±0.1 pF	B	
1	10	±0.25 pF	C	
2	100	±0.5 pF	D	
3	1000	±1.0 pF	F	±1%
4	10,000	±2.0 pF	G	±2%
5	100,000		H	±3%
			J	±5%
8	0.01		K	±10%
9	0.1		M	±20%

*DuPont Registered Trademark

PARTS LIST

( ) Unpack the kit and check each part against the following list. The key numbers correspond to the numbers on the "Parts Pictorial" (Illustration Booklet, Pages 1 and 2). Return any part that is packed in an individual envelope, with the part number on it, to its envelope after you identify it until that part is called for in a step. Do not throw away any packing material until you account for all the parts.

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

RESISTORS

NOTES:

1. The following resistors have a tolerance of 5% unless otherwise listed. A 5% tolerance is indicated by a gold fourth color band.
2. Resistors may be packed in more than one envelope. Open all of the resistor envelopes before you check the resistors against the following list.

1/4-Watt Resistors

KEY No.	HEATH Part No.	QTY.	DESCRIPTION	CIRCUIT Comp. No.
A1	6-151-12	9	150 Ω (brn-grn-brn)	R2, R36, R37, R38, R39, R41, R42, R43, R44
A1	6-302-12	2	3000 Ω (org-blk-red)	R53, R54
A1	6-472-12	9	4700 Ω (yel-viol-red)	R18, R19, R21, R22, R23, R24, R25, R26, R56
A1	6-682-12	1	6800 Ω (blu-gry-red)	R6
A1	6-822-12	26	8200 Ω (gry-red-red)	R5, R12, R17, R27, R28, R29, R31, R32, R33, R34, R35, R45, R46, R47, R48, R49, R51, R52, R55, R57, R101, R102, R103, R104, R105, R106
A1	6-153-12	1	15 kΩ (brn-grn-org)	R8
A1	6-273-12	1	27 kΩ (red-viol-org)	R1
A1	6-104-12	3	100 kΩ (brn-blk-yel)	R13, R14, R16

1/4-Watt Resistors (cont'd.)

KEY No.	HEATH Part No.	QTY.	DESCRIPTION	CIRCUIT Comp. No.
A1	6-154-12	1	150 kΩ (brn-grn-yel)	R11
A1	6-224-12	2	220 kΩ (red-red-yel)	R7, R9
A1	6-824-12	1	820 kΩ (gry-red-yel)	R15

Other Resistors

KEY No.	HEATH Part No.	QTY.	DESCRIPTION	CIRCUIT Comp. No.
A2	6-680	2	68 Ω, 1/2-watt (blu-gry-blk)	R3, R4
A3	9-117	6	Resistor pack	RP1, RP2, RP3, RP4, RP5, RP6

CAPACITORS

Electrolytic Capacitors

KEY No.	HEATH Part No.	QTY.	DESCRIPTION	CIRCUIT Comp. No.
B1	25-200	1	.68 μF tantalum	C16
B1	25-221	2	2.2 μF tantalum	C12, C13
B1	25-220	2	10 μF tantalum (10M)	C14, C15
B2	25-918	1	100 μF	C17
B3	25-876	2	1000 μF	C9, C11
B3	25-272	1	6000 μF	C4

Other Capacitors

KEY No.	HEATH Part No.	QTY.	DESCRIPTION	CIRCUIT Comp. No.
B4	21-743	2	27 pF ceramic	C18, C19
B4	21-71	2	.001 μF (1000 pF) ceramic	C2, C3
B5	21-769	11	.01 μF (103) glass ceramic	C7, C8, C21, C22, C23, C24, C25, C26, C27, C28, C29
B6	27-127	1	.047 μF Mylar	C1
B7	27-85	2	.22 μF Mylar	C5, C6

NOTE: These glass ceramic capacitors look like some diodes. Leave them in the bag until they are called for in a step.

Heathkit®

KEY No.	HEATH Part No.	QTY.	DESCRIPTION	CIRCUIT Comp. No.

DIODES

C1	56-56	4	1N4149	D8, D18, D19, D20
C1	56-89	1	GD510	D17
C1	57-42	2	3A1	D1, D2
C1	57-65	4	1N4002	D4, D5, D6, D7

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. (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 these IC's from the foam until a step directs you to do so.

D1	442-30	1	μA309K	U31
D3	442-616	1	LM3302N, LM2901N, or μA775	U18
D2	442-644	1	78L12	U29
D2	442-646	1	79L12	U30
D3	443-897	1	74S04	U16
D3	443-26	1	74S00	U21
D3	443-45	1	7408	U5
D3	443-717	1	74126	U4
D3	443-720	1	40097	U13
D3	443-754	1	74LS240	U1
D3	443-764	2	2114	U14, U15
D3	443-804	6	74LS259	U23, U24, U25, U26, U27, U28
D3	443-807	4	74LS42	U2, U3, U20, U22
D3	443-824	3	74LS241	U6, U7, U8
D3	443-839	2	74LS243	U9, U10
D3	443-939	1	MC6808P	U11
D3	444-76	1	ROM	U12

SWITCHES — INSULATORS

E1	60-34	1	Rocker switch	SW1
E2	60-621	1	Switch assembly (may be slide or rocker switches)	SW2
E3	64-839	17	Pushbutton switch (may look different than the one shown)	SW101, SW102, SW103, SW104, SW105, SW106, SW107, SW108, SW109, SW110, SW111, SW112, SW113, SW114, SW115, SW116, SW117
E4	73-4	1	Rubber grommet	
E5	75-724	1	Insulator plate	
E6	75-788	2	Insulator paper	

HARDWARE

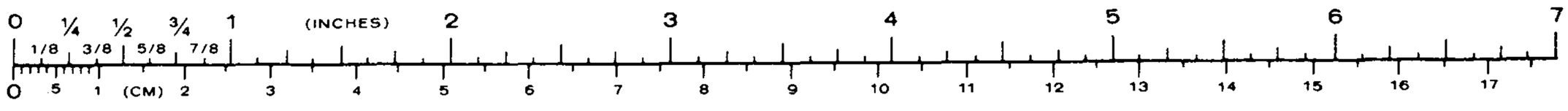
NOTE: Hardware may be packed in more than one packet. Open all of the hardware packets before you check the hardware against the following list.

F1	250-163	3	#4 × 5/16″ self-tapping screw
F2	250-138	2	6-32 × 3/16″ screw
F3	250-56	13	6-32 × 1/4″ screw
F4	250-475	10	#6 × 3/8″ hex head screw
F5	250-32	1	6-32 × 3/8″ flat head screw
F6	250-162	2	6-32 × 1/2″ screw
F7	250-559	8	#6 × 5/8″ self-tapping screw
F8	250-1137	2	#6 × 1-1/8″ self-tapping screw
F9	252-3	6	6-32 nut
F10	254-1	13	#6 lockwasher
F11	255-23	4	Spacer
F12	259-1	4	#6 solder lug

WIRE — BRAID — LINE CORD

89-54	1	Line cord
344-15	9″	Black wire
344-51	18″	Brown wire
344-52	3″	Red wire
344-53	9-1/2″	Orange wire
344-54	20′	Yellow wire
344-71	18″	White-brown wire
344-73	9-1/2″	White-orange wire
344-74	9″	White-yellow wire
344-99	18″	White wire
345-1	6″	Flat braid
346-7	6″	Clear sleeving
346-21	2″	Teflon* sleeving
346-46	2″	Black heat-shrinkable sleeving

*DuPont Registered Trademark



TERMINAL STRIPS — CONNECTORS — SOCKETS

KEY No.	HEATH Part No.	QTY.	DESCRIPTION	CIRCUIT Comp. No.
G1	431-2	1	2-lug terminal strip	
G2	431-44	1	4-lug terminal strip	
G3	432-874	13	4-pin connector block	
G4	432-875	1	Large connector block	
G5	432-921	2	3-pin IC socket	
G6	432-973	11	8-pin connector block	
G7	434-253	1	40-pin IC socket	
G7	434-298	13	14-pin IC socket	
G7	434-299	11	16-pin IC socket	
G7	434-307	1	24-pin IC socket	
G7	434-310	2	18-pin IC socket	
G7	434-311	4	20-pin IC socket	
G8	434-336	1	TO-3 socket	

CIRCUIT BOARDS — CABINET — BRACKET

	85-2010-1	1	Keyboard circuit board	
	85-2712-1	1	Main circuit board	
H1	92-737	1	Cabinet top	
H2	92-738	1	Cabinet bottom	
H3	204-2291	1	Support bracket	
H4	204-2539	1	Strain relief plate	

LIGHT — EMITTING DIODES (LED's) — FUSE

J1	411-831	6	7-segment LED	V1, V2, V3, V4, V5, V6
J2	412-640	9	Red LED	D3, D9, D10, D11, D12, D13, D14, D15, D16
J3	421-42	1	3/8-ampere, 3AG, slow-blow fuse	F1

MISCELLANEOUS

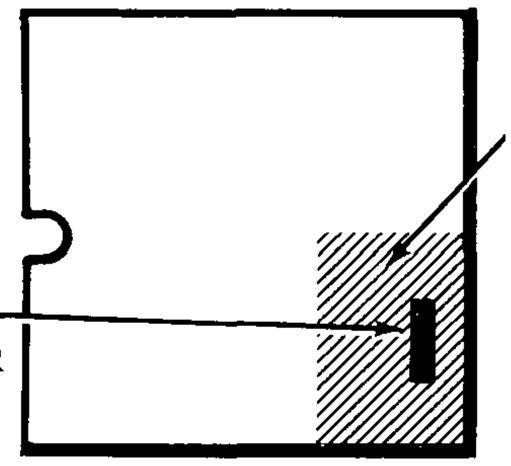
KEY No.	HEATH Part No.	QTY.	DESCRIPTION	CIRCUIT Comp. No.
K1	45-615	1	RF choke	FL1
K2	54-990	1	Power transformer	T1
K3	75-29	1	Strain relief	
K4	260-700	1	LED grommet	
K5	261-34	4	Foot	
K6	262-13	2	Terminal pin	
K7	352-13	1	Silicone grease	
K8	354-7	1	Cable tie	
K9	404-536	1	4 MHz crystal	Y1
K10	423-11	1	Fuseholder	
K11	462-1023	17	Square knob	
K12	475-12	1	Ferrite bead	
K13	490-111	1	IC lifter	
			Solder	

PRINTED MATERIAL

	74-34	1	Double-stick tape (1/2" × 4")
L1	390-1391	1	Heathkit label
L2	390-1395	1	Keyboard label set
	390-1404	1	Red label set
L3	390-1855	1	FCC label
L4	390-1921	1	Power label
L5		1	Blue and white label
	597-260	1	Parts Order Form
		1	Assembly Manual (See Page 1 for part number.)

STEP-BY-STEP ASSEMBLY

MAIN CIRCUIT BOARD



IDENTIFICATION DRAWING

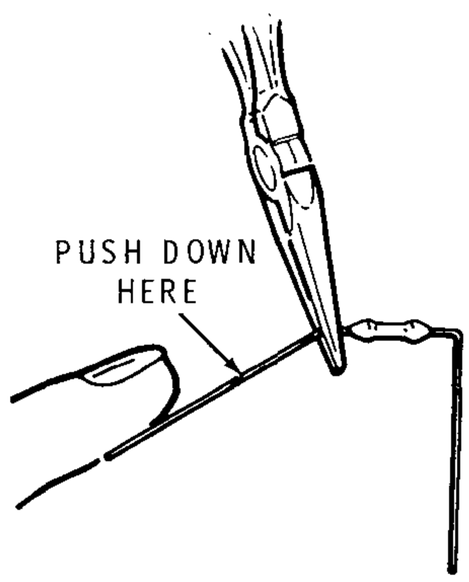
START

NOTE: Due to its large size, only a part of the circuit board is shown. The shaded area in the identification drawing at the top of the Page indicates the area you are assembling.

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 procedure whenever you install parts on a circuit board.

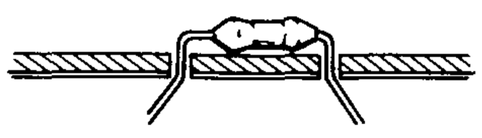
( ) Position the main circuit board as shown in the identification drawing with the printed (not the foil) side up.

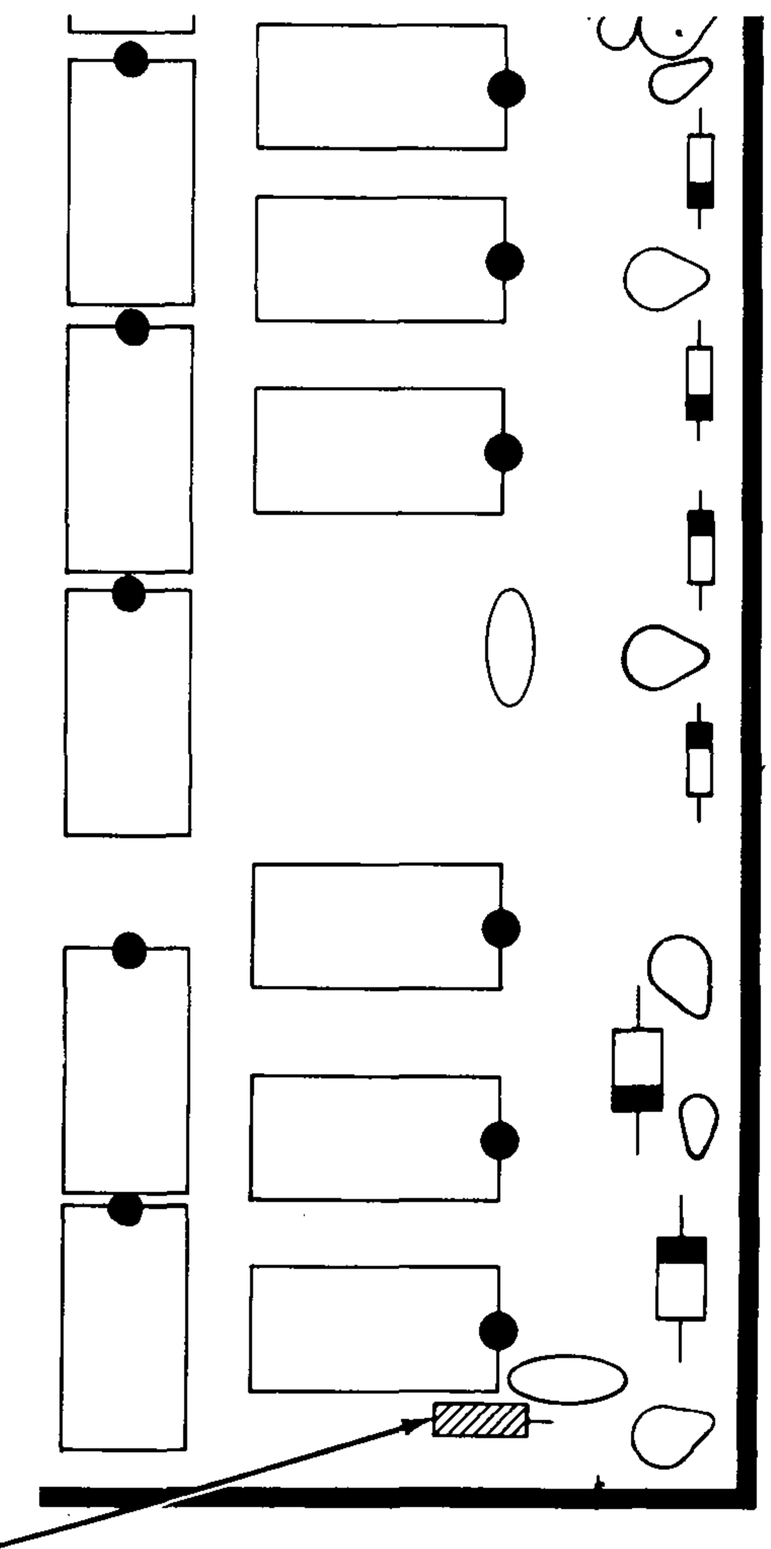
( ) R56: Hold a 4700 Ω (yel-viol-red) resistor with long-nose pliers as shown and bend the leads straight down to fit the hole spacing on the circuit board.



( ) 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 circuit board. Then bend the leads outward slightly to hold the resistor in place.



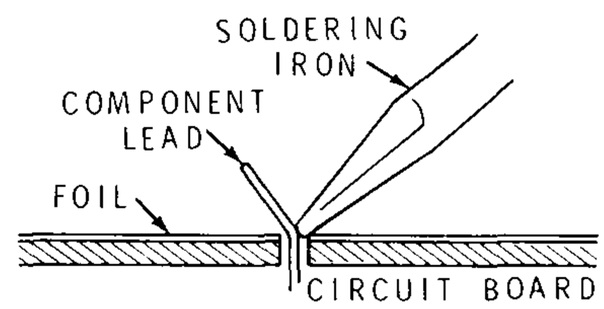


PICTORIAL 1-1

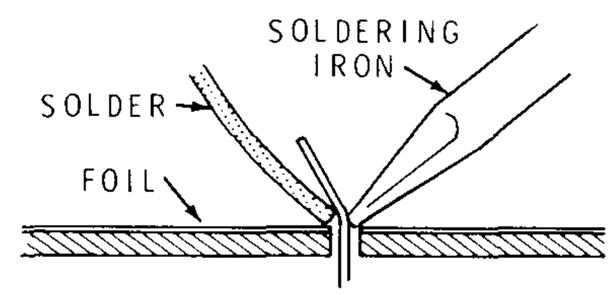
CONTINUE

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

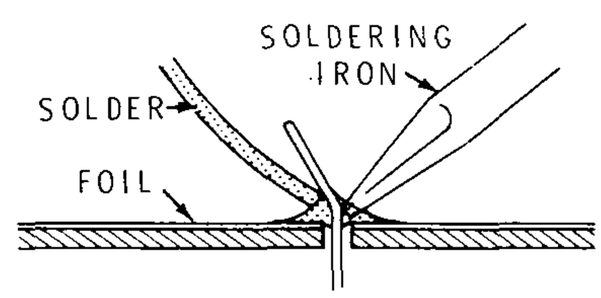
1. Push the soldering iron tip against both the lead and the circuit board foil. Heat **both** for two or three seconds.



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



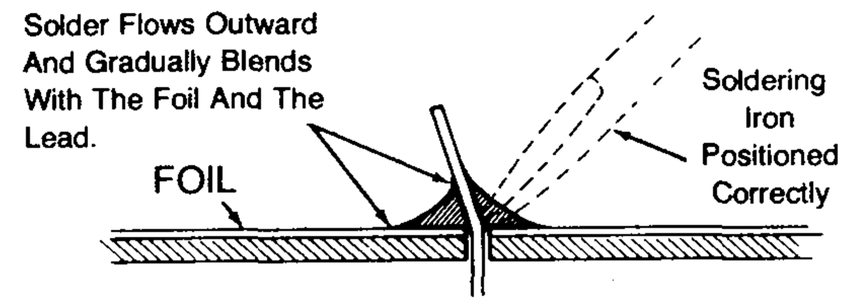
3. As the solder begins to melt, allow it to flow around the connection. Then remove the solder and the iron and let the connection cool.



( ) Cut off the excess lead lengths close to the connection. WARNING: Clip the leads so the ends will not fly toward your eyes.

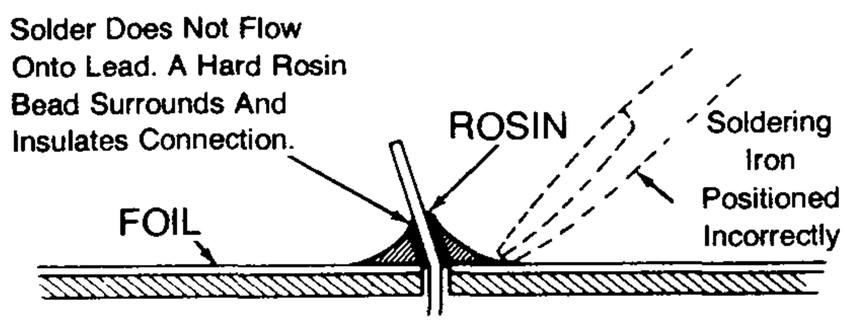
( ) Check each connection. Compare it to the illustrations on Page **12**. After you have checked the solder connections, proceed with the assembly on Page **13**. Use the same soldering procedure for each connection.

A GOOD SOLDER CONNECTION

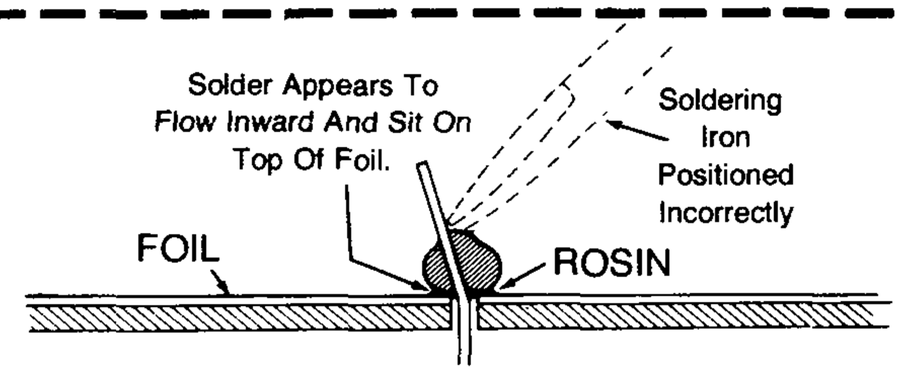


When you heat the lead and the circuit board foil at the same time, the solder will flow evenly onto the lead and the foil. The solder will make a good electrical connection between the lead and the foil.

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.

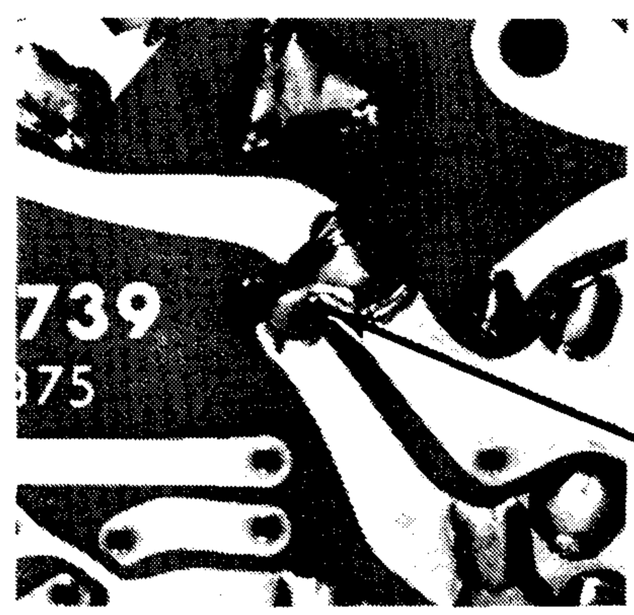


When the foil is not heated sufficiently the solder will blob on the circuit board as shown above. To correct, reheat the connection and, if necessary, apply a small amount of additional solder to obtain a good connection.

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.

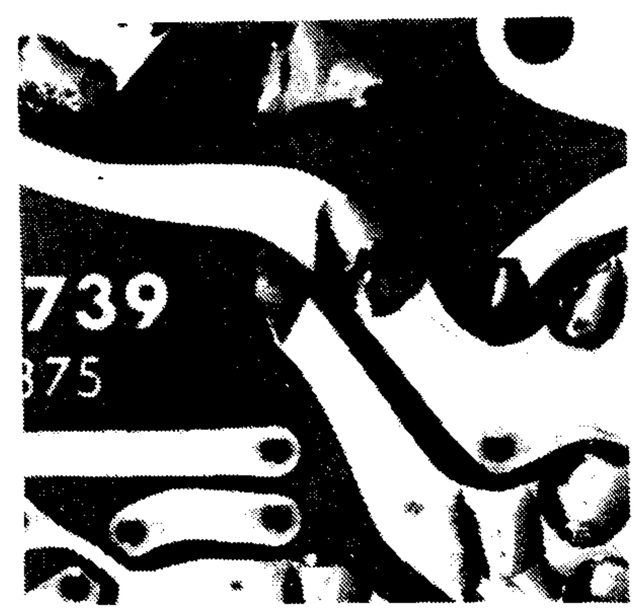
Use only enough solder to make a good connection, and lift the soldering iron straight up from the circuit board. If a solder bridge should develop, turn the circuit board foil-side-down and heat the solder between connections. The excess solder will run onto the tip of the soldering iron, and this will 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.

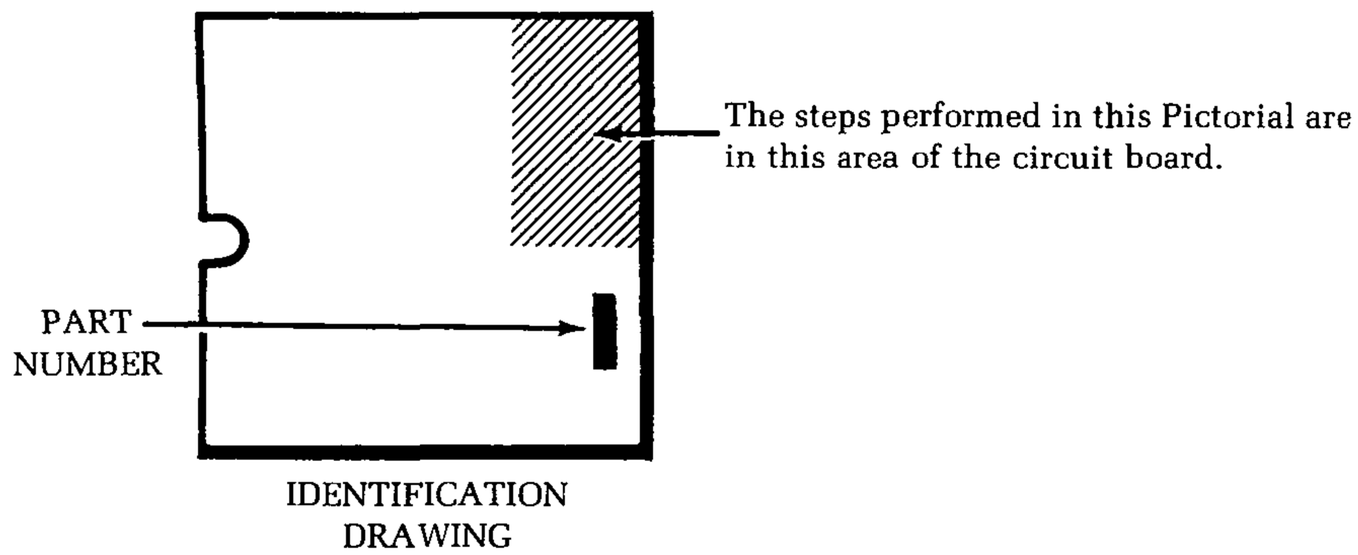


A

SOLDER BRIDGE

B





START

NOTE: Make sure you installed resistor R56 in Pictorial 1-1.

( ) R8: 15 kΩ (brn-grn-org).

( ) R11: 150 kΩ (brn-grn-yel).

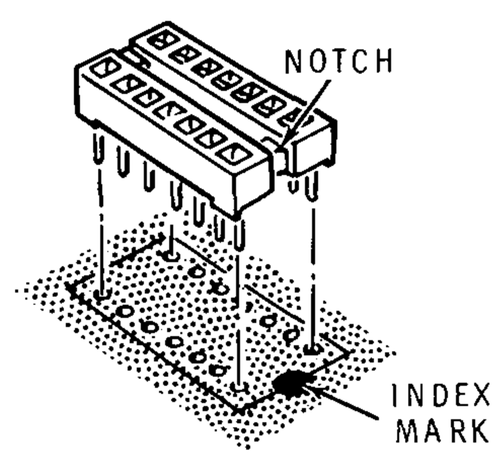
( ) R12: 8200 Ω (gry-red-red).

( ) R9: 220 kΩ (red-red-yel).

( ) R7: 220 kΩ (red-red-yel).

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

You will install IC sockets in the following steps. Be sure the socket pins are straight and then install the socket on the circuit board so the index mark on the board can still be seen. Turn the circuit board over, be sure there is a pin in each of the holes, and then solder the pins to the foil.



( ) 14-pin IC socket at U18.

( ) 16-pin IC socket at U20.

( ) 14-pin IC socket at U21.

( ) 16-pin IC socket at U22.

CONTINUE

( ) R2: 150 Ω (brn-grn-brn).

( ) R17: 8200 Ω (gry-red-red).

( ) R15: 820 kΩ (gry-red-yel).

( ) R16: 100 kΩ (brn-blk-yel).

( ) R14: 100 kΩ (brn-blk-yel).

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

( ) R13: 100 kΩ (brn-blk-yel).

( ) R5: 8200 Ω (gry-red-red).

( ) R6: 6800 Ω (blu-gry-red).

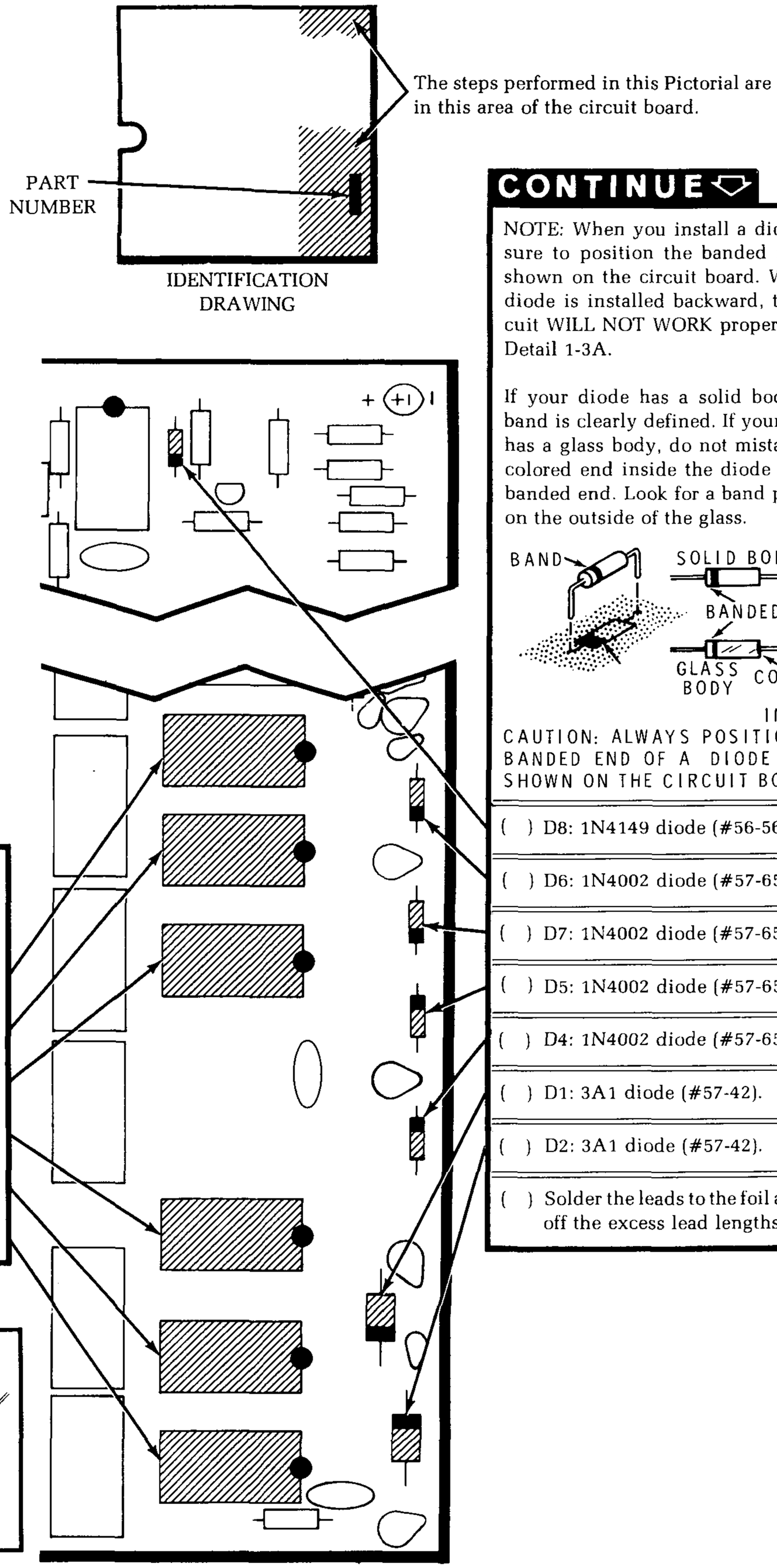
( ) R1: 27 kΩ (red-viol-org).

( ) R3: 68 Ω, **1/2-watt** (blu-gry-blk).

( ) R4: 68 Ω, **1/2-watt** (blu-gry-blk).

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

PICTORIAL 1-2



Detail 1-3A

PICTORIAL 1-3

START

Install 16-pin IC sockets at the following locations. Solder the pins to the foil as you install each socket.

( ) U23. . . . . .

( ) U24. . . . . .

( ) U25. . . . . .

( ) U26. . . . . .

( ) U27. . . . . .

( ) U28. . . . . .

CONTINUE

NOTE: When you install a diode, be sure to position the banded end as shown on the circuit board. When a diode is installed backward, the circuit WILL NOT WORK properly. See Detail 1-3A.

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.

( ) D8: 1N4149 diode (#56-56).

( ) D6: 1N4002 diode (#57-65).

( ) D7: 1N4002 diode (#57-65).

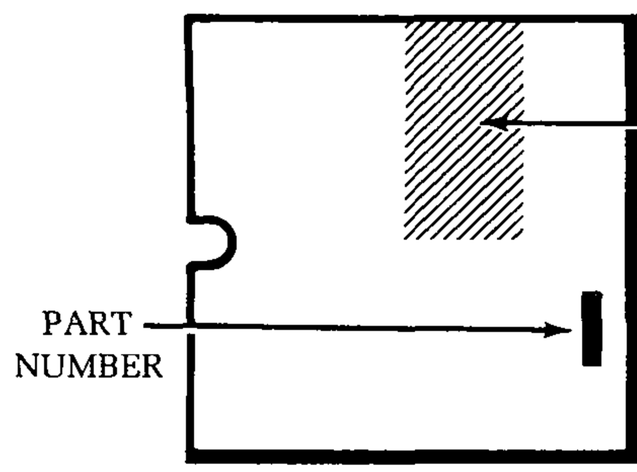
( ) D5: 1N4002 diode (#57-65).

( ) D4: 1N4002 diode (#57-65).

( ) D1: 3A1 diode (#57-42).

( ) D2: 3A1 diode (#57-42).

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



IDENTIFICATION DRAWING

START

( ) R55: 8200 Ω (gry-red-red).

( ) R52: 8200 Ω (gry-red-red).

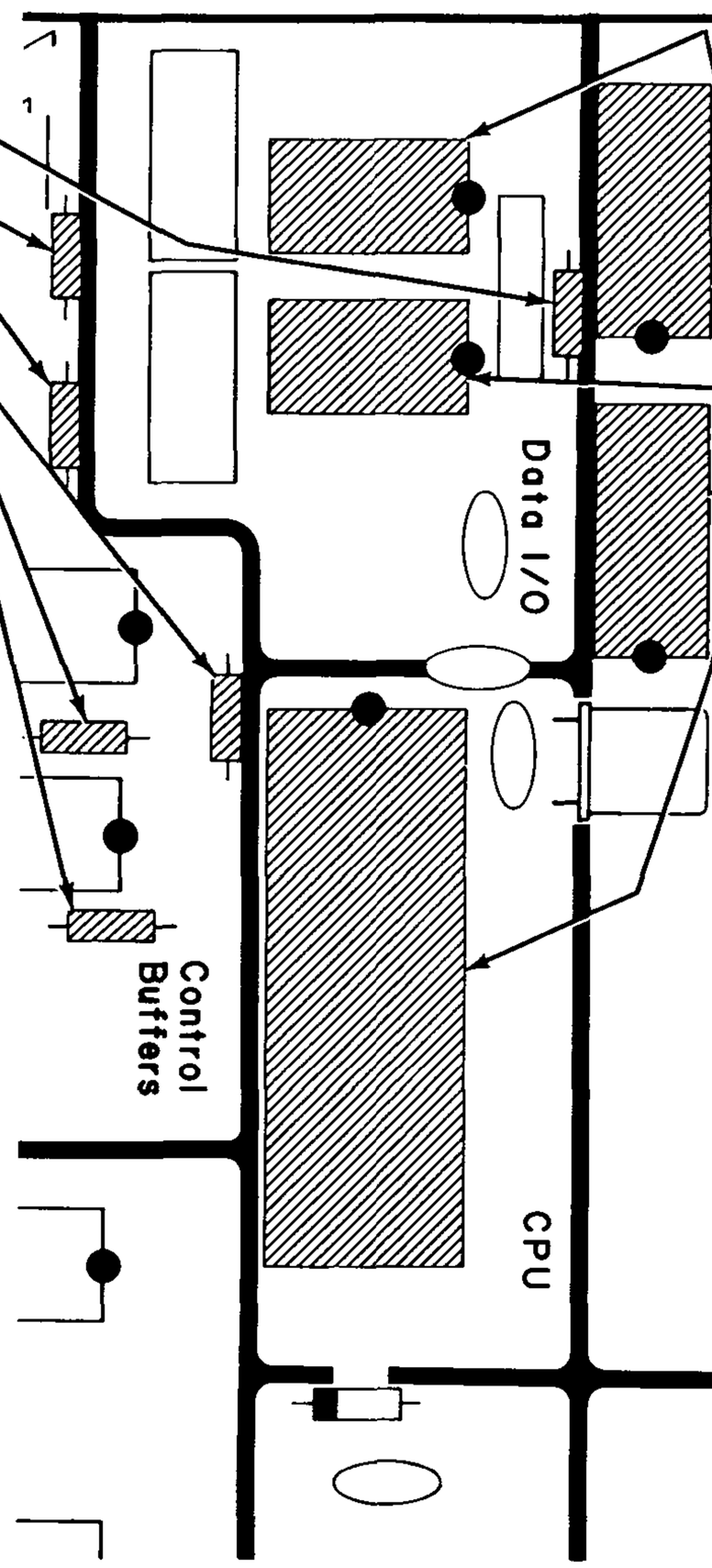
( ) R47: 8200 Ω (gry-red-red).

( ) R54: 3000 Ω (org-blk-red)

( ) R48: 8200 Ω (gry-red-red).

( ) R57: 8200 Ω (gry-red-red).

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



CONTINUE

NOTE: Solder the pins on each of the following IC sockets to the foil as you install it.

( ) 14-pin IC socket at U9.

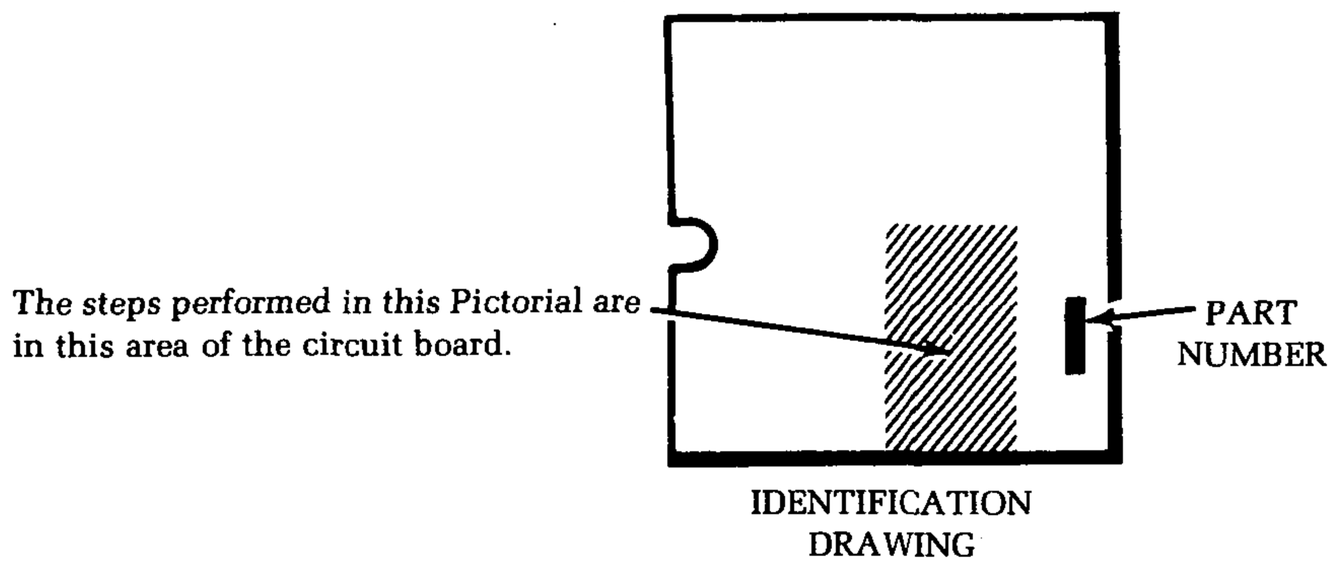
( ) 18-pin IC socket at U14.

( ) 14-pin IC socket at U10.

( ) 18-pin IC socket at U15.

( ) 40-pin IC socket at U11.

PICTORIAL 1-4



START

NOTE: When you install diodes, in some of the following steps, be sure to match the band on the diode with the band mark on the circuit board.

( ) D17: GD510 diode (#56-89).

( ) 24-pin IC socket at U12. Solder the pins to the foil.

( ) D18: 1N4149 diode (#56-56).

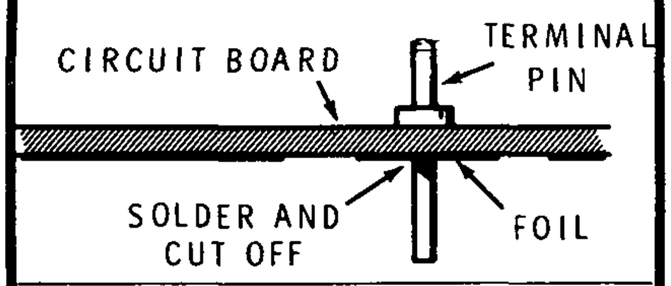
( ) D19: 1N4149 doode (#56-56).

( ) D20: 1N4149 diode (#56-56).

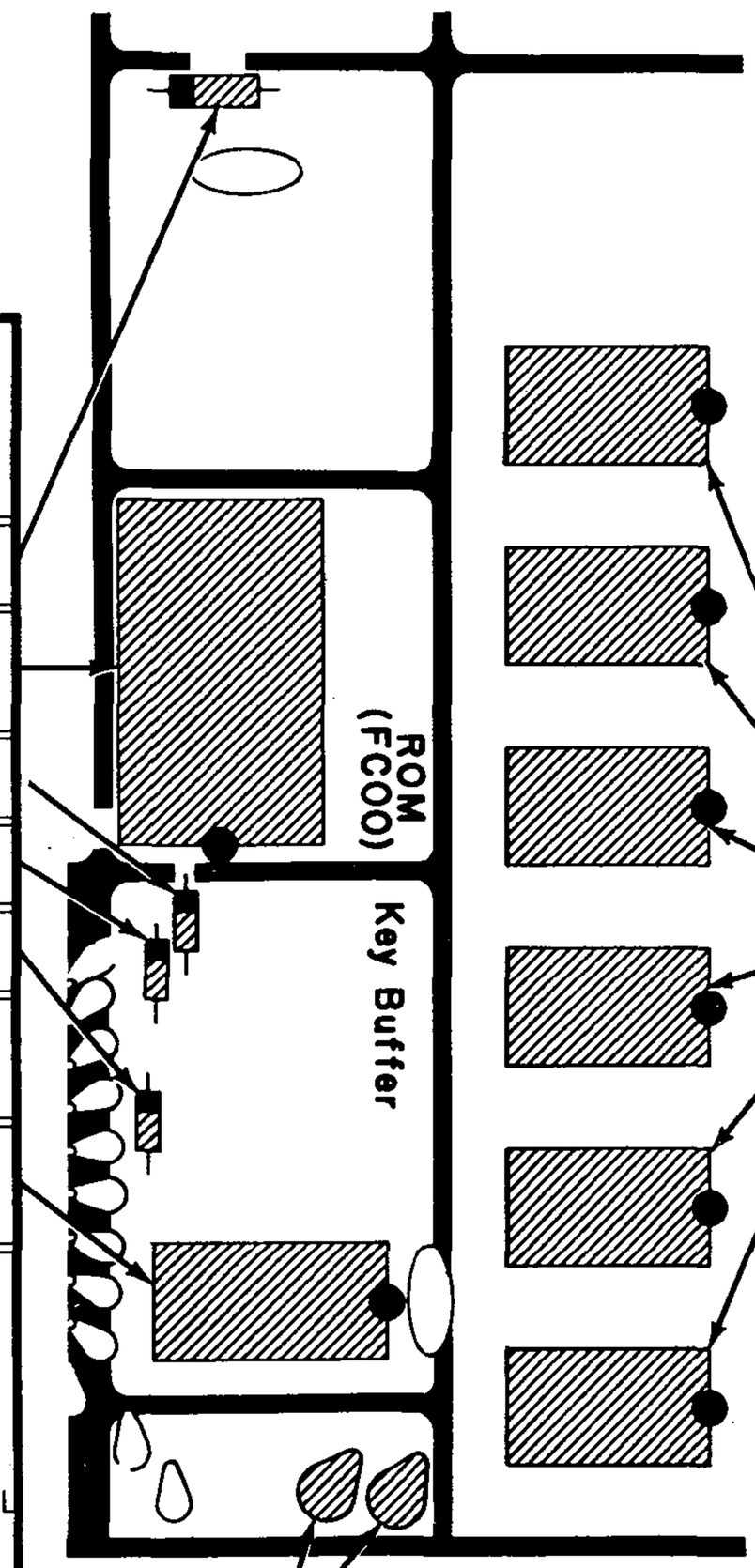
( ) Solder the leads to the foil and cut off the excess lead lengths.

( ) 16-pin IC socket at U13. Solder the pins to the foil.

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.



( ) Two terminal pins at ST1 and ST2.



PICTORIAL 1-5

CONTINUE

Install 14-pin IC sockets at the following locations. Solder the pins of each socket to the foil as you install it.

( ) V1.

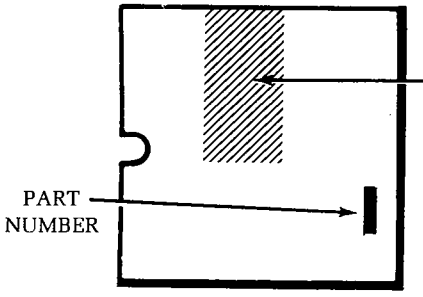
( ) V2.

( ) V3.

( ) V4.

( ) V5.

( ) V6.



IDENTIFICATION DRAWING

START

( ) R45: 8200 Ω (gry-red-red).

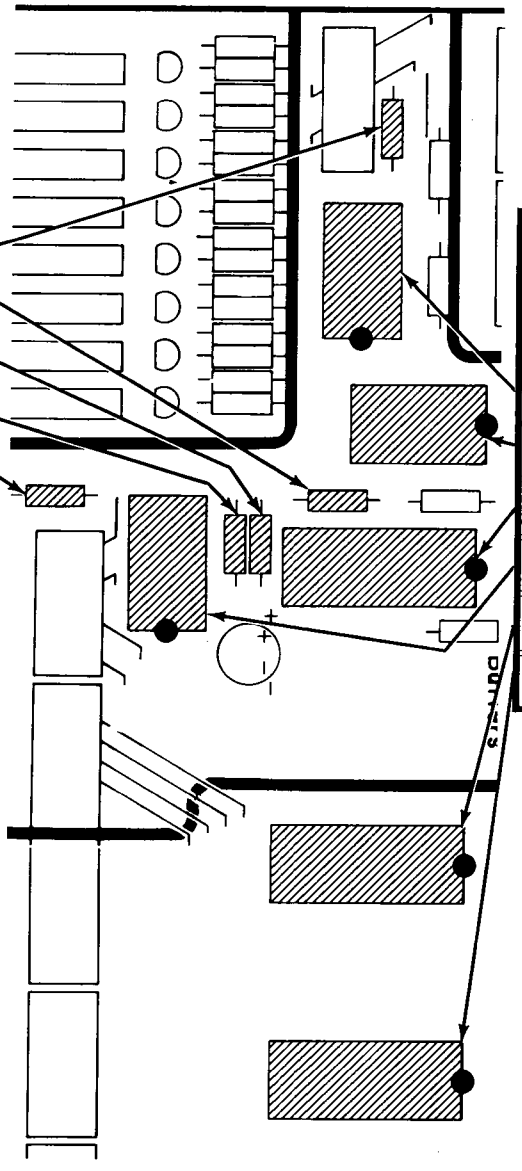
( ) R49: 8200 Ω (gry-red-red).

( ) R51: 8200 Ω (gry-red-red).

( ) R53: 3000 Ω (org-blk-red).

( ) R46: 8200 Ω (gry-red-red).

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



CONTINUE

NOTE: Solder the pins on each of the following IC sockets to the foil as you install it.

( ) 14-pin IC socket at U4.

( ) 14-pin IC socket at U16.

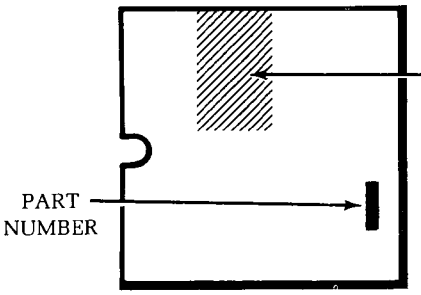
( ) 20-pin IC socket at U6.

( ) 14-pin IC socket at U5.

( ) 20-pin IC socket at U7.

( ) 20-pin IC socket at U8.

PICTORIAL 1-6



IDENTIFICATION DRAWING

START

Install eight 8200 Ω (gry-red-red) resistors at the following locations:

( ) R27 .....

( ) R28 .....

( ) R29 .....

( ) R31 .....

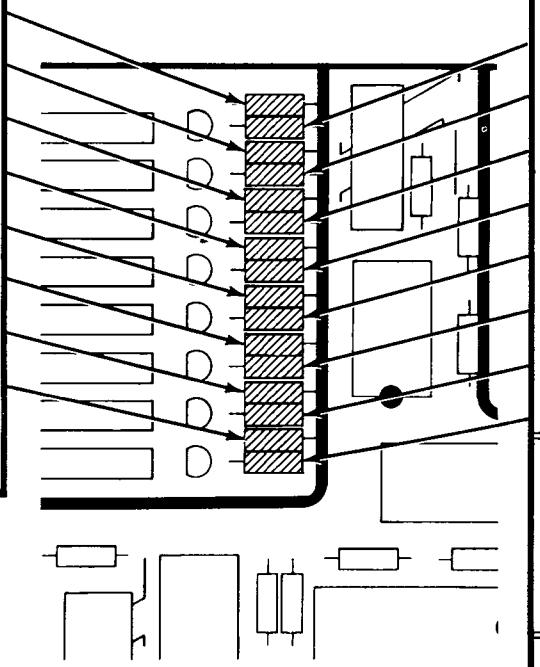
( ) R32 .....

( ) R33 .....

( ) R34 .....

( ) R35 .....

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



CONTINUE

Install eight 150 Ω (brn-grn-brn) resistors at the following locations:

( ) R36.

( ) R37.

( ) R38.

( ) R39.

( ) R41. (Do not cut off the leads of this resistor.)

( ) R42.

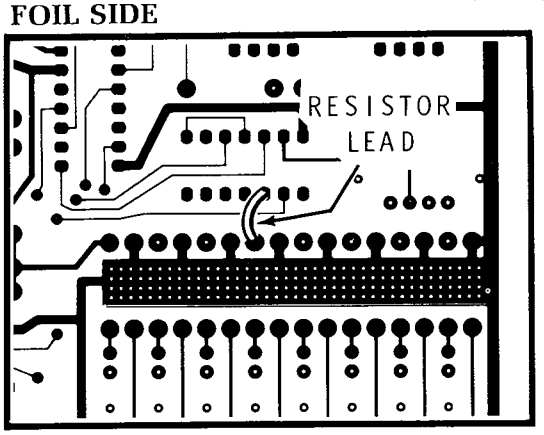
( ) R43.

( ) R44.

( ) Refer to Detail 1-7A and solder the indicated resistor lead to the indicated foil pad of U4. Form a small arch in this lead so it does not touch any foil pattern. Cut off the excess lead length.

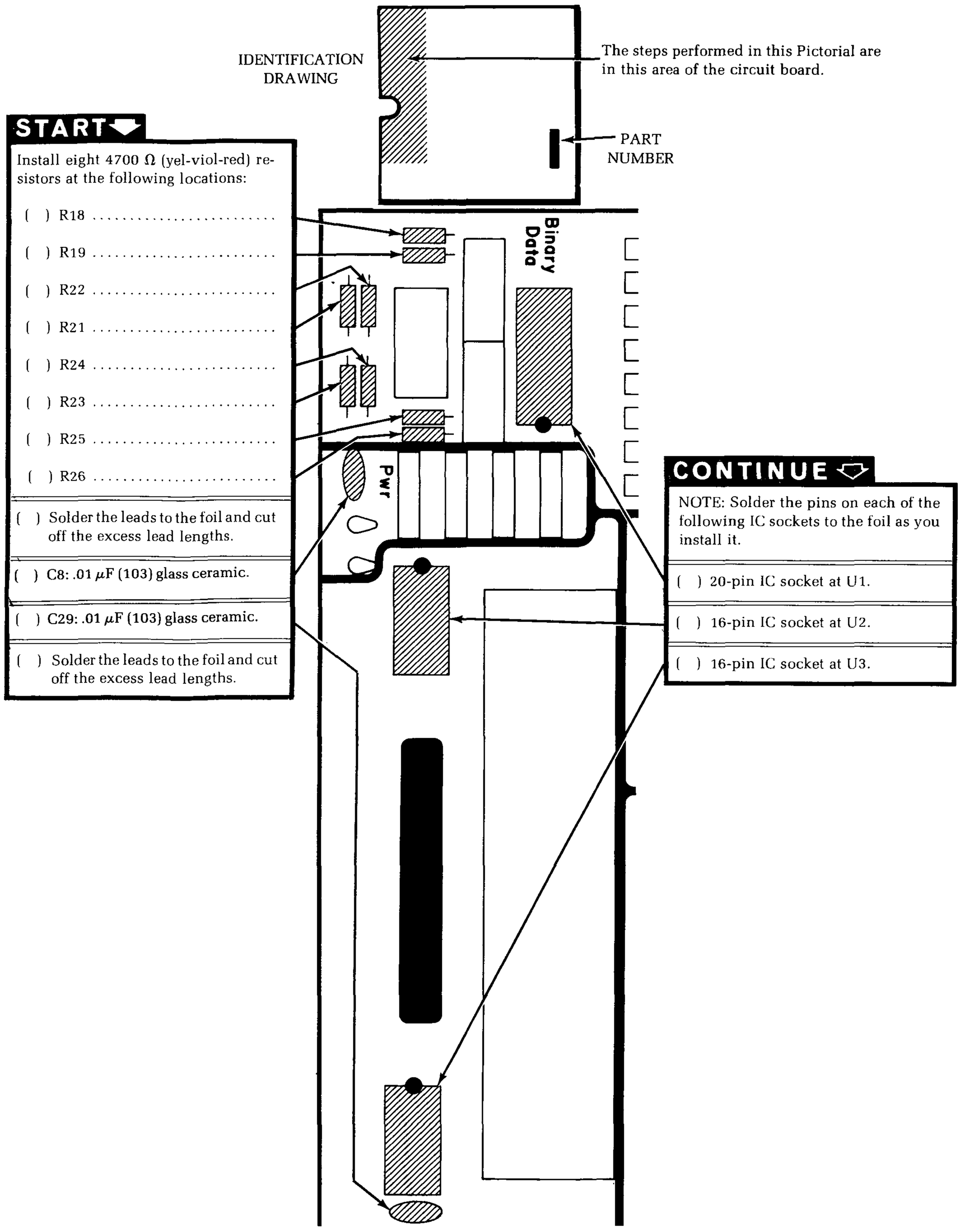
( ) Solder the leads to the foil and cut off the excess lead lengths.

FOIL SIDE

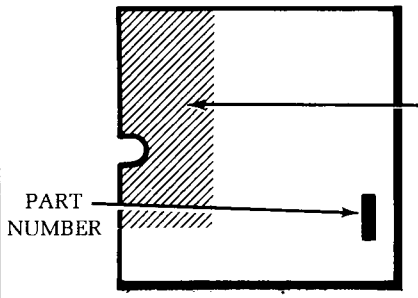


Detail 1-7A

PICTORIAL 1-7



PICTORIAL 1-8

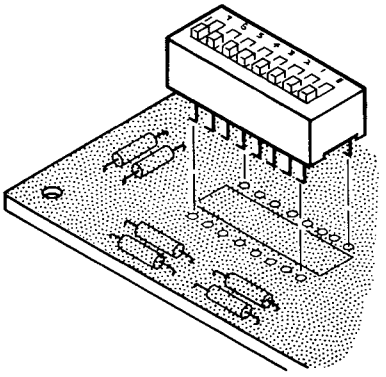


IDENTIFICATION DRAWING

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

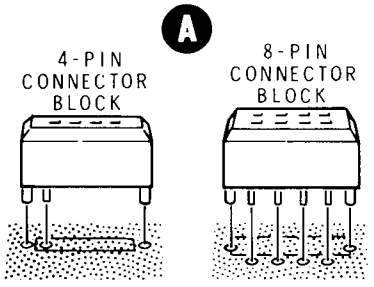
START

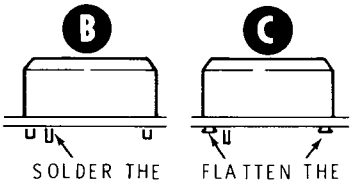
NOTE: In the next step, be sure to position the switch assembly as shown. (It may have slide or rocker switches.)

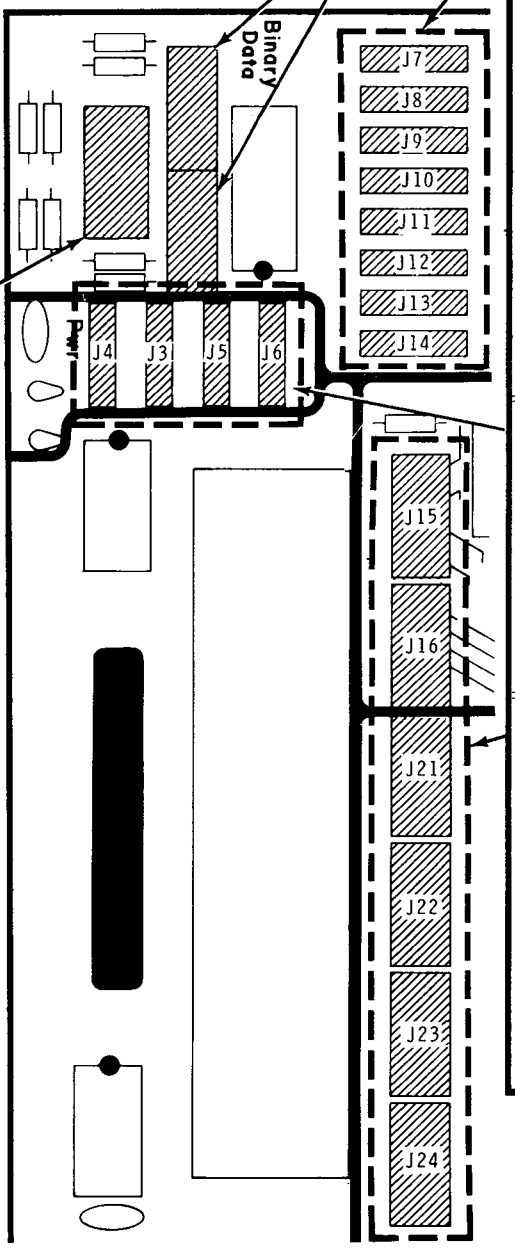


( ) SW2: Switch assembly (#60-621). Solder the pins to the foil.

NOTE: Use the following procedure to install the connector blocks in the following steps.







CONTINUE

( ) 8-pin connector block at J1.

( ) 8-pin connector block at J2.

Install 4-pin connector blocks at the following locations:

( ) J7.

( ) J8.

( ) J9.

( ) J10.

( ) J11.

( ) J12.

( ) J13.

( ) J14.

Install 4-pin connector blocks at the following locations:

( ) J4.

( ) J3.

( ) J5.

( ) J6.

Install 8-pin connector blocks at the following locations:

( ) J15.

( ) J16.

( ) J21.

( ) J22.

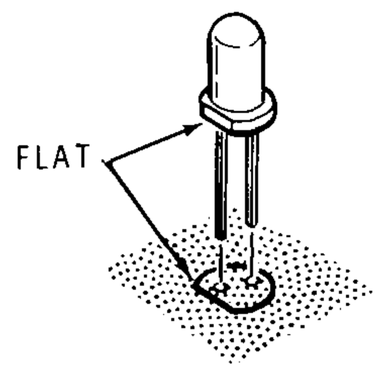
( ) J23.

( ) J24.

PICTORIAL 1-9

START

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



Install red LED's at the following locations:

( ) D9 .....

( ) D10 .....

( ) D11 .....

( ) D12 .....

( ) D13 .....

( ) D14 .....

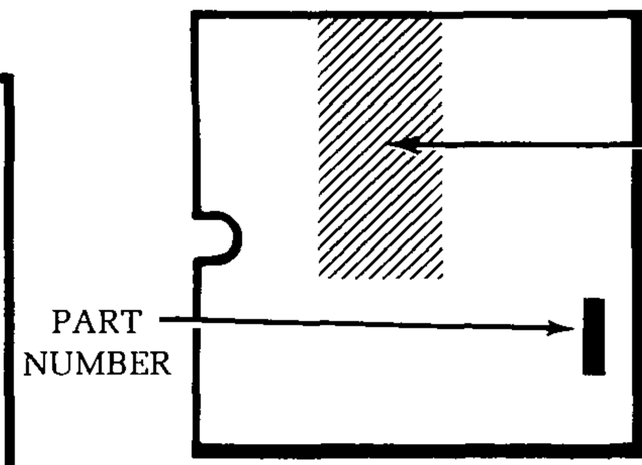
( ) D15 .....

( ) D16 .....

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

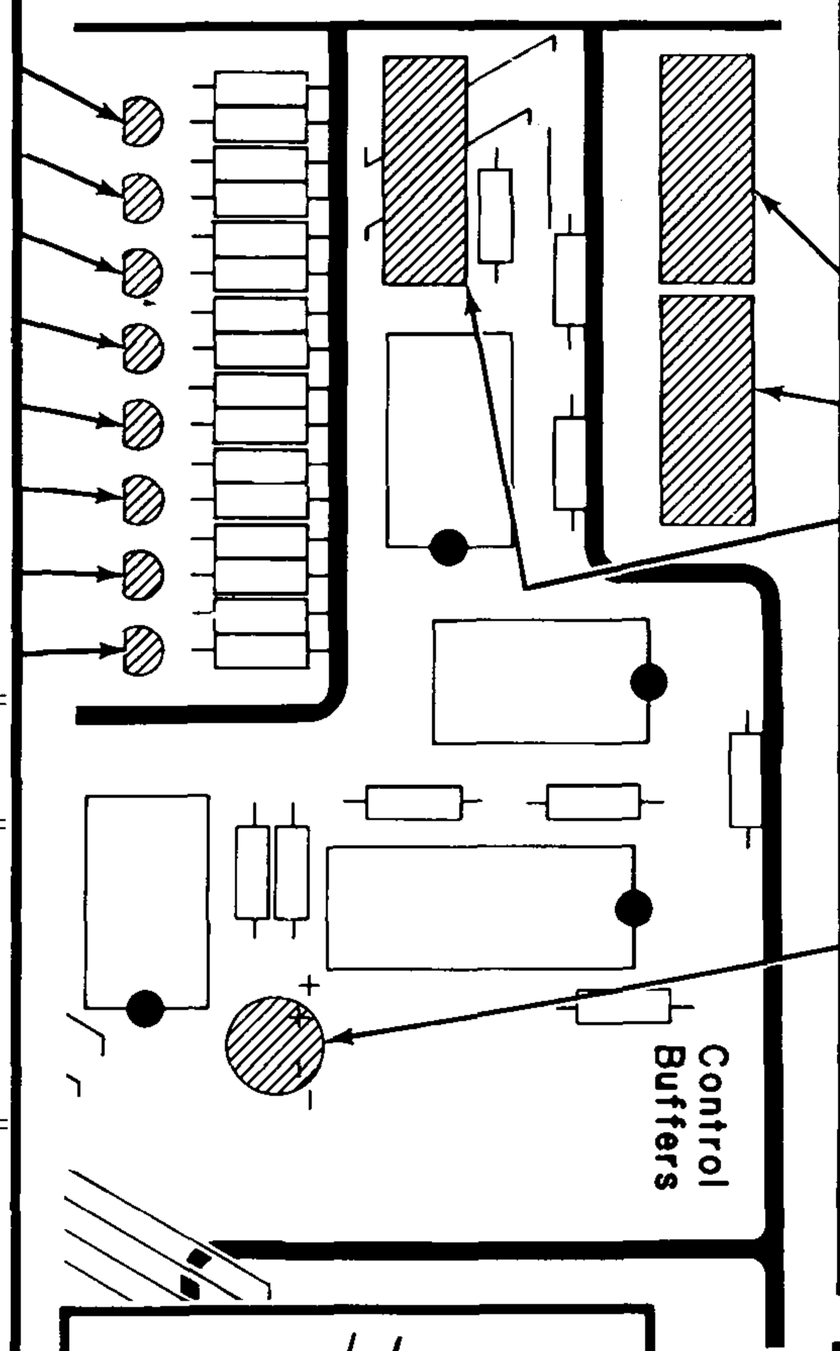
NOTE: When you perform the next step, be sure to remove any burrs on raised edges completely. Be sure the contacts on each block cannot short through to the foil runs on the top of the circuit board.

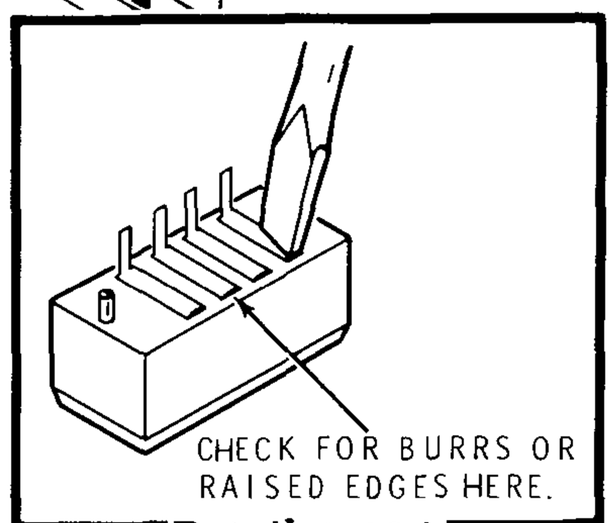
( ) Locate three 8-pin connector blocks. Then refer to Detail 1-10A below and check all four contacts on the bottom of each block. If you find any burrs or raised edges, press them down with a screwdriver blade or similar tool. This will prevent them from causing a short circuit on the circuit board. NOTE: Make sure you use these connector blocks in the next three steps.



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

IDENTIFICATION DRAWING





Detail 1-10A

CONTINUE

( ) Cut three 1/4" × 1/2" lengths of double-stick tape. Use these lengths of tape in the next step.

( ) Carefully peel the thicker backing paper from one side of each piece of tape. Then press the tape onto the bottom of each prepared connector as shown in Detail 1-10B.

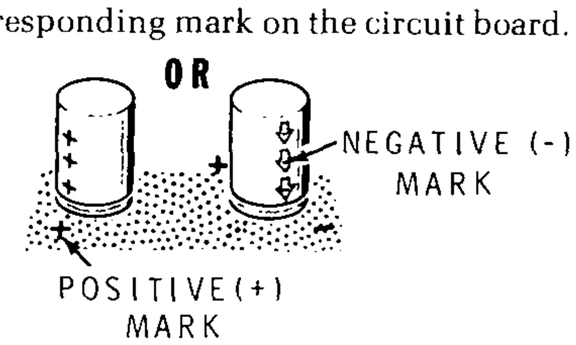
NOTE: Use the same procedure as you did earlier to install the following connector blocks.

( ) Prepared 8-pin connector block at J18.

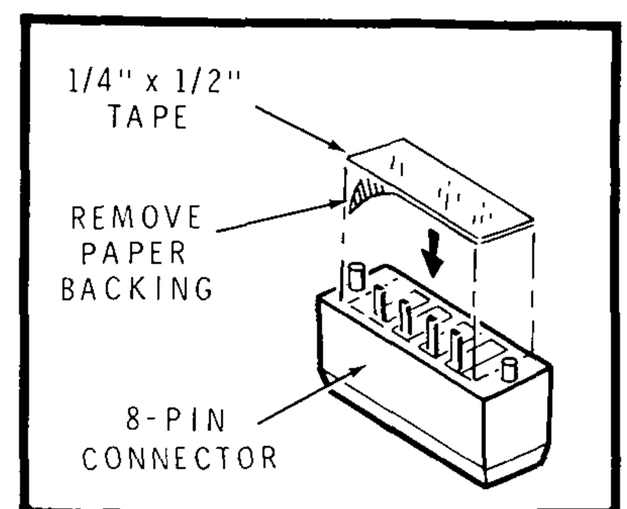
( ) Prepared 8-pin connector block at J19.

( ) Prepared 8-pin connector block at J17.

NOTE: When you install the following electrolytic capacitor, be sure to position the positive (+) or negative (−) mark on the capacitor toward the corresponding mark on the circuit board.

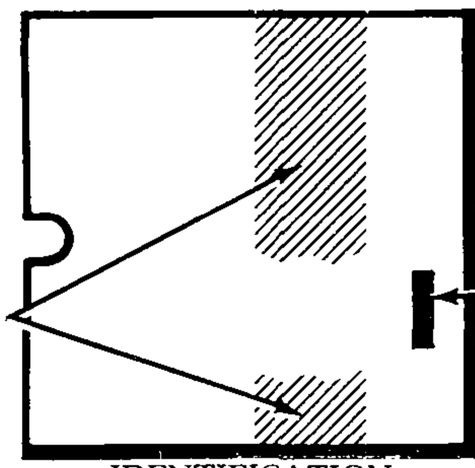


( ) C17: 100 μF electrolytic. Solder the leads to the foil and cut off the excess lead lengths.



Detail 1-10B

PICTORIAL 1-10



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

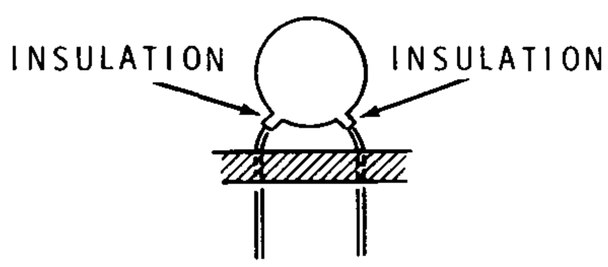
IDENTIFICATION DRAWING

START

( ) 4-pin connector block at J20. Use the same procedure to install this connector block as you did earlier.

( ) C26: .01 μF (103) glass ceramic.

NOTE: When you install the next two ceramic capacitors, do not push the insulated portions of the leads into the circuit board holes. This could make it difficult to solder the leads to the foil.



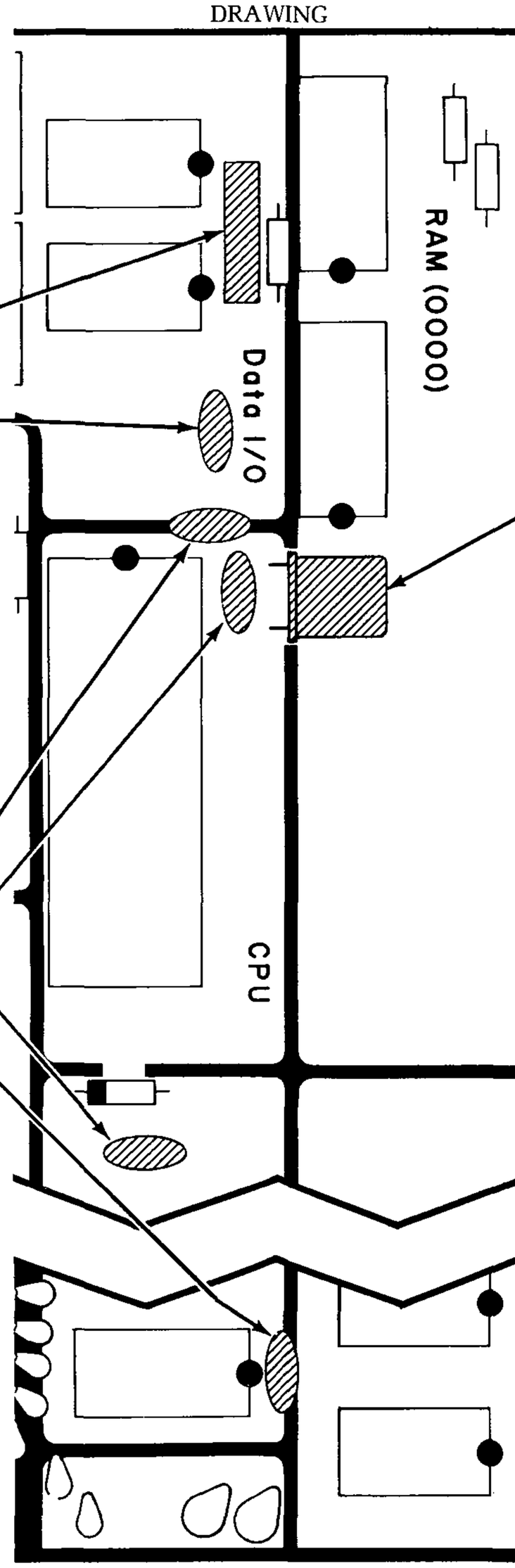
( ) C18: 27 pF ceramic.

( ) C19: 27 pF ceramic.

( ) C27: .01 μF (103) glass ceramic.

( ) C28: .01 μF (103) glass ceramic.

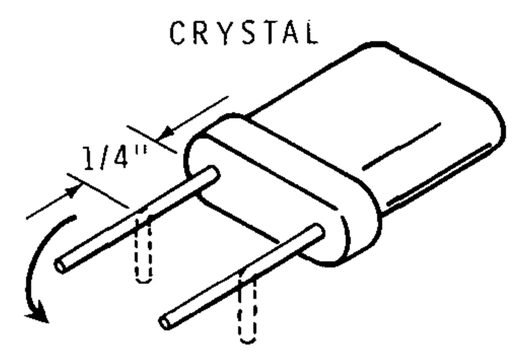
( ) Solder the leads to the foil and cut off the excess lead lengths.



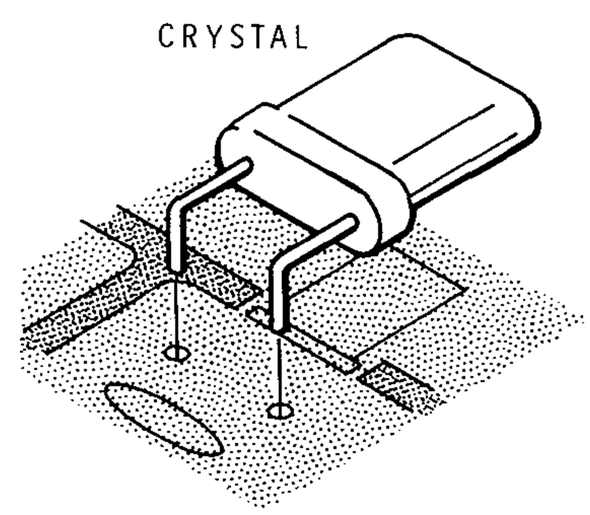
CONTINUE

( ) Y1: Use the following procedure to install a 4 MHz crystal (#404-536) at Y1:

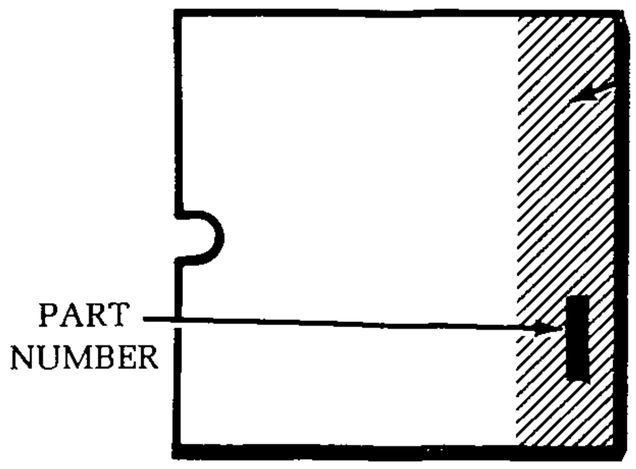
1. Bend each lead at a right angle 1/4" away from the bottom of the crystal as shown.



2. Insert the leads into the circuit board holes and push the crystal down against the circuit board. Then solder the leads to the foil and cut off the excess lead lengths.



PICTORIAL 1-11

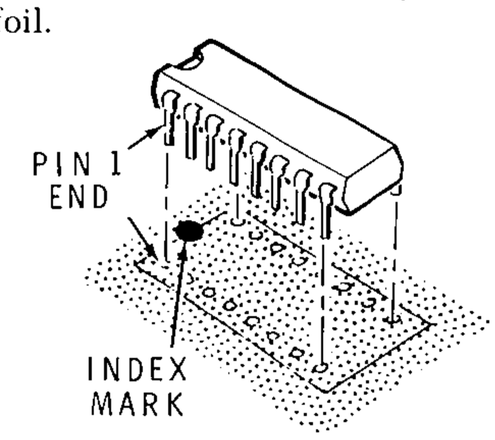


IDENTIFICATION DRAWING

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

START

NOTE: The pin 1 end of the following resistor packs may be marked in any of the ways shown in Detail 1-12A. Be sure to match the pin 1 end of each resistor pack with the index mark shown on the circuit board. Then insert the pins into the circuit board holes, push the pack down against the circuit board, and solder the pins to the foil.



Install four resistor packs (#9-117) at the following locations:

( ) RP1 .....

( ) RP2 .....

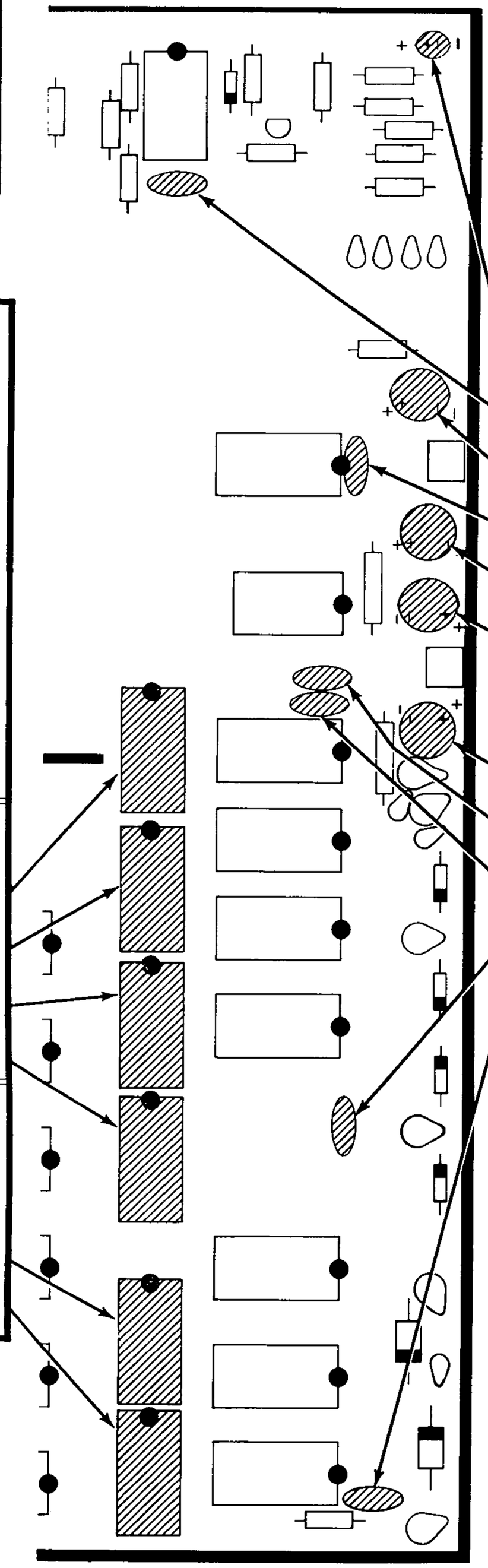
( ) RP3 .....

( ) RP4 .....

( ) Use the same procedure to install two resistor packs (#9-117) at RP5 and RP6. Install both of the packs in the circuit board before you solder the pins to the foil.

( ) RP5 .....

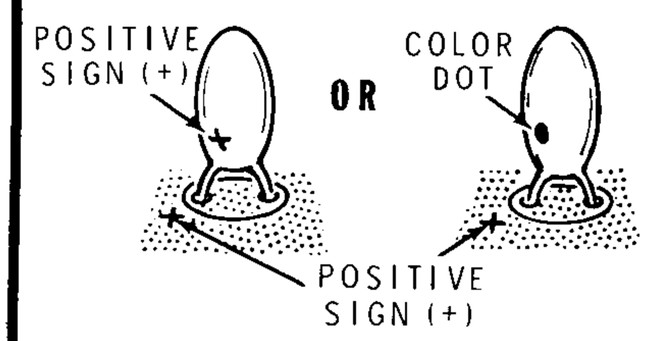
( ) RP6 .....



PICTORIAL 1-12

CONTINUE

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.



( ) C16: .68 μF tantalum.

( ) C25: .01 μF (103) glass ceramic.

( ) C14: 10 μF tantalum.

( ) C21: .01 μF (103) glass ceramic.

( ) C12: 2.2 μF tantalum.

( ) C15: 10 μF tantalum.

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

( ) C13: 2.2 μF tantalum.

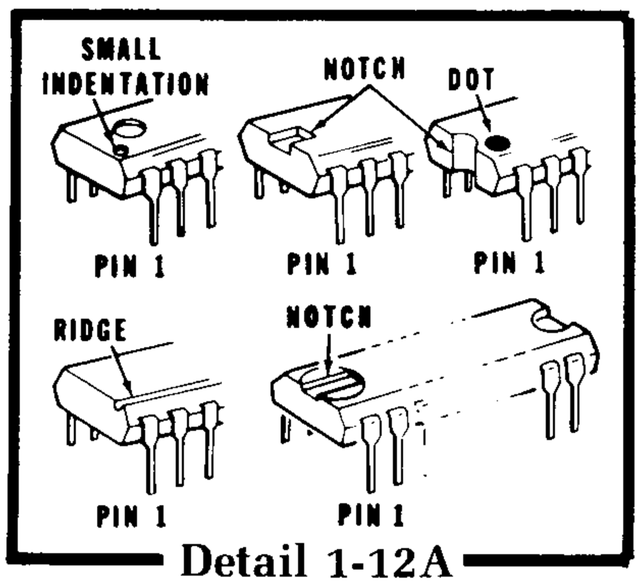
( ) C7: .01 μF (103) glass ceramic.

( ) C22: .01 μF (103) glass ceramic.

( ) C23: .01 μF (103) glass ceramic.

( ) C24: .01 μF (103) glass ceramic.

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



Detail 1-12A

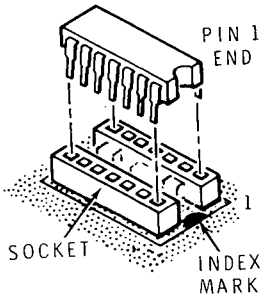
INTEGRATED CIRCUIT INSTALLATION

CAUTION: Integrated Circuits (IC's) are complex electronic devices that perform many complicated functions in the circuit. However, these devices can be damaged during installation. Read ALL of the following information before you install any IC's.

The pins on the IC's are bent out at an angle, so they will not line up with the holes in the IC socket. DO NOT try to install an IC without first bending the pins inward, or the pin or the socket may be damaged, causing intermittent contact.

Before you install an IC, lay it down on its side as shown and very carefully roll it toward the pins to bend the lower pins into line. Then turn the IC over and bend the pins on the other side in the same manner.

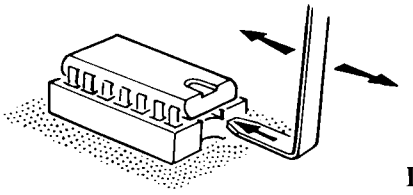
Make sure that the pin one end of the IC (refer to Detail 1-13A) is positioned over the index mark on the circuit board. Also make sure that all of the pins are started into the socket. Then press the IC firmly into the socket. NOTE: An IC pin can become bent under the IC and it will appear as though it is correctly installed in the socket.



Detail 1-13A

Refer to Pictorial 1-13 (Illustration Booklet, Page 3) for the following steps.

NOTE: An IC lifter has been furnished so you can remove a dual in-line IC from its socket if it ever becomes necessary. Push the shorter end of the lifter in between the IC and the socket, as shown in Detail 1-13B, and rock the longer portion back and forth. Be very careful, as the IC pins bend easily.



Detail 1-13B

Install IC's in the sockets on the main circuit board as follows. NOTE: Some of the IC's will be installed later.

( ) U18: LM3302N, LM2901N, or μA775 IC (#442-616).

( ) U20: 74LS42 IC (#443-807).

( ) U21: 74S00 IC (#443-26).

( ) U22: 74LS42 IC (#443-807).

( ) U23: 74LS259 IC (#443-804).

( ) U24: 74LS259 IC (#443-804).

( ) U25: 74LS259 IC (#443-804).

( ) U26: 74LS259 IC (#443-804).

( ) U27: 74LS259 IC (#443-804).

( ) U28: 74LS259 IC (#443-804).

( ) U9: 74LS243 IC (#443-839).

( ) U10: 74LS243 IC (#443-839).

( ) U4: 74126 IC (#443-717).

( ) U16: 74S04 IC (#443-897).

( ) U6: 74LS241 IC (#443-824).

( ) U7: 74LS241 IC (#443-824).

( ) U8: 74LS241 IC (#443-824).

( ) U5: 7408 IC (#443-45).

( ) U1: 74LS240 IC (#443-754).

NOTE: Be sure to install the next two IC's as shown on the circuit board; the notches are not in the same direction as the other IC's.

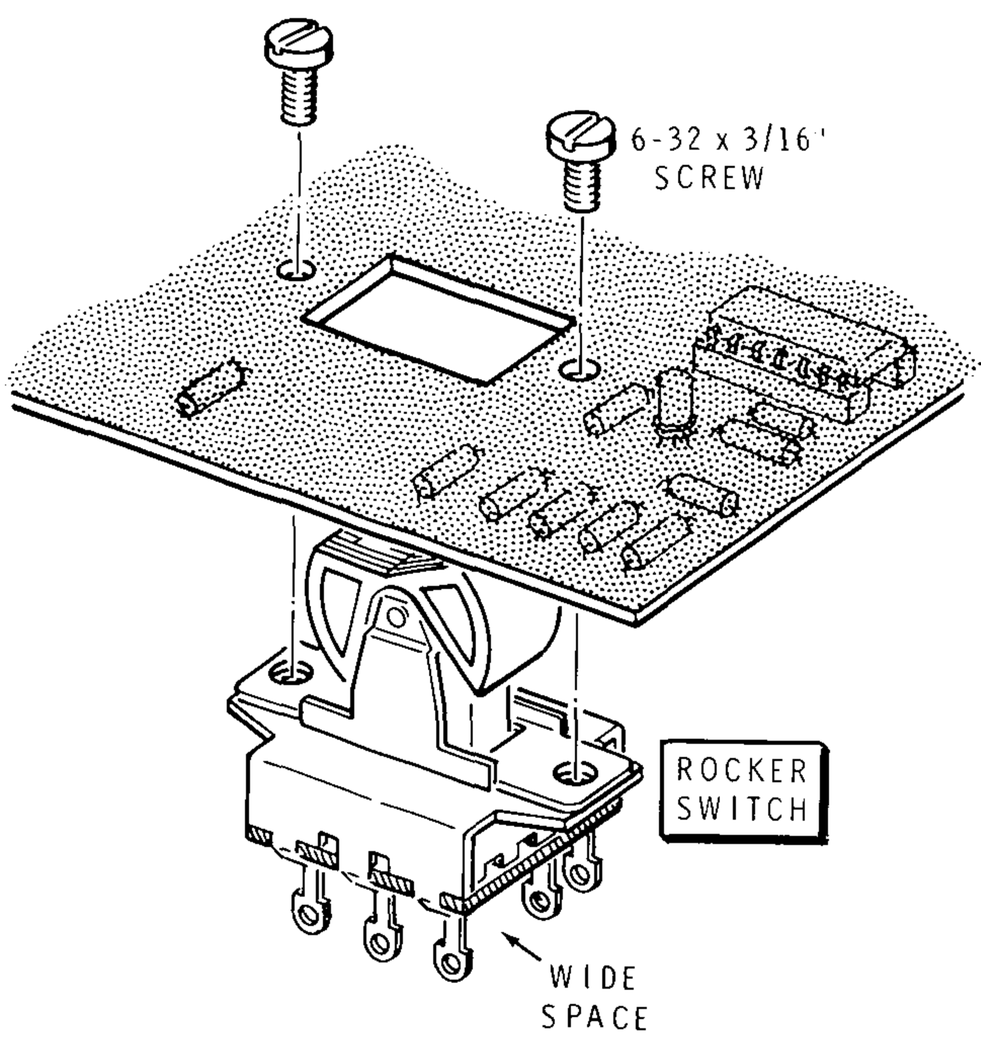
( ) U2: 74LS42 IC (#443-807).

( ) U3: 74LS42 IC (#443-807).

Refer to Pictorial 1-14 (Illustration Booklet, Page 3) for the following steps.

( ) Position the main circuit board as shown in the Pictorial.

( ) SW1: Refer to Detail 1-14A and mount the rocker switch onto the main circuit board at SW1 with two 6-32 × 3/16" screws. Be sure to mount this switch so the lugs are positioned as shown in the Detail.



Detail 1-14A

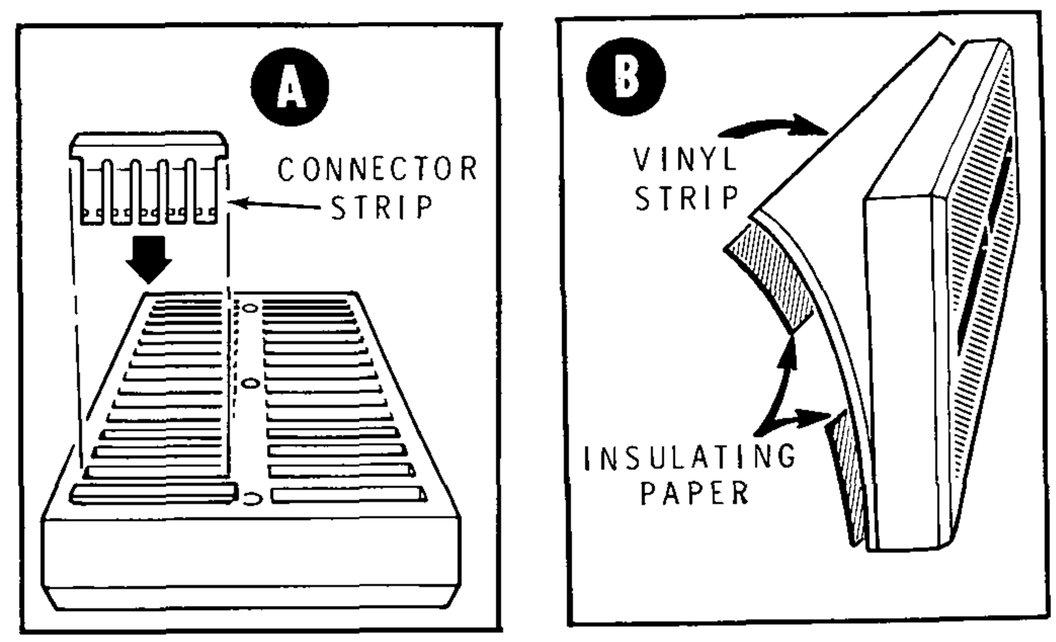
( ) Refer to Detail 1-14B Part A and push the connector strips (supplied with the large connector block) into the block as shown. NOTE: You may have some connector strips left over.

( ) Turn the connector block right side up. Then use a screwdriver or similar tool to tap on the top of the block until all of the connector strips are fully seated in the block.

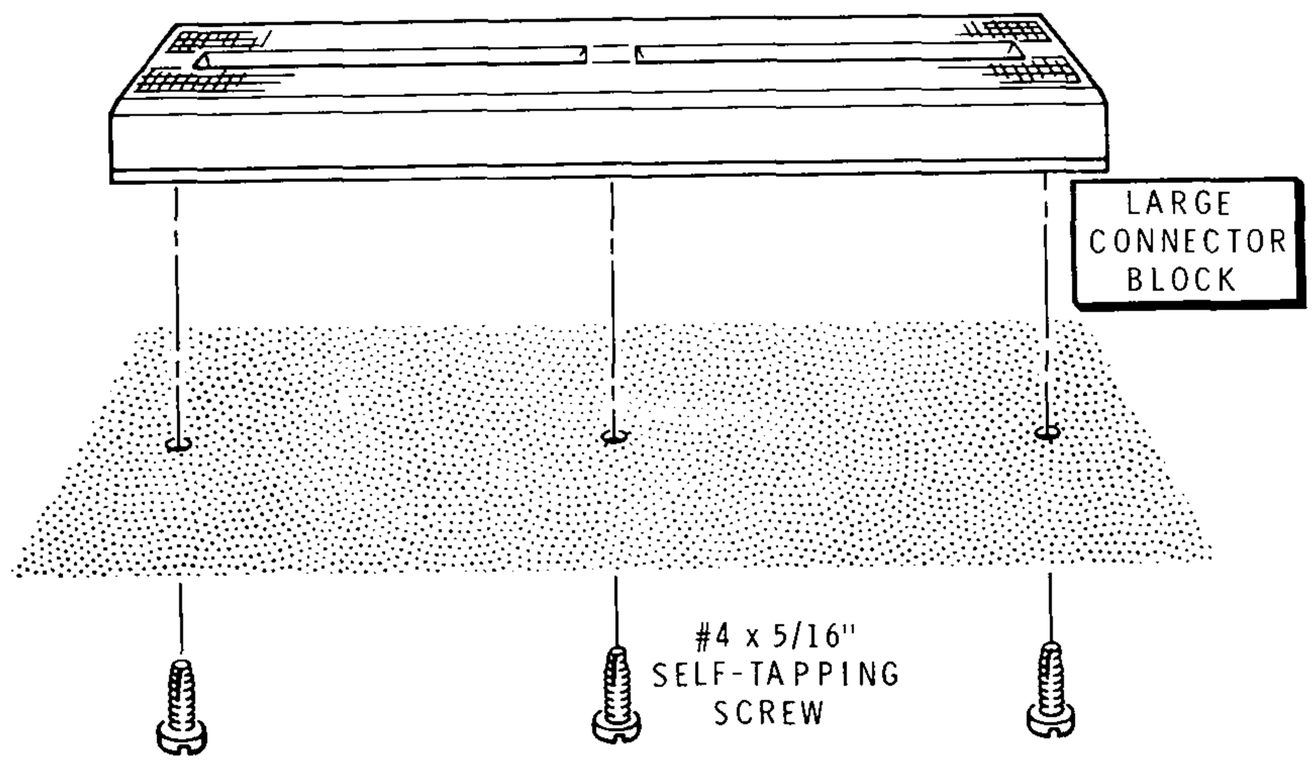
( ) Carefully peel the backing paper from the vinyl strip supplied with the connector block. Then position the connector block as shown in Detail 1-14B Part B, line up the long edge of the vinyl strip with the long edge of the connector block, and firmly press the strip onto the block.

( ) Carefully peel the backing paper from each piece of insulating paper. Then refer again to Detail 1-14B Part B and press the insulating papers along the indicated edges of the vinyl strip. Keep the paper even with the edges of the large connector block.

( ) Use the tip of a pencil to push through the three mounting hole locations in the vinyl strip.



Detail 1-14B



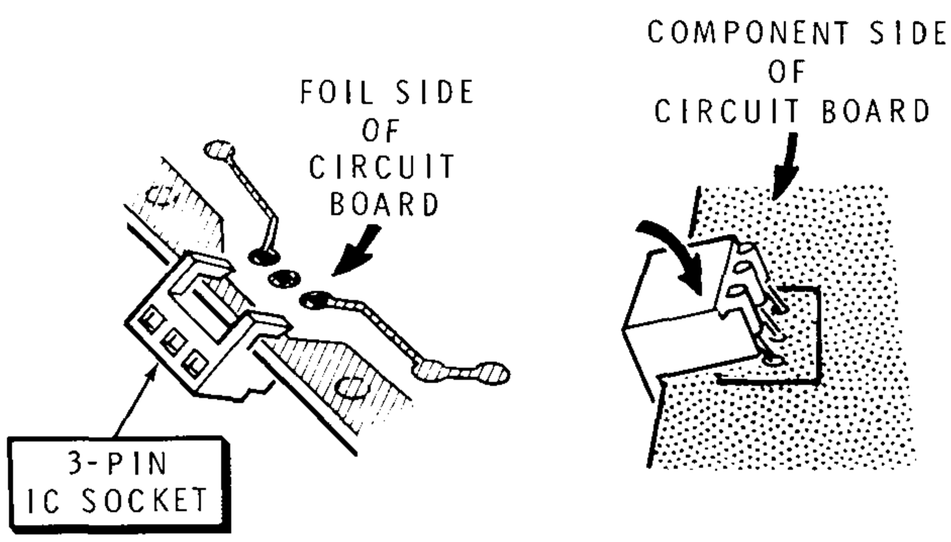
Detail 1-14C

( ) Refer to Detail 1-14C and mount the large connector block onto the main circuit board with three #4 × 5/16″ self-tapping screws.

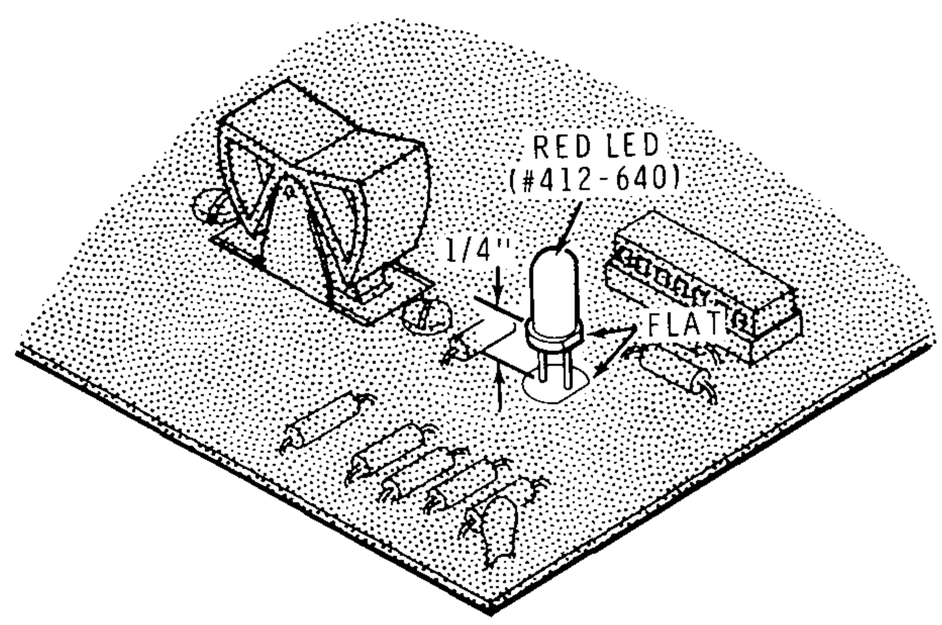
( ) Refer to Detail 1-14D and mount a 3-pin IC socket onto the top edge of the main circuit board at U29 as shown. First place the edge-mount retainers over the edge of the circuit board. Then rotate the pins into their holes on the component side of the circuit board and carefully solder the three pins to the foil.

( ) Similarly, mount a 3-pin IC socket onto the main circuit board at U30.

( ) D3: Refer to Detail 1-14E and mount a red LED (#412-640) onto the main circuit board near the rocker switch as shown. Be sure to match the flat on the LED with the outline of the flat on the circuit board. NOTE: Be sure the bottom of the LED is 1/4″ above the top of the circuit board and is not tilted. Then solder the leads to the foil and cut off any excess lead lengths.



Detail 1-14D



Detail 1-14E

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 Views" (Illustration Booklet, Page 14).

( ) Protruding leads which could touch together.

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

( ) Integrated circuits for the proper type and installation. NOTE: Some of the IC's will be installed later.

( ) Electrolytic and tantalum capacitors for the correct position of the positive (+) and negative (−) leads.

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

( ) LED's for the correct position of the flats.

NOTE: There are many unused connections on the foil side of the main circuit board, some of which you will use later. As you make further connections to the circuit board, be sure to inspect each one carefully to make sure you do not have any solder bridges.

Set the main circuit board aside temporarily.

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

NOTE: 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.

Install six 8200 Ω (gry-red-red) resistors at the following locations. Solder the leads to the foil and cut off the excess lead lengths.

( ) R101 .....

( ) R104 .....

( ) R102 .....

( ) R105 .....

( ) R103 .....

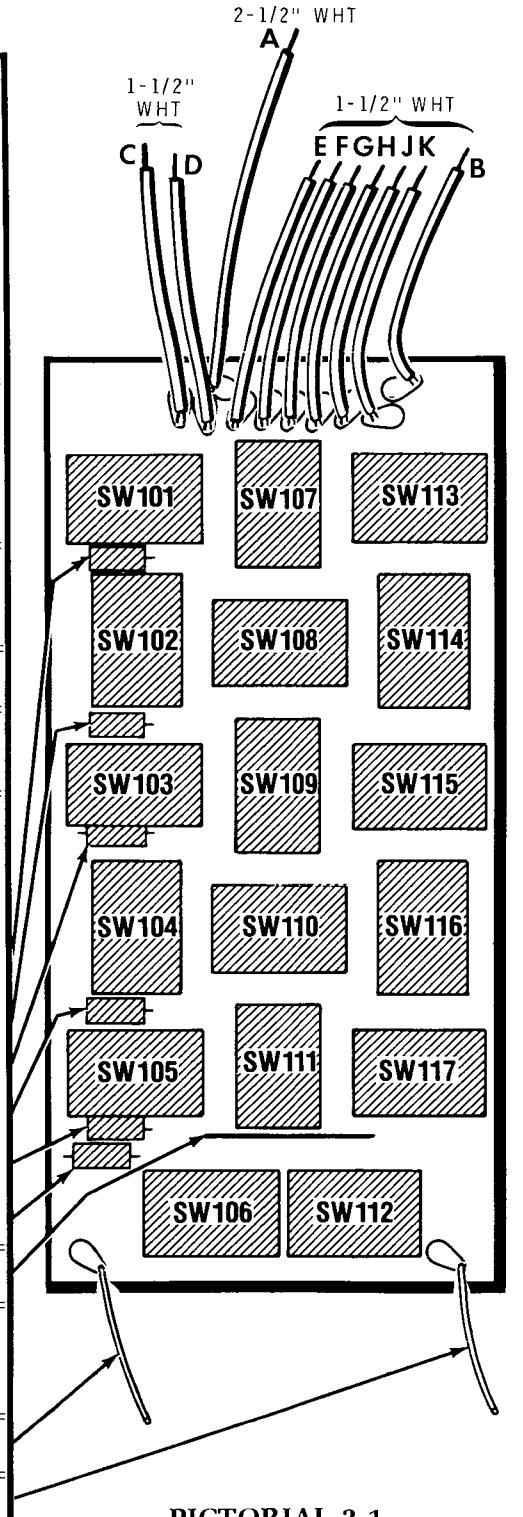
( ) R106 .....

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

( ) Remove the insulation from 2" of the length of brown wire. Then cut off two 1" lengths of bare wire.

( ) 1" bare wire at L.

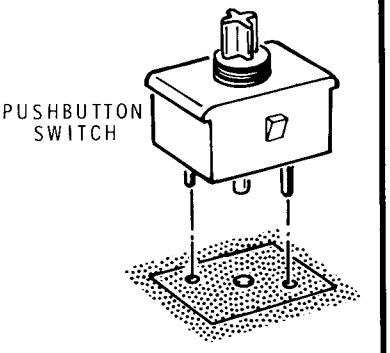
( ) 1" bare wire at M.



PICTORIAL 2-1

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.



( ) SW101 — SW117: Install 17 pushbutton switches at SW101 through SW117.

CIRCUIT BOARD CHECKOUT

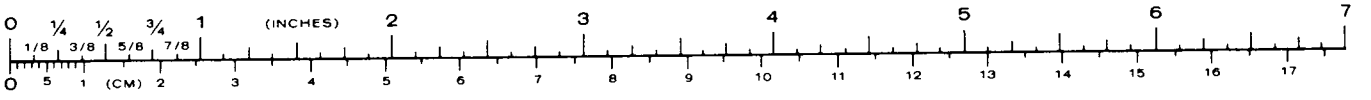
Carefully inspect the circuit board for the following conditions.

( ) Unsoldered connections.

( ) Poor solder connections.

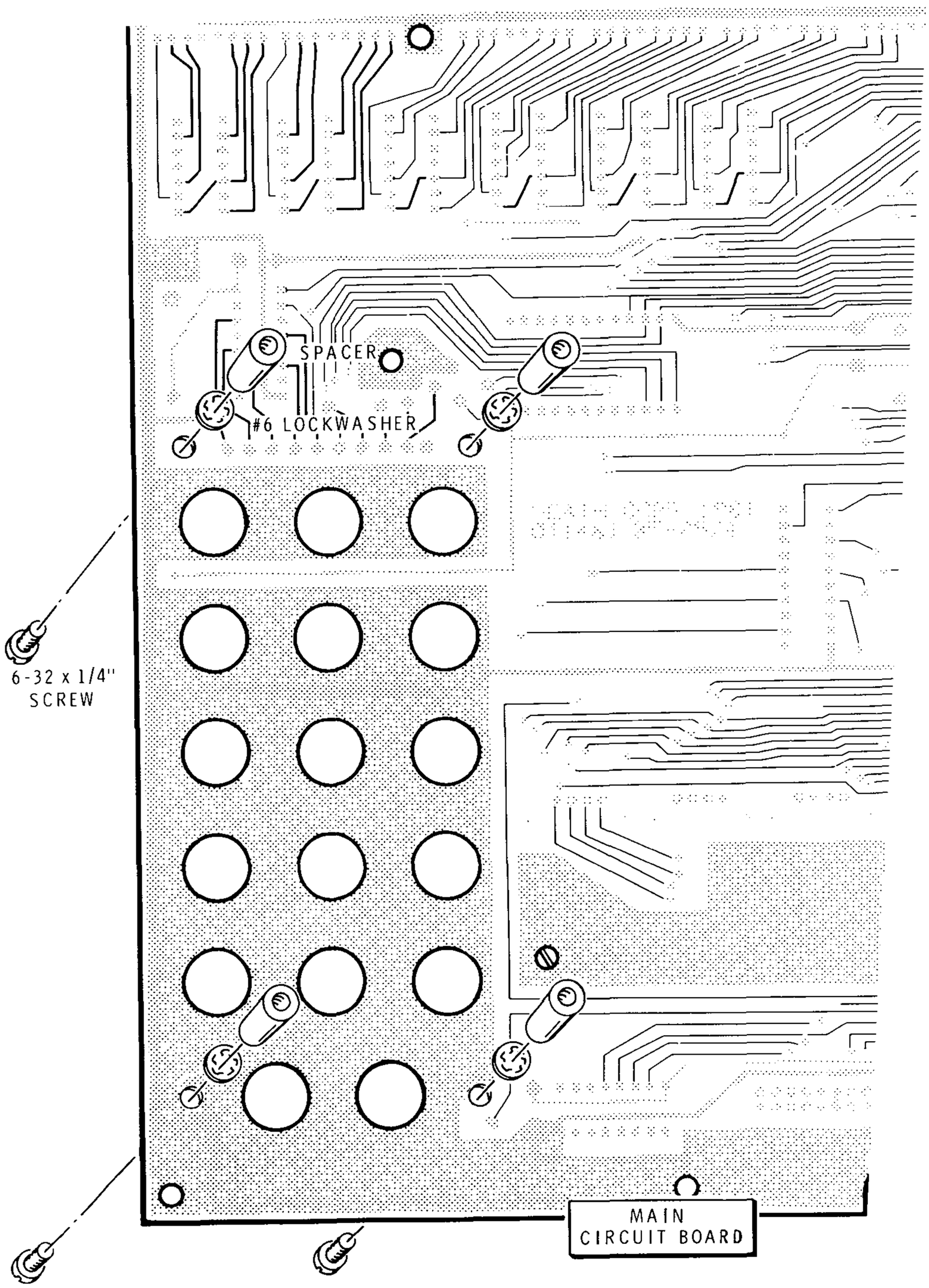
( ) Solder bridges between foil patterns.

( ) Protruding leads which could touch together.

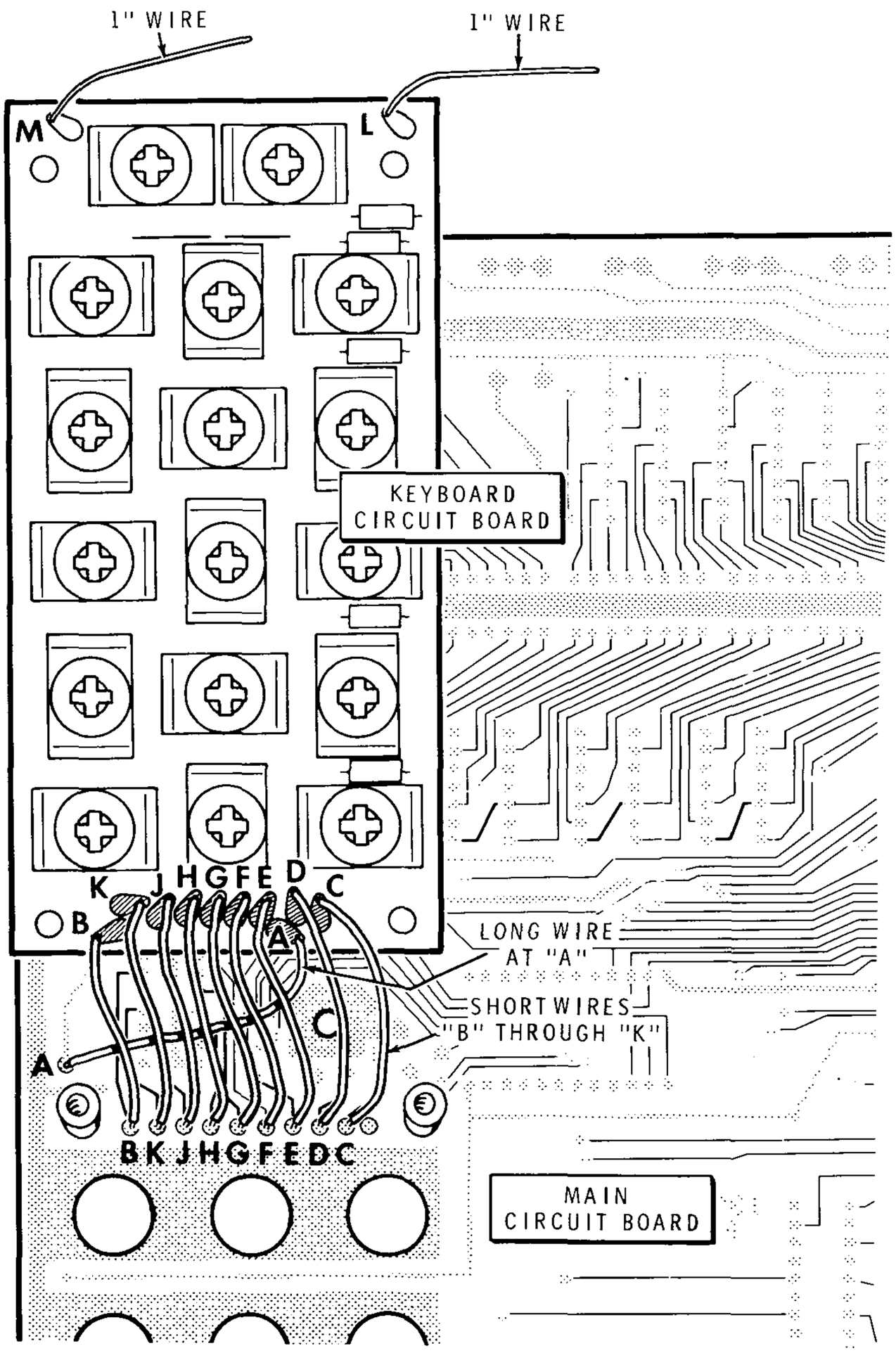


Refer to Pictorial 3-1 (Illustration Booklet, Page 4) for the following steps.

( ) Position the main circuit board foil side up as shown in Detail 3-1A. Then loosely mount spacers onto the foil side at the four locations shown in the Pictorial. Use 6-32 × 1/4" screws and #6 lockwashers.



Detail 3-1A


1" WIRE
1" WIRE
M
L
KEYBOARD CIRCUIT BOARD
K J H G F E D C
B
A
LONG WIRE AT "A"
SHORT WIRES "B" THROUGH "K"
C
A
B K J H G F E D C
MAIN CIRCUIT BOARD

Detail 3-1B

( ) Position the keyboard circuit board foil side down near the main circuit board as shown in Detail 3-1B.

NOTE: As you connect each of the following wires, push the bare end into its designated hole and leave approximately 1/16" of the bare wire above the foil so you can make a good solder connection. Solder each wire to the foil as you connect it and cut off any excess wire end on the component side of the circuit board.

Connect the free ends of the wires coming from the keyboard circuit board to the main circuit board as follows:

( ) Wire from A to hole A.

( ) Wire from B to hole B.

( ) Wire from K to hole K.

( ) Wire from J to hole J.

( ) Wire from H to hole H.

( ) Wire from G to hole G.

( ) Wire from F to hole F.

( ) Wire from E to hole E.

( ) Wire from D to hole D.

( ) Wire from C to hole C.

( ) Lower the keyboard circuit board onto the main circuit board (keep the wires out of the way) so the tops of the pushbutton switches enter 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 free end of the wire coming from keyboard circuit board hole M to hole M on the main circuit board. Do not solder the connection yet.

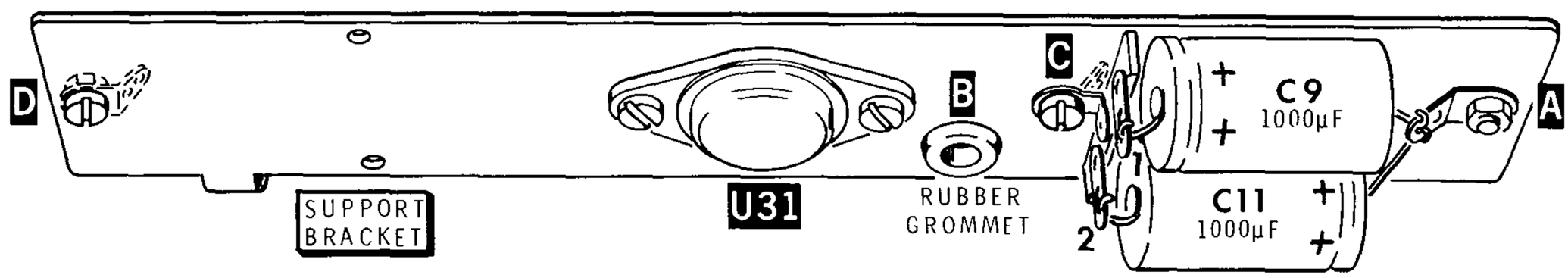
( ) Connect the free end of the wire coming from keyboard circuit board hole L to hole L on the main circuit board. Do not solder the connection yet.

( ) Loosely install four 6-32 × 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 two bare wires to the main circuit board at L and M and cut off any excess wire ends.

Set the circuit board assembly aside temporarily.



PICTORIAL 4-1

SUPPORT BRACKET ASSEMBLY

Refer to Pictorial 4-1 for the following steps.

( ) Position the support bracket on your work surface as shown in the Pictorial.

( ) Refer to Detail 4-1A and mount a #6 solder lug at A with a 6-32 × 1/4″ screw and a 6-32 nut. Position the solder lug as shown in the Pictorial.

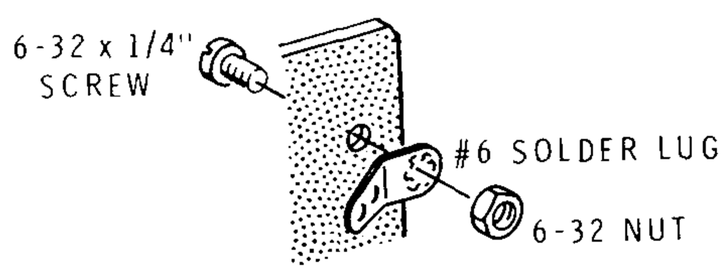
( ) Similarly, mount a #6 solder lug onto the other side of the support bracket at D. Use a 6-32 × 1/4″ screw and a 6-32 nut. Be sure to position the solder lug as shown in the Pictorial.

( ) Press a rubber grommet into hole B.

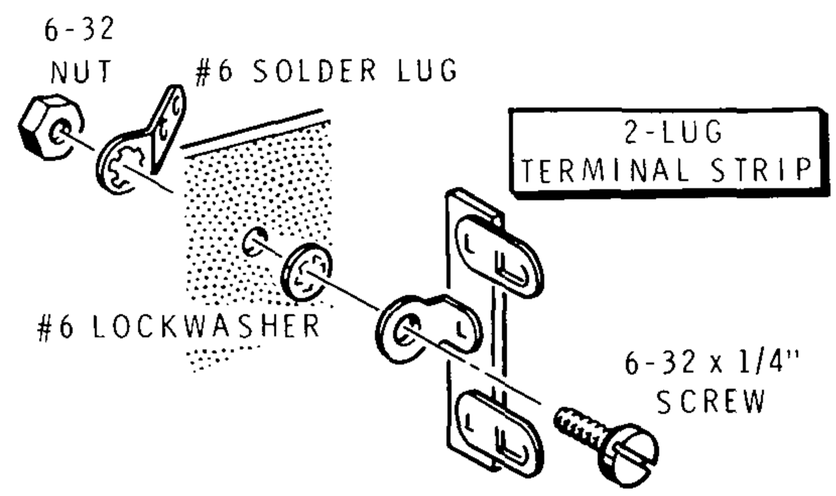
( ) Refer to Detail 4-1B and mount a 2-lug terminal strip at C. Use a 6-32 × 1/4″ screw, a #6 lockwasher, a #6 solder lug, and a 6-32 nut. Position the terminal strip and solder lug as shown in the Pictorial.

NOTE: In the following steps, (NS) means not to solder the connection because you will add other wires later. "S-" with a number, such as (S-2), means to solder the connection. The number following the "S-" tells you how many wires should be at the connection. This helps you check your work for errors as you go.

( ) C9: Cut the lead at the **positive** (+) end of a 1000 µF electrolytic capacitor to 1/2″. NOTE: Be sure you cut the correct lead of the capacitor; only the negative (−) lead may be marked. Then connect the **positive** (+) lead of the 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.



Detail 4-1A



Detail 4-1B

( ) C11: Cut the **negative** (−) lead of a 1000 μF electrolytic capacitor to 1/2". Then connect the negative (−) lead of this capacitor to terminal strip C lug 2 (NS) and the positive (+) lead to solder lug A (S-2).

WARNING: The silicone grease you will use in the following step(s) helps transfer heat away from the integrated circuit. The grease is not caustic, but make sure you do not get it into your eyes, ears, nose, mouth, or clothing. Always wash your hands after you use the grease. Keep this and all chemicals out of the reach of children.

( ) Refer to the inset drawing on Detail 4-1C and open the container of silicone grease. Then apply a thin coating of the grease to the flat side of a μA309K integrated circuit (442-30).

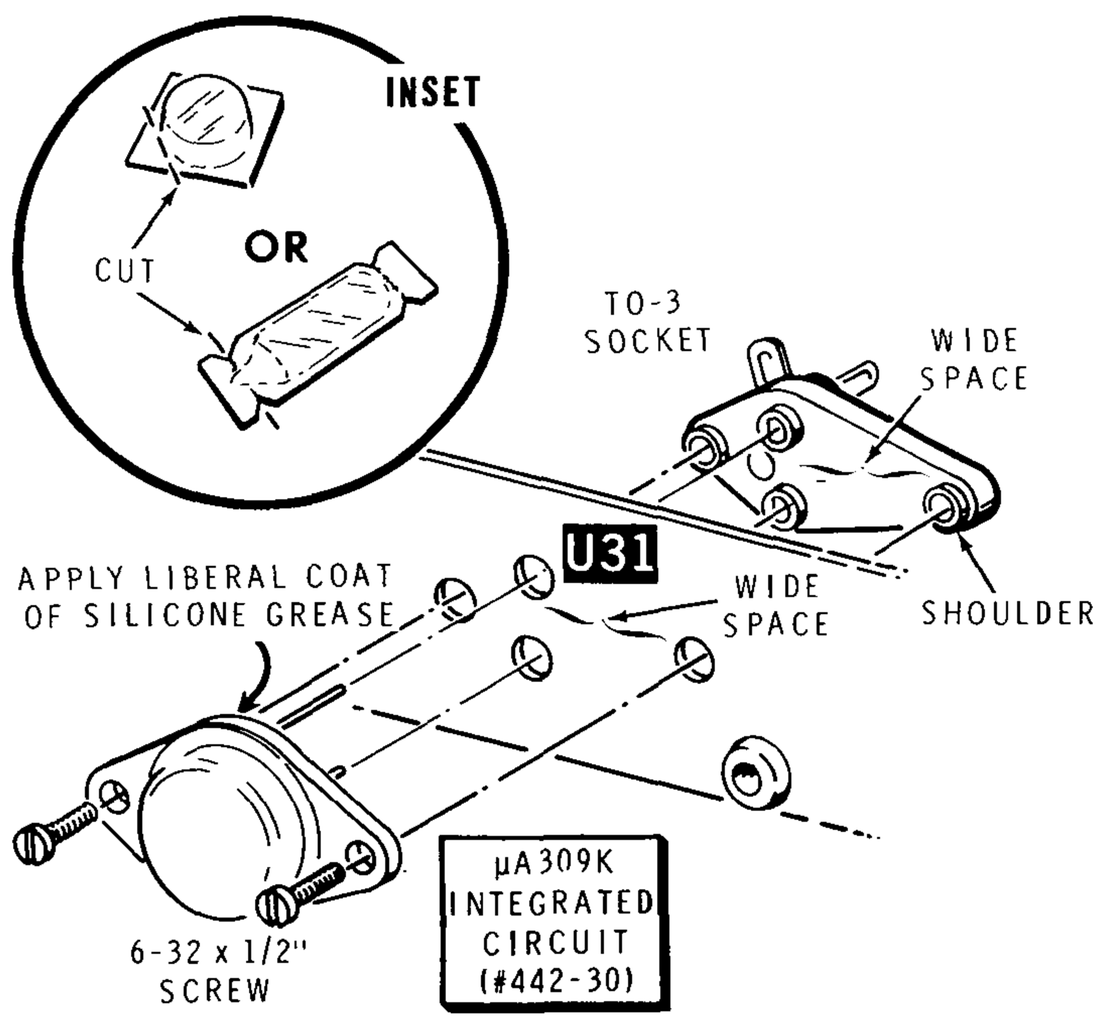
( ) Discard the remaining silicone grease.

( ) U31: Refer again to Detail 4-1C and use the following procedure to mount the prepared integrated circuit onto the support bracket at U31:

1. Carefully observe the wide spacing of the holes in the support bracket at U31 and position the TO-3 socket on the indicated side of the bracket as shown. Be sure the shoulders on the socket enter the holes in the bracket.

2. Pass the pins of the μA309K IC through the holes in the bracket and push them into the socket.

3. Make sure the shoulders on the socket are still seated in the holes in the bracket. Then secure the IC with two 6-32 × 1/2" screws.



Detail 4-1C

Refer to Pictorial 4-2 (Illustration Booklet, Page 4) for the following steps.

( ) Cut a 1″ length of Teflon sleeving. Use this sleeving in the next step.

( ) C4: Cut the negative (−) lead of a 6000 μF electrolytic capacitor to 1″. Then connect the negative lead to solder lug D (S-1). Slide the prepared length of Teflon sleeving onto the positive (+) lead of the capacitor and connect the lead to socket U31 lug 1 (NS).

( ) Prepare a 3-1/2″ black wire. Then connect the wire from solder lug C (S-1) to socket U31 lug 3 (NS).

( ) C5: Cut one lead of a .22 μF Mylar capacitor to 5/8″. Then connect this lead to socket U31 lug 3 (NS), and connect the other lead to lug 1 (NS).

( ) C6: Cut one lead of a .22 μF Mylar capacitor to 5/8″. Then connect this lead to socket U31 lug 3 (S-3), and connect the other lead to lug 2 (NS).

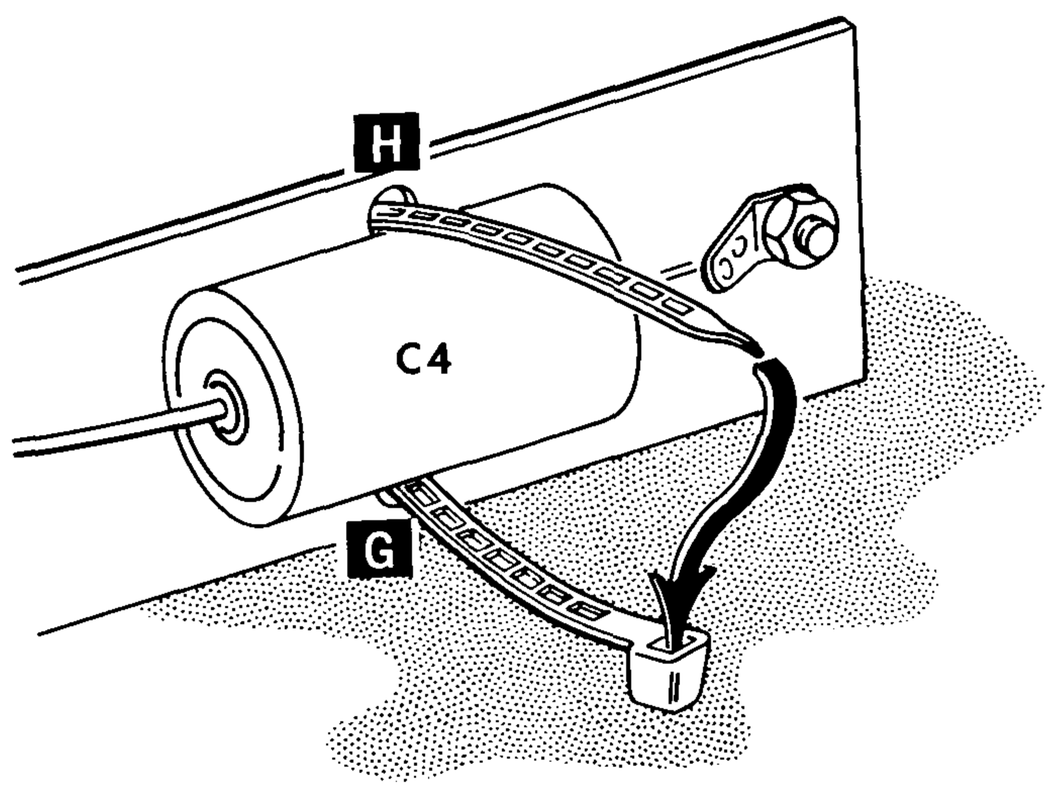
( ) Prepare a 7-1/4″ orange wire and a 9″ white-orange wire. Use these wires in the following two steps.

( ) Connect one end of the prepared orange wire to socket U31 lug 2 (S-2). The other end of this wire will be connected later.

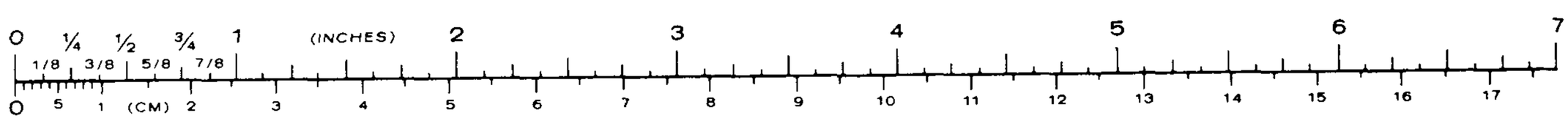
( ) Connect one end of the prepared white-orange wire to socket U31 lug 1 (S-3). The other end of this wire will be connected later.

( ) Mount the support bracket to the foil side of the main circuit board at E and F with 6-32 × 1/4″ screws, #6 lockwashers, and 6-32 nuts.

( ) Refer to Detail 4-2A and pass the tip of a cable tie through hole G in the support bracket. Make sure you position the cable tie as shown. Then pass the tie across the rear of the bracket and back through hole H. Pass the cable tie around capacitor C4 and push the tip through the retainer on the other end of the cable tie as shown. Then pull the cable tie tight and cut off the excess tie.



Detail 4-2A



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

( ) Reposition the main circuit board as shown in the Pictorial.

( ) 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: When you install each wire in the following steps, form it as it is shown in the Pictorial. After you solder a wire to the foil or to a switch lug, be sure to 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 onto the free end of this wire; then connect the free end to main circuit board hole R (S-1).

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

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

( ) Connect a 1-3/4" orange wire from circuit board 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 U31 downward and across the circuit board as shown. Then connect the free end of the wire to switch SW1 lug 2 (S-2).

( ) Route the free end of the white-orange wire coming from socket U31 downward and across the circuit board as shown. Then connect the free end of the wire to circuit board hole X (S-1).

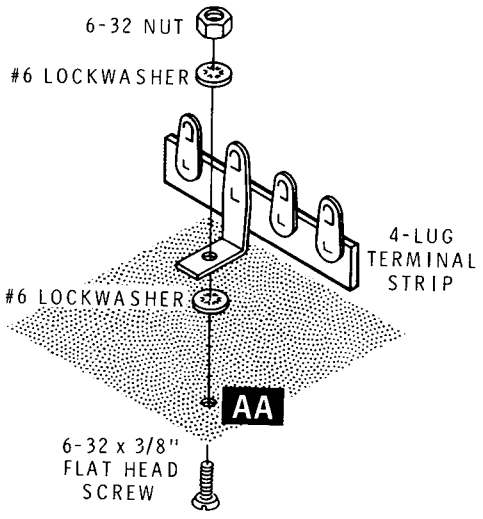
( ) Connect one end of a 6-1/2" yellow wire to circuit board hole W (S-1). Route the wire rearward, through support bracket grommet B. Then connect the free end of the wire to terminal strip C lug 1 (S-2).

( ) Connect one end of a 6-1/4" white-yellow wire to circuit board hole V (S-1). Route this wire rearward, through support bracket grommet B. Then connect the free end of the wire 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. Then 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, through support bracket grommet B. Then connect the free end of the wire to switch SW1 lug 8 (S-1).

Set the main circuit board assembly aside temporarily.



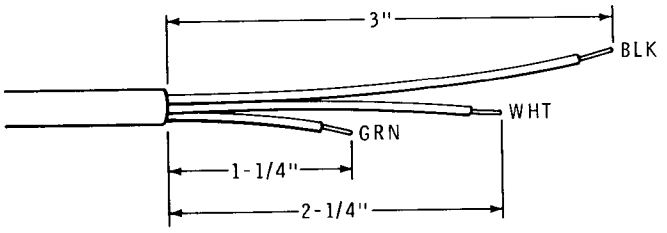
Detail 5-1A

CABINET ASSEMBLY AND WIRING

Refer to Pictorial 5-1 (Illustration Booklet, Page 6) for the following steps.

( ) Position the cabinet bottom as shown in the Pictorial. Note the locations of holes AB and AC in the cabinet bottom for the proper orientation.

( ) Refer to Detail 5-1A and mount a 4-lug terminal strip at AA in the cabinet bottom as shown. Use a 6-32 × 3/8" flat head screw, two #6 lockwashers, and a 6-32 nut. Be sure to position the terminal strip as shown in the Pictorial before you tighten the hardware.



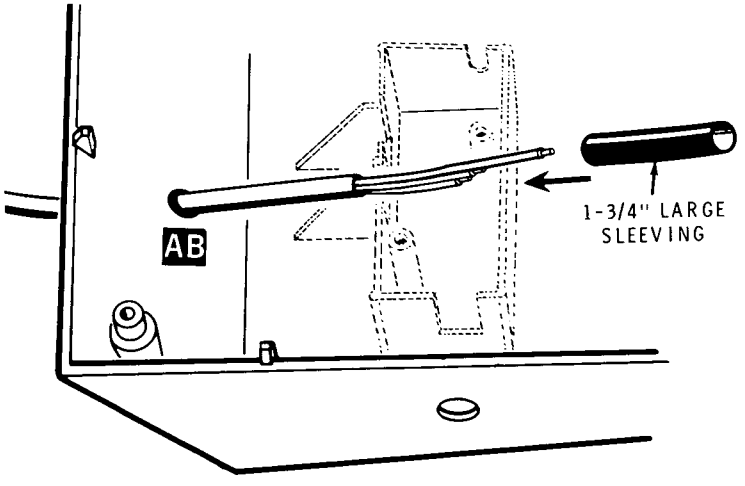
Detail 5-1B

( ) Refer to Detail 5-1B and prepare the free end of the line cord as follows:

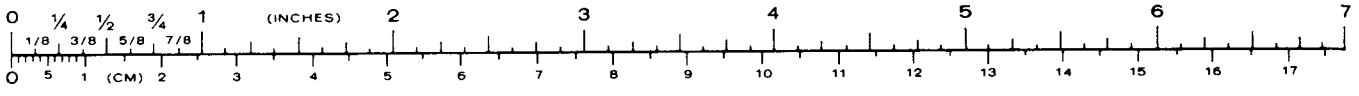
1. Carefully remove the outer insulation of the line cord for 3".
2. Cut the exposed white lead to a length of 2-1/4" and the green lead to a length of 1-1/4". Then remove 1/4" of insulation from the end of each lead.
3. Twist together the fine wire strands at the end of each lead. Then melt a small amount of solder onto the end of each lead to hold the fine strands together.

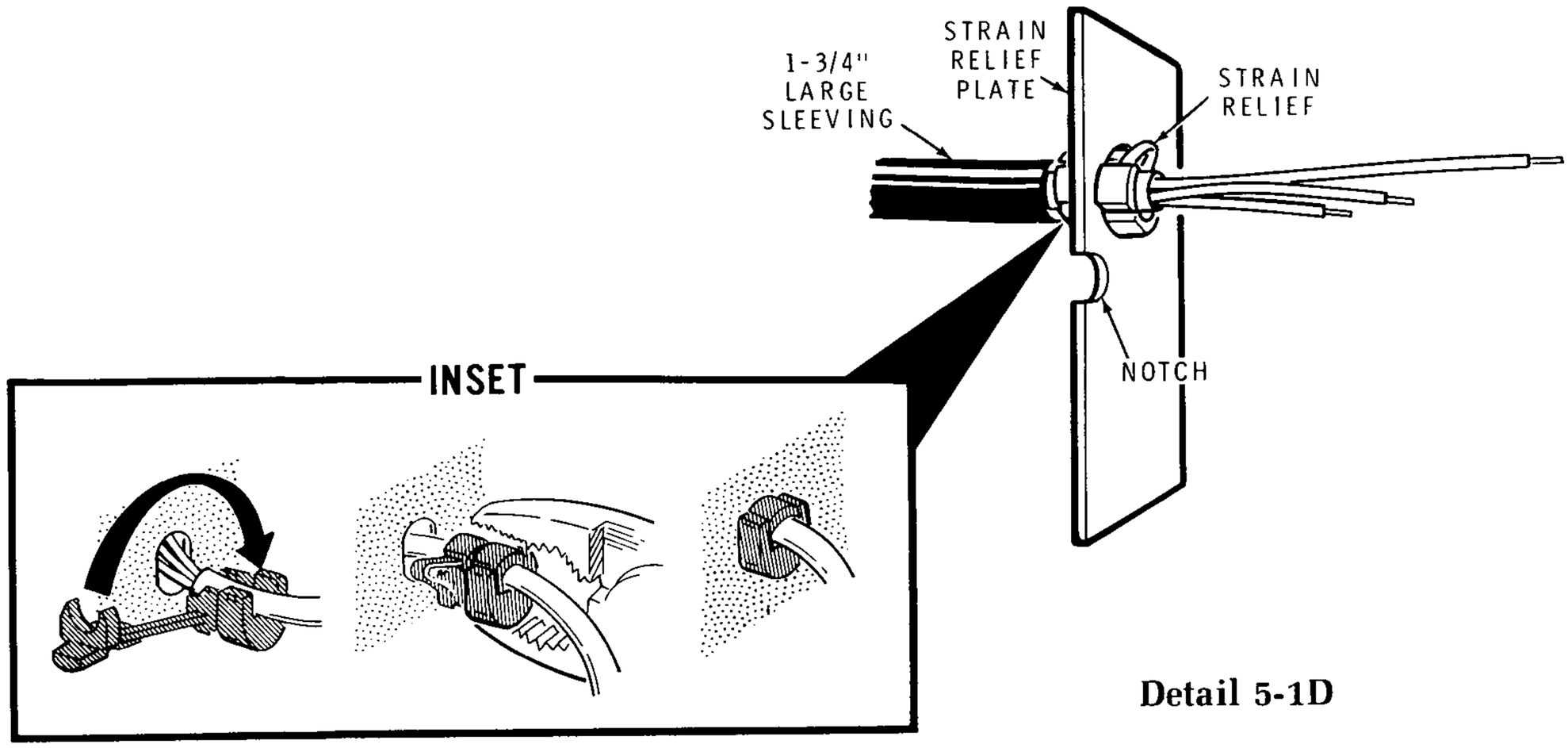
( ) Refer to Detail 5-1C and push the prepared end of the line cord through hole AB in the cabinet bottom (from the outside of the cabinet bottom).

( ) Cut a 1-3/4" length of black heat-shrinkable sleeving. Then refer again to Detail 5-1C and slide the sleeving onto the free end of the line cord.



Detail 5-1C





Detail 5-1D

( ) Refer to Detail 5-1D and install a strain relief on the line cord so the smaller end of the strain relief is even with the end of the outer insulation on the line cord. Then position the strain relief plate as shown (note the locations of the large hole and the notch) and push the strain relief into the large hole until it locks in place.

( ) Position the sleeving so it is against the strain relief. Then use the heat from a match, hot soldering iron, or candle to shrink the sleeving so it is tight against the line cord.

( ) Refer to Detail 5-1E and slide the strain relief plate into place in the cabinet bottom. Be sure the plate is against the wall of the box in the cabinet bottom.

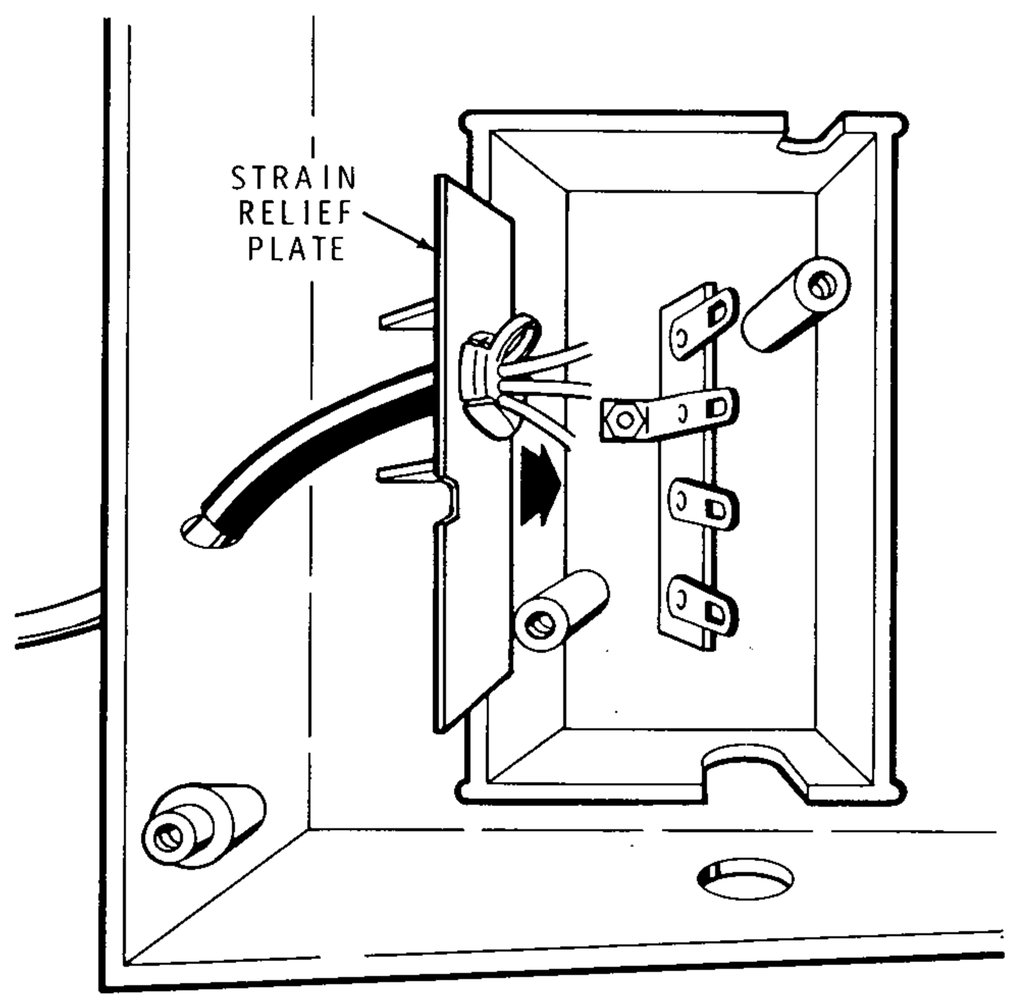
NOTE: When you are directed to "make a mechanically secure connection," as in the following steps, refer to the inset drawing on the Pictorial.

( ) Connect the free end of the white line cord lead to terminal strip AA lug 1 (NS). Make a mechanically secure connection.

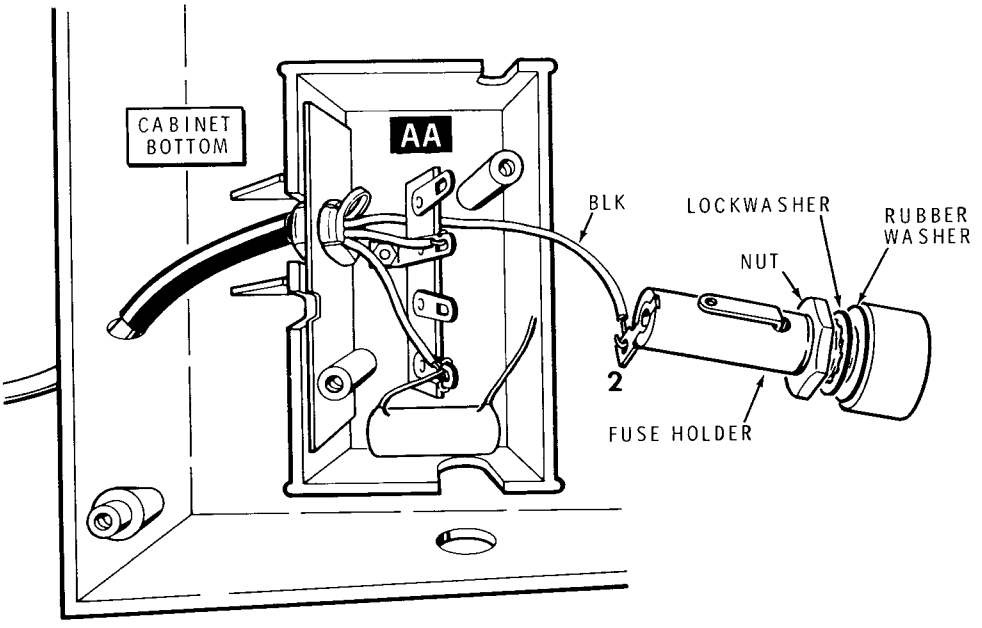
( ) Connect the free end of the green line cord lead to terminal strip AA lug 3 (NS). Make a mechanically secure connection.

( ) Route the free end of the black line cord lead under terminal strip AA as shown; it will be connected later.

( ) C1: Cut one lead of a .047 μF Mylar capacitor to 3/4" and the other lead to 1-1/4". Then position the body of the capacitor down against the cabinet bottom as shown and connect the shorter lead to terminal strip AA lug 1 (NS). Make a mechanically secure connection. The other capacitor lead will be connected later.



Detail 5-1E



Detail 5-2A

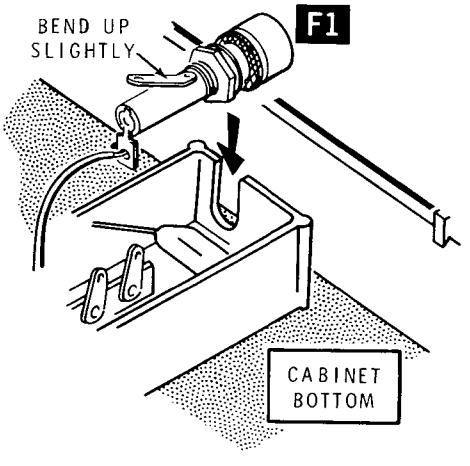
Refer to Pictorial 5-2 (Illustration Booklet, Page 6) for the following steps.

( ) Refer to Detail 5-2A and slide the rubber washer, lockwasher, and nut (supplied with the fuseholder) onto the fuseholder. Then position the fuseholder near terminal strip AA in the cabinet bottom and connect the black line cord lead, coming from under the terminal strip, to fuseholder lug 2 (S-1). Make a mechanically secure connection. Also be sure to connect the lead to the side of the lug as shown.

( ) Refer to Detail 5-2B and mount the fuseholder in the cabinet bottom at F1 as shown. Be sure lug 1 of the fuseholder is up as shown.

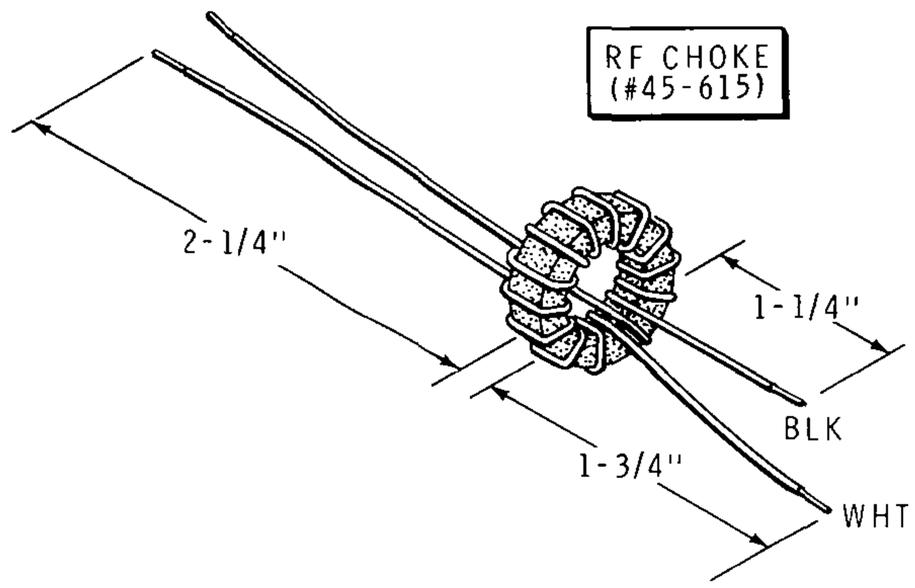
( ) Connect the free lead of capacitor C1 to fuseholder F1 lug 1 (NS). Make a mechanically secure connection.

( ) Position the RF choke (#45-615) as shown in Detail 5-2C. Then cut the leads of the choke to the indicated lengths.

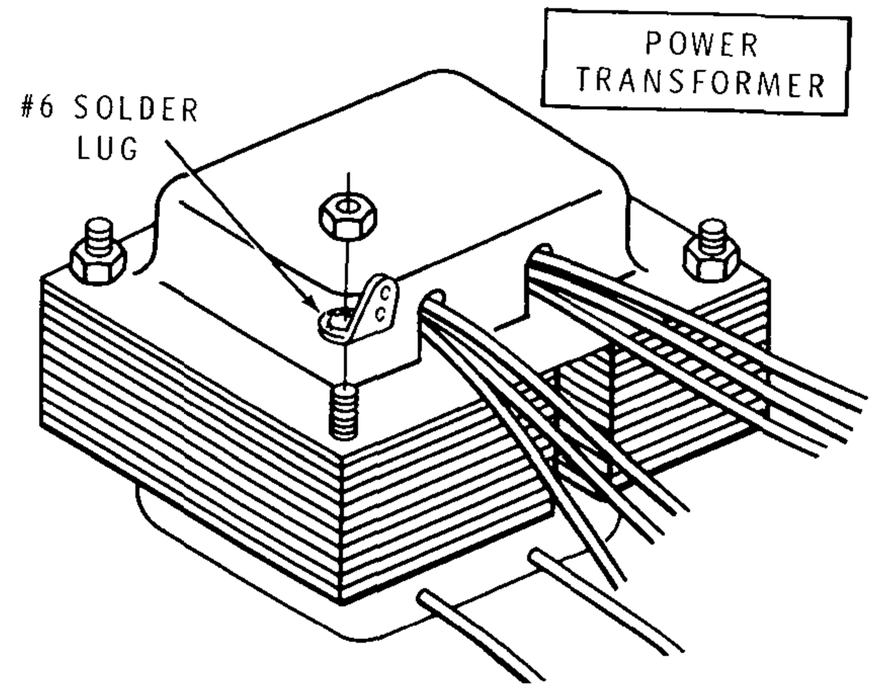


Detail 5-2B

Heathkit®



Detail 5-2C



Detail 5-3A

( ) FL1: Position the prepared RF choke in the indicated corner of the box in the cabinet bottom as shown so the shorter choke leads are toward terminal strip AA.

Connect the leads of the RF choke as follows. Be sure to make mechanically secure connections. Also be sure to dress each lead down against the cabinet bottom.

( ) Longer white lead to terminal strip AA lug 1 (S-3).

( ) Longer black lead to fuseholder F1 lug 1 (S-2).

( ) F1: Use a screwdriver to remove the cap from the fuseholder, if this has not already been done. Then insert a 3/8-ampere slow-blow fuse into the fuseholder and replace the cap. Do not overtighten the cap.

( ) Remaining white lead to terminal strip AA lug 2 (NS).

( ) Remaining black lead to terminal strip AA lug 4 (NS).

Refer to Pictorial 5-3 (Illustration Booklet, Page 6) for the following steps.

( ) Cut both black power transformer leads to 5-1/2". Measure the leads from the point where they come out of the transformer. Then twist together the fine strands at the end of each black lead and apply just enough solder to hold the fine strands together. Do not shorten the other six transformer leads.

( ) Refer to Detail 5-3A and remove the hardware from the indicated corner of the power transformer. Then use the hardware to mount a #6 solder lug at this corner of the transformer.

( ) T1: Mount the transformer to the cabinet bottom as shown in the Pictorial. Use two #6 × 1-1/8" self-tapping screws. Be sure to position the transformer as shown.

( ) Connect either black power transformer lead to terminal strip AA lug 2 (NS). Make a mechanically secure connection.

( ) Connect the remaining black power transformer lead to terminal strip AA lug 4 (NS). Make a mechanically secure connection.

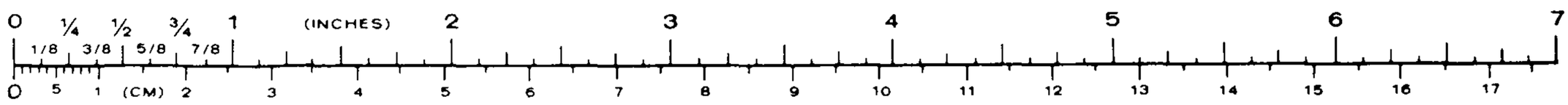
( ) Prepare a 5" black wire. Then connect the wire from terminal strip AA lug 3 (NS) to the lower hole in the solder lug on the power transformer (NS). Make mechanically secure connections.

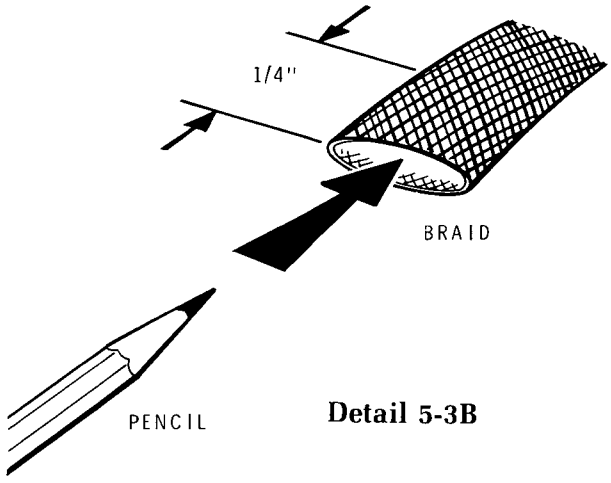
( ) Locate the two .001 μF (1000 pF) ceramic capacitors and cut the leads of each capacitor to 1/2". Use these capacitors in the next two steps.

( ) C3: Connect a .001 μF (1000 pF) ceramic capacitor between terminal strip AA lugs 2 (S-3) and 3 (NS). Make mechanically secure connections.

( ) C2: Connect a .001 μF (1000 pF) ceramic capacitor between terminal strip AA lugs 3 (S-4) and 4 (S-3). Make mechanically secure connections.

( ) Locate the length of flat braid. Then cut the braid to a length of 5-1/2".





Detail 5-3B

( ) Refer to Detail 5-3B and use the tip of a sharp pencil to force one end of the 5-1/2" braid open. Then push the open end of the braid onto the solder lug on the power transformer. Crimp and solder the braid to the lug. Also be sure to solder the black wire that was previously connected to the solder lug. NOTE: The free end of the braid will be connected later.

Refer to Pictorial 5-4 for the following steps.

Position the main circuit board assembly near the cabinet bottom as shown in the Pictorial. Then connect the remaining power transformer leads to the circuit board as follows. Solder each lead to the foil side of the circuit board as you connect it and cut off any excess lead ends.

( ) Either green lead to hole AC.

( ) Remaining green lead to hole AA.

( ) Either red lead to hole AD.

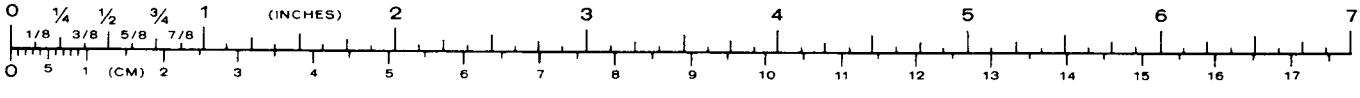
( ) Remaining red lead to hole AF.

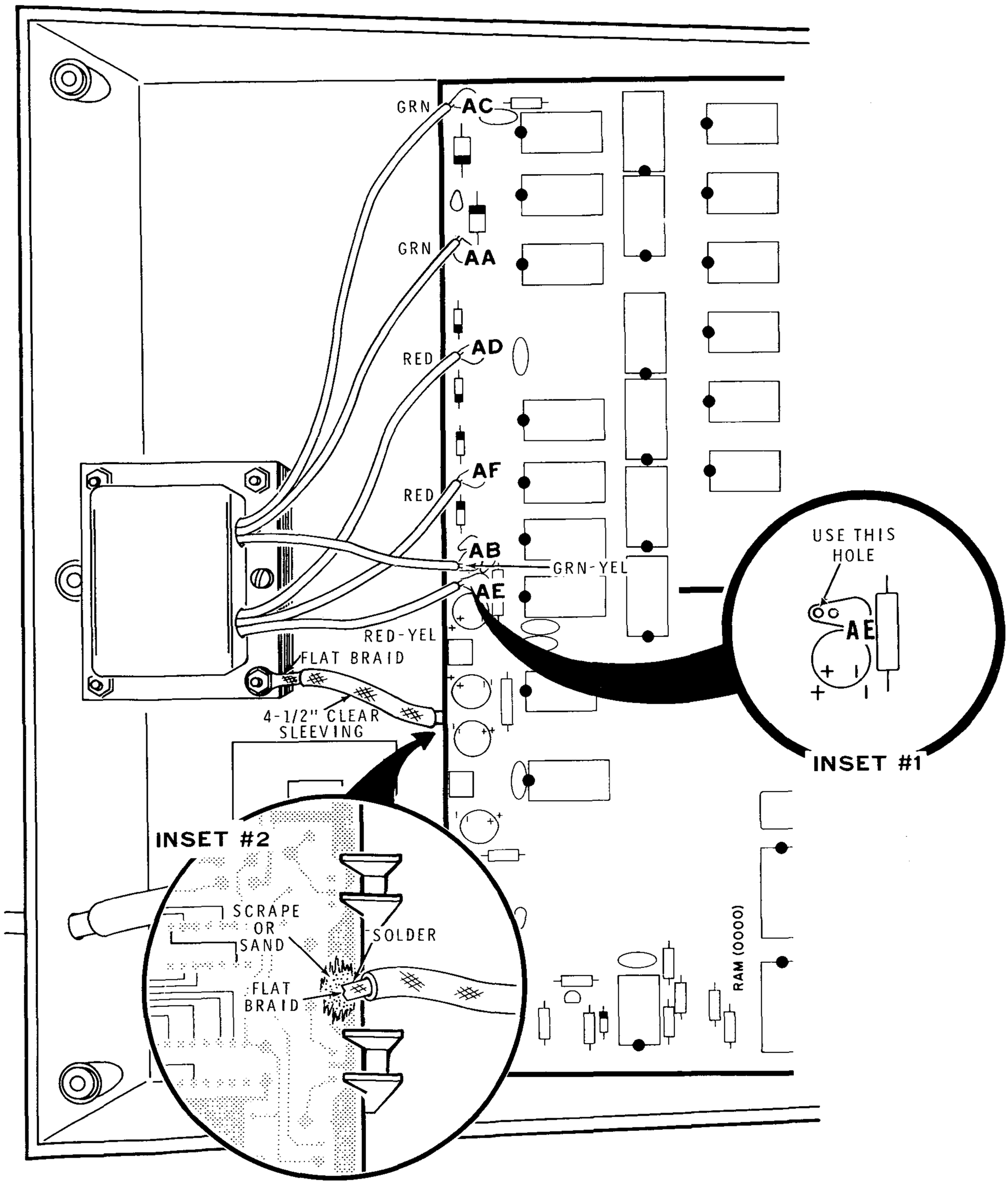
( ) Green-yellow lead to hole AB.

( ) Red-yellow lead to hole AE. Refer to inset #1 for correct hole location.

( ) Refer to inset #2 on the Pictorial and scrap or sand the area of the circuit board shown to expose the bare foil.

( ) Cut a 4-1/2" length of clear sleeving and slide it onto the flat braid (coming from the solder lug on the power transformer). Then refer to the inset drawing on the Pictorial and solder the free end of the braid to the foil side of the circuit board in the area shown.





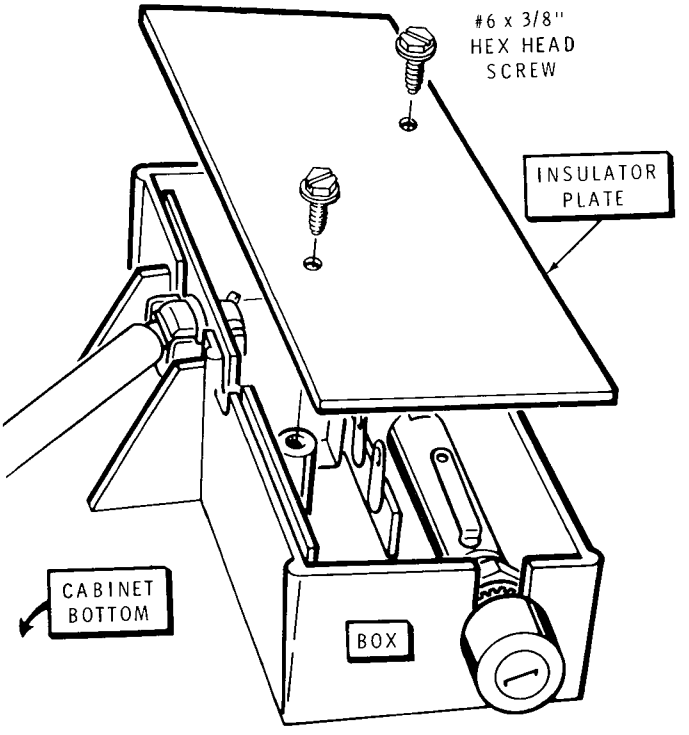
PICTORIAL 5-4

Refer to Pictorial 5-5 for the following steps.

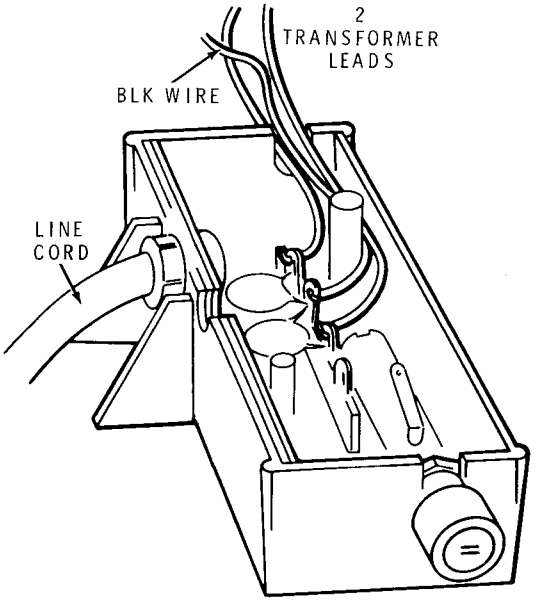
( ) Refer to Detail 5-5A and route the leads and wires coming from terminal strip AA as shown.

( ) Mount the insulator plate to the box in the cabinet bottom as shown in the Pictorial. Use two #6 × 3/8" hex head screws. Be careful not to pinch any wires between the plate and the box.

This completes the "Step-by-Step Assembly" of your Trainer. Check the unit over carefully at this time to be sure there are no poor solder connections, solder bridges, loose wire ends, solder splashes, etc. Then proceed to "Initial Tests."

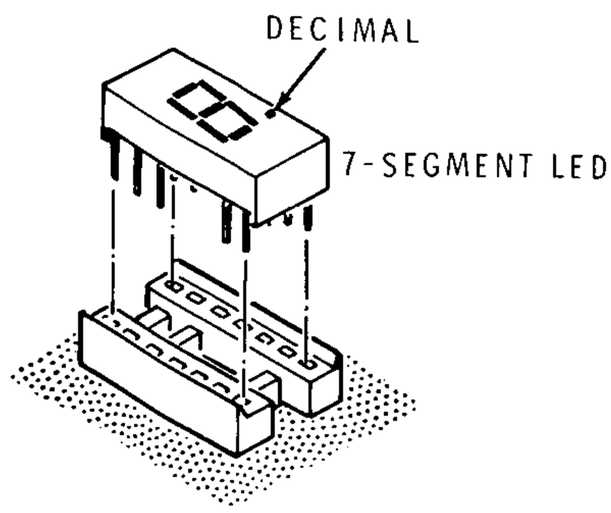


PICTORIAL 5-5

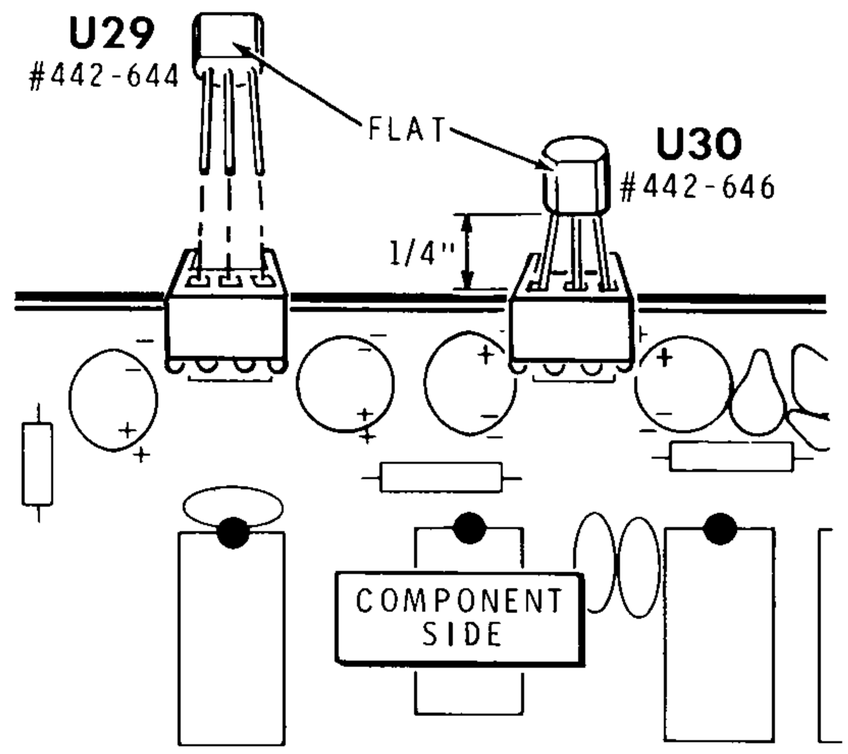


Detail 5-5A

INITIAL TESTS



Detail 6-1A



Detail 6-1B

Refer to Pictorial 6-1 (Illustration Booklet, Page 7) for the following steps.

( ) Position the main circuit board part way out of the cabinet bottom as shown in the Pictorial.

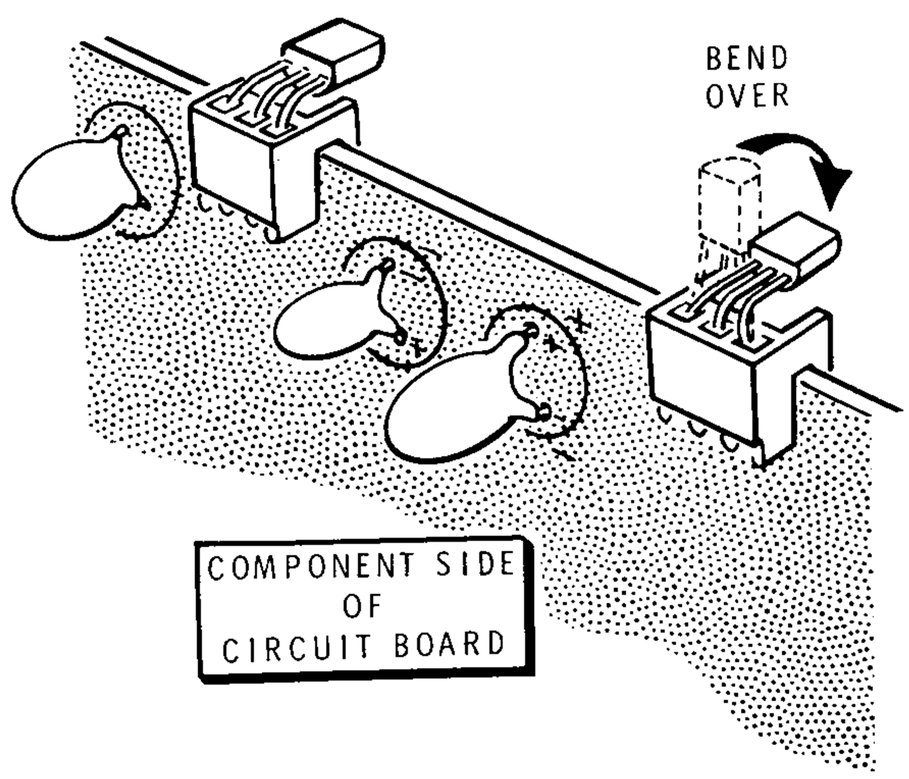
( ) Refer to Detail 6-1A and carefully install a 7-segment LED at V1. NOTE: Be sure a decimal point is at the **bottom right** as shown in the Pictorial. (Do not shorten the leads of the LED; they act as heat sinks.)

( ) Similarly, install the remaining 7-segment LED's at V2, V3, V4, V5, and V6.

( ) U29: Refer to Detail 6-1B and carefully push a 78L12 IC (#442-644) into the socket at U29 as shown.

( ) U30: Similarly, push a 79L12 IC (#442-646) into the socket at U30.

( ) Refer to Detail 6-1C and bend integrated circuits U29 and U30 over as shown. This will provide clearance for the cabinet, which will be installed later.



Detail 6-1C

VOLTAGE TESTS

NOTE: If at any time during the following tests you fail to obtain the proper 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 of this Manual (Page 105).

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

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

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

( ) Plug the line cord plug into an AC outlet. The red LED next to the POWER switch should light immediately and will remain lit, 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 that are labeled "+5" and "GND." Push one end of a prepared wire into each of these blocks. Make sure the free ends of these wires do not touch anything.

( ) Set your voltmeter to read +5 volts DC. Then connect the positive meter lead to the bare end of the wire at "+5" and the negative meter lead to the bare end of the wire at "GND."

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

( ) The meter should indicate between 4.5 and 5.5 volts.

( ) Set the voltmeter to read +12 volts DC. Then move the positive meter lead and the test wire from "+5" to "+12." The meter should indicate between 10.8 and 13.2 volts.

( ) 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." The meter should indicate between 10.8 and 13.2 volts.

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

TESTS CONTINUED

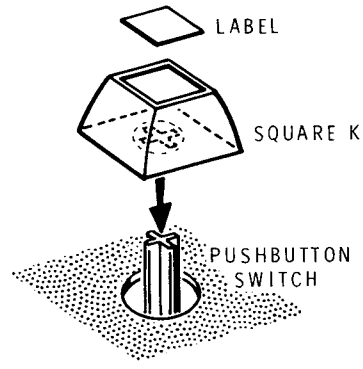
( ) Plug in the line cord and push down on the right side of the POWER switch (SW1), if this has not already been done. The red LED next to the POWER switch should light. (This LED will be lit regardless of the power switch setting.

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

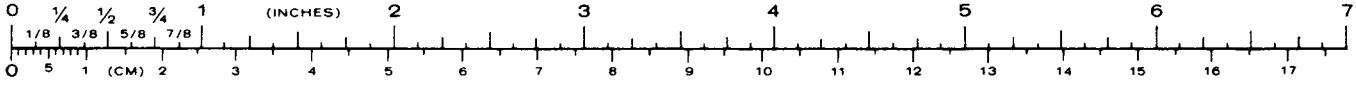
( ) Push the POWER switch to STANDBY and unplug the line cord from the AC outlet.

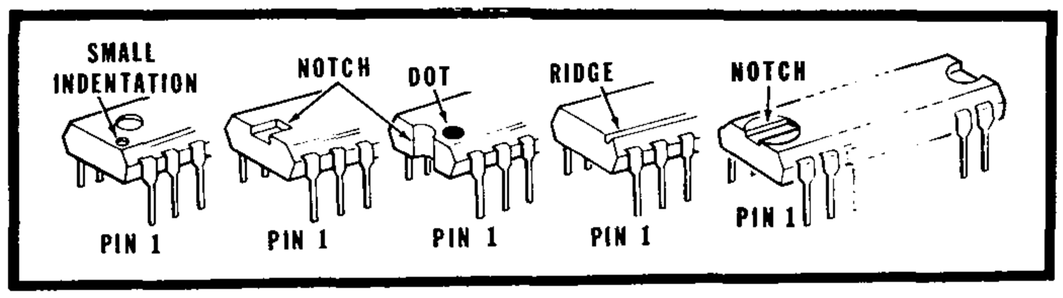
Refer to Pictorial 6-2 (Illustration Booklet, Page 8) for the following steps.

( ) Refer to Detail 6-2A and push 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.



Detail 6-2A





Detail 6-2B

( ) Similarly, push square knobs on the remaining pushbutton switches.

( ) If any of the pushbutton switches are crooked, reheat the solder connections on the switch and reposition it.

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

( ) Locate the red label set. Then, one at a time, remove one of the red labels from the backing paper and press it squarely over a 7-segment LED. (You should have two labels left over.)

NOTE: Some integrated circuits are packed in conductive foam. These IC's are rugged and reliable components. However, normal static electricity discharged from your body through an IC pin to another object can damage the IC. To prevent the IC from becoming damaged, always hold it in one hand; then touch the other object (table, circuit board, etc.) with your other hand before you touch the object with the IC itself. Read all of the following information before you install the IC's.

The pins on the IC's are bent out at an angle, so they do not line up with the holes in the IC socket. Do **NOT** try to install an IC without first bending the pins as described below. To do so may damage the IC pins or the socket, causing intermittent contact.

Before you install an IC, lay it down on its side as shown below and very carefully roll it toward the pins to bend the lower pins into line. Then turn the IC over and bend the pins on the other side in the same manner.

Refer to Detail 6-2B to identify the pin 1 end of each IC. Make sure that the pin 1 end is positioned over the index mark on the circuit board as shown below. Also, make sure that all of the pins are started into the socket. Then press the IC firmly into the socket. NOTE: An IC pin can become bent under the IC and it will appear as though it is correctly installed in the socket.

( ) U14: Install a 2114 IC (#443-764) in the socket at U14.

( ) U15: Install a 2114 IC (#443-764) in the socket at U15.

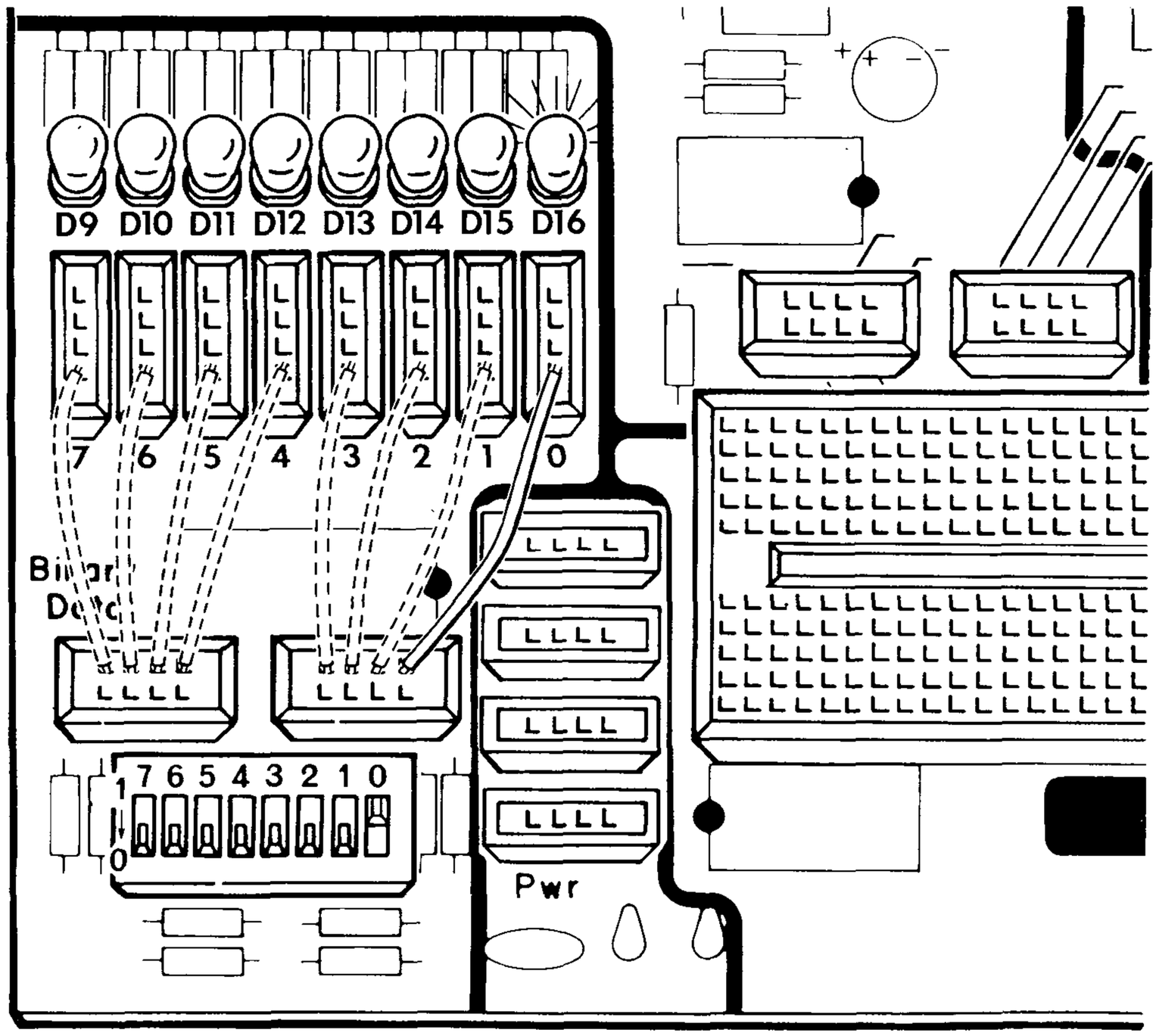
( ) U13: Install a 40097 IC (#443-720) in the socket at U13.

( ) U12: Install an ROM IC (#444-76) in the socket at U12.

( ) U11: Install an MC6808P IC (#443-939) in the socket at U11.

( ) Prepare a 4" yellow wire.

( ) Plug in the line cord and push the POWER switch (SW1) to ON.



Detail 6-2C

Refer to Detail 6-2C when you perform the next 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 its corresponding 4-pin connector block, from "0" to "7." In addition, these connectors and the LED's have corresponding switches on the slide switch assembly (SW2) and pairs of connector pins in the two 8-pin connector blocks located immediately above the slide switch assembly.

( ) 1. Connect the 4" jumper wire from 4-pin connector block number "0" to 8-pin block pair "0" (as shown in the Detail). Operate slide switch SW2 section 0 back and forth and observe that the right (zero) LED lights and then goes out.

( ) 2. Move the jumper wire to the "1" connector blocks (second from the right). Operate the corresponding slide switch and observe that the "1" LED lights and then goes out.

( ) 3. Progressively, 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 lights. Then remove the wire.

OPERATIONAL TESTS

This section of the Manual checks the basic Trainer functions to make sure they are operating properly. The entries that you will make on the keyboard are not necessarily related to your actual use of the unit. Refer to the "Operation" section of this Manual to learn how to actually use each function.

Refer to Pictorial 6-3 (Illustration Booklet, Page 8) to identify the function of each keyboard key.

NOTE: If you encounter any trouble as you perform the following steps, push the POWER switch to STANDBY and unplug the line cord. Then refer to the "In Case of Difficulty" section of this Manual.

The following abbreviations are used on the Trainer 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. Each numbered step in the following charts shows which number or letter key to push, and what the resultant readout should be. Always push the keys in the sequence shown. If you make an incorrect entry, return to step 1.

2. In the following charts, the symbol "*" indicates a blank readout indication. The symbol "×" indicates a random figure.

3. When you make two-digit entries, the indicated Readout display will appear after you release the second key.

( ) Perform the tests in "Test Chart #1".

TEST CHART #1

STEP	FIRST PRESS:	THEN PRESS:	READOUT**
1	RESET		CPU*UP.
2	EXAM E		____Ad.
3		0 ACCA 1 ACCB 2 PC 3	0123XX
4	CHAN C		0123__
5		INDEX 4 CC 5	012345
6	EXAM E		____Ad.
7		SP 6 RTI 7 SS 8 BR 9	6789XX
8	CHAN C		6789__
9		AUTO A BACK B	6789Ab
10	EXAM E		____Ad.
11		CHAN C DO D EXAM E FWD F	CdEFXX

** Shows the readout as presented on the LED's after you release the key.

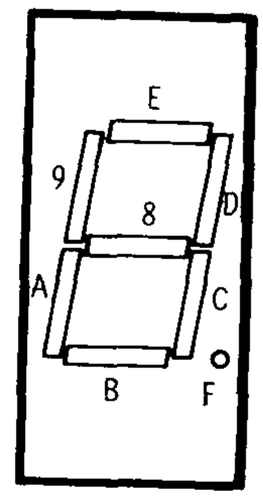
If all the tests in "Test Chart #1" produced the proper results, the Trainer keys are operating properly. Perform the steps in "Test Chart #2" to check the memory circuits in the Trainer.

TEST CHART #2

STEP	FIRST PRESS:	THEN PRESS:	READOUT
1	RESET		CPU* UP.
2	AUTO A		_ _ _ _ Ad.
3		0 0 0 0 (Memory address)	0000 _ _
4		SS 8 SP 6 (Load Accumulator A w/following two characters)	0001 _ _
5		0 ACCA 1 (Data for first step)	0002 _ _
6		BACK B RTI 7 (Store Accumulator A at extended address of following four characters)	0003 _ _
7		CHAN C ACCA 1	0004 _ _
8		SP 6 FWD F	0005 _ _
9		RTI 7 EXAM E (Jump to extended address of following four bytes of information)	0006 _ _
10		0 0	0007 _ _
11		0 0	0008 _ _
12	RESET	(To terminate AUTO addressing sequence)	CPU* UP.
13	DO D		_ _ _ _ do.
14		0 0 0 0	.* * * * *
15	RESET		CPU* UP.

The simple program that you entered in "Test Chart #2" stored specific information in the Trainer's internal memory. This information directed the microprocessor circuit to turn on the decimal point in the "H" LED (V1). Steps 13 and 14 proved the program.

Note the characters "6F" in step 8. This is the information that told the microprocessor that you wanted to address the "H" LED, and in particular, the decimal point (the "6" addressed the LED and the "F" addressed the decimal point). You can similarly address any segment of an LED as shown in Detail 6-3A. Thus, to turn on each segment of the "H" (or "6") LED in turn, you must change the terminal character to agree with the segment address.



7-SEGMENT LED

Detail 6-3A

( ) Perform the steps in "Test Chart #3." NOTE: This program addresses and examines the top bar of the "H" LED.

TEST CHART #3

STEP	FIRST PRESS:	THEN PRESS:	READOUT
1	RESET		CPU*UP.
2	EXAM E		____Ad.
3		0 0 0 INDEX 4	00046F
4	CHAN C		0004__
5		SP 6 EXAM E : "H" LED, TOP BAR.	00046E
6	RESET		CPU*UP.
7	DO D		____do.
8		0 0 0 0	_ *****
9	RESET		CPU*UP.

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 "Test Chart #3," with one exception. After you push the "CHAN/C" key in step 4, enter "6**D**;" then proceed with the steps 6 through 8. Next time, at step 5, enter "6**C**", and proceed. Similarly, at step 5 of each repetition, enter "6**B**, 6**A**, 6**9**, and 6**8**." Refer to Detail 6-3A to determine which LED segments should light.

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 "5**F**," and proceed with steps 6 through 8. Note that the decimal on the "I" LED lights. Then, at step 5, one at a time, enter "5E, 5D, 5C, 5B, 5A, 59, and 58." You can call up the segments of the remaining four LED's in the same way by using "4F, 3F, 2F, and 1F" at step 5 to light the respective decimal segments.

( ) Push the POWER switch to STANDBY and unplug the line cord.

This completes the "Operational Tests" of your Microprocessor Trainer. Proceed to "Final Assembly."

FINAL ASSEMBLY

SYSTEM EXPANSION

The circuit board has provision for you to install a 40-pin connector for system expansion. If you do not wish to install a connector at this time, proceed directly to "Cabinet Completion," below.

You can obtain these 40-pin connectors from the following manufacturers. NOTE: The connectors must have .025" square pins on .100" centers. Some of the connectors are actually single strips of connectors that you must cut to length.

Heath Company
432-1062

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

Molex
22-04-2201 (2 strips required)

AMP
2-87215-0
2-87543-0

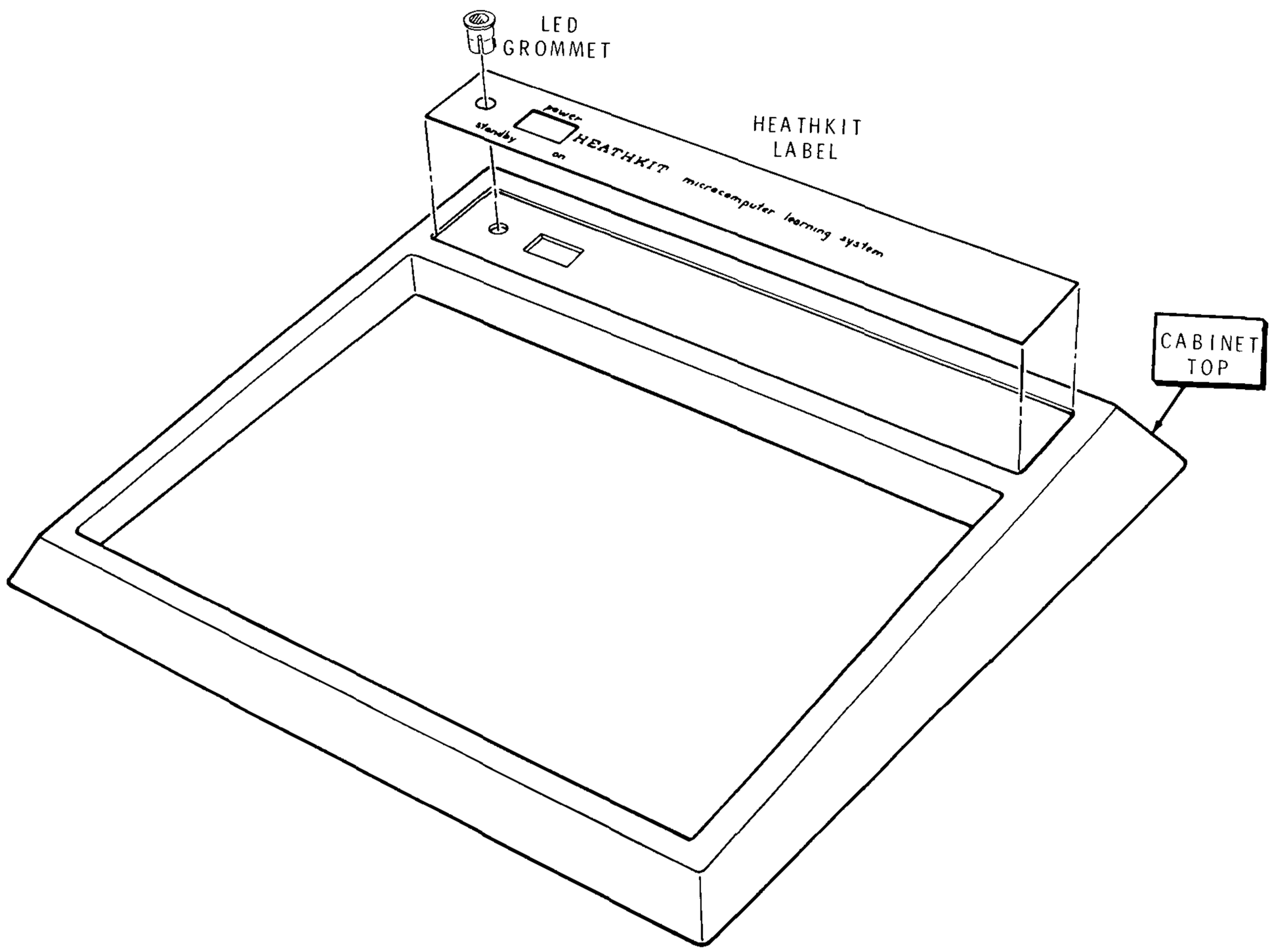
Install your 40-pin connector on the circuit board between integrated circuits U2 and U3. Then connect eight wires from the eight circuit board holes that connect the connector data pins to the eight data holes near integrated circuits U9 and U10. (Refer to the "Schematic" and the "Circuit Board X-Ray 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.

CABINET COMPLETION

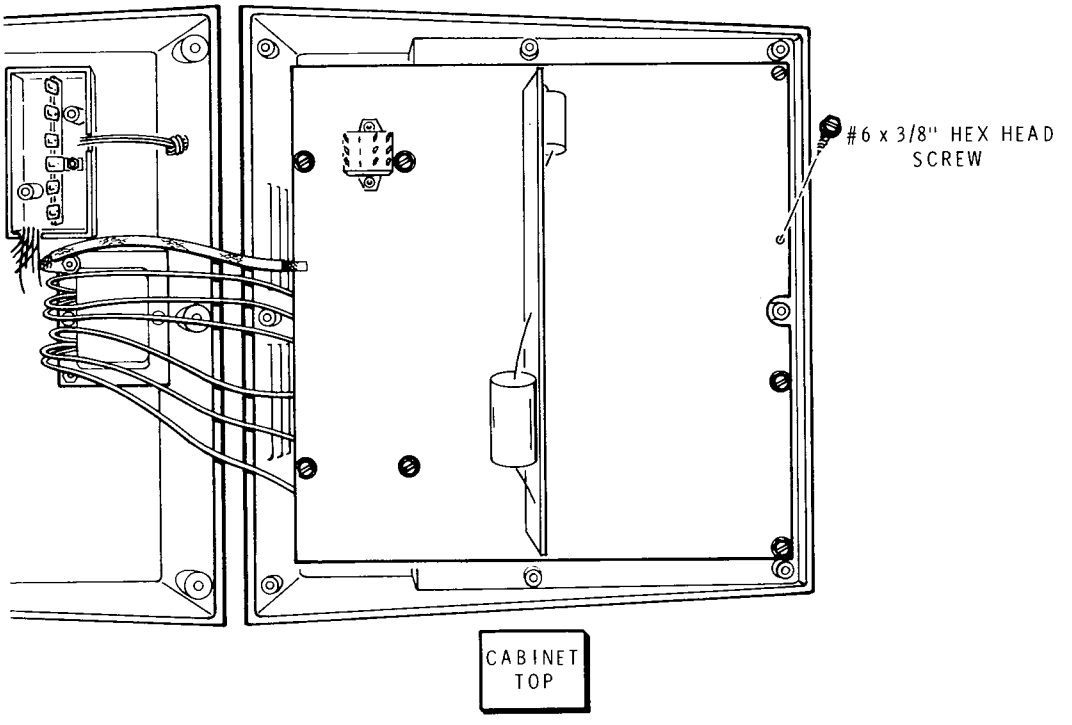
Refer to Pictorial 7-1 for the following steps.

( ) Carefully peel the backing paper from the "Heathkit" label. Then 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.



PICTORIAL 7-1



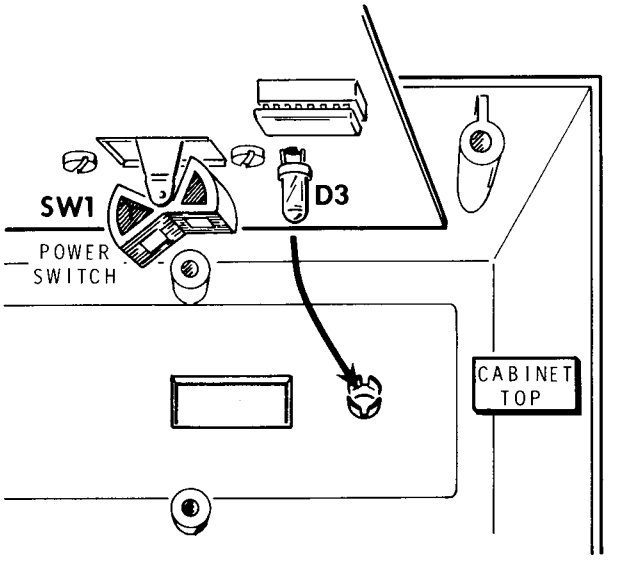
PICTORIAL 7-2

Refer to Pictorial 7-2 for the following steps.

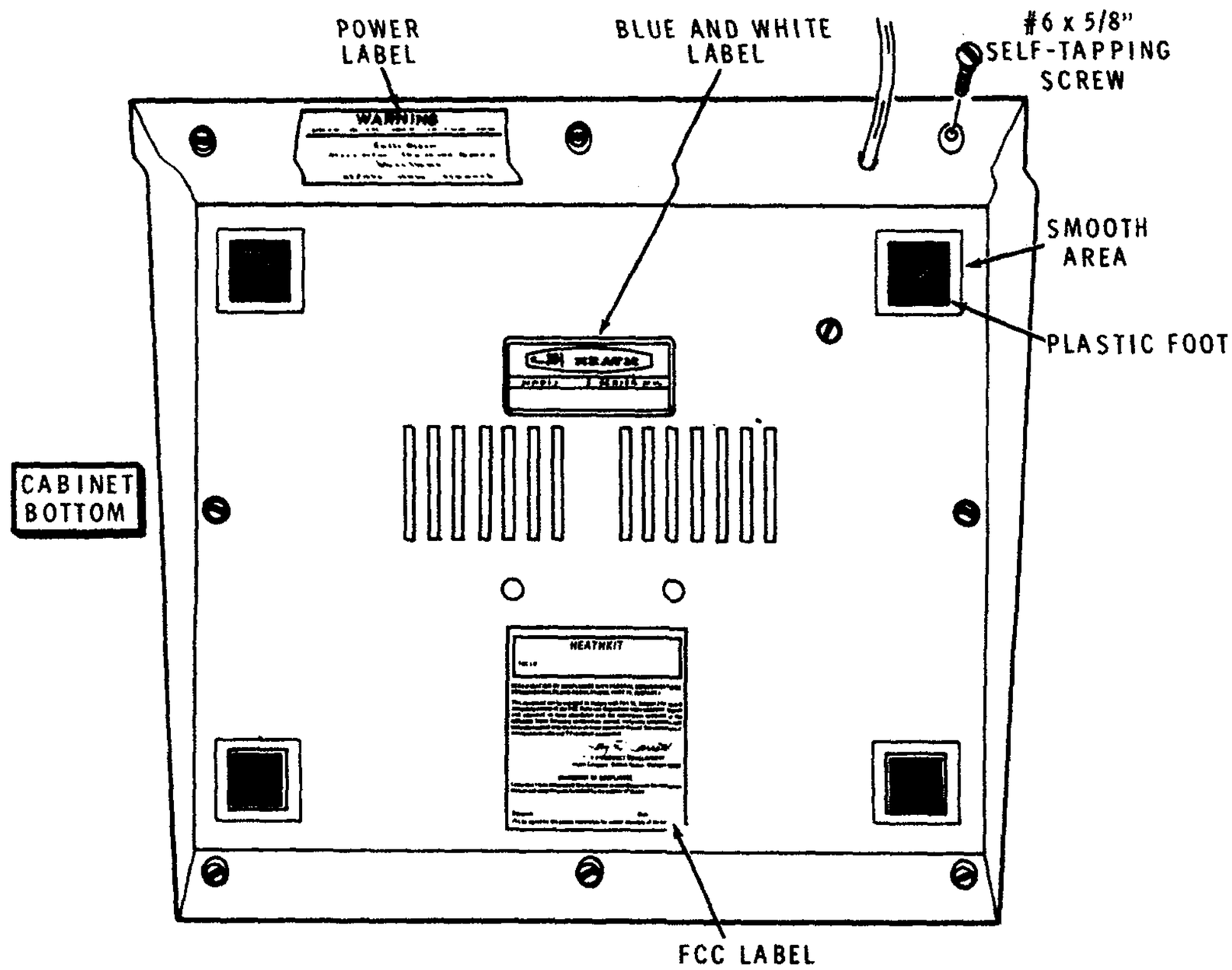
( ) Unplug the line cord, if this has not already been done.

( ) 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 LED D3, 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 × 3/8" hex head screws.



Detail 7-2A



PICTORIAL 7-3

Refer to Pictorial 7-3 for the following steps.

( ) Turn the cabinet top and the 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 × 5/8" self-tapping screws.

( ) Carefully peel the backing paper from the four feet. Then press the feet in place on the cabinet bottom in the smooth areas near the four corners as shown.

( ) Carefully peel the backing paper from the blue and white label. Then press the label onto the cabinet bottom in the area shown. 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. This assures you that you will receive the most complete and up-to-date information in return.

( ) Carefully peel the backing paper from the power label. Then press the label onto the back of the cabinet bottom in the area shown.

( ) Read, sign and date the FCC label. Carefully peel the backing paper from the label and press the FCC label onto the cabinet bottom in the area shown.

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 9) 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 $4A_{16}$ (or binary 01001010).

Example: Acca.4A

Now you may change the contents of accumulator A if you wish. To do this, press the 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 $5F_{16}$ (binary 01011111).

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

Display Program Counter

PC 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

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

Display Index Register

INDEX 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 CHAN C key and then entering a new hexadecimal number.

Display Condition Codes Register

CC 5 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 103.)

Example: 001001

This register **cannot** be changed by pressing the CHAN C key.

Display Stack Pointer Register

SP 6 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 CHAN C key.

Resume User's Program

RTI 7 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

SS 8 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 CHAN 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

BR 9 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 CHAN C key as many times as necessary for the display to return to _ _ _ _ br. Then enter the correct address.

Auto Load Of Memory

AUTO A Press this key and _ _ _ _ Ad will de 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

BACK B 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 CHAN C key and then entering the new data.

Change Displayed Value

CHAN C 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

DO D 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 RTI 7 key.

Examine Memory

EXAM E 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 CHAN C key or you can step backwards or forwards through memory using the BACK B and FWD F keys.

Display Next Address

FWD F 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 CHAN C 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.

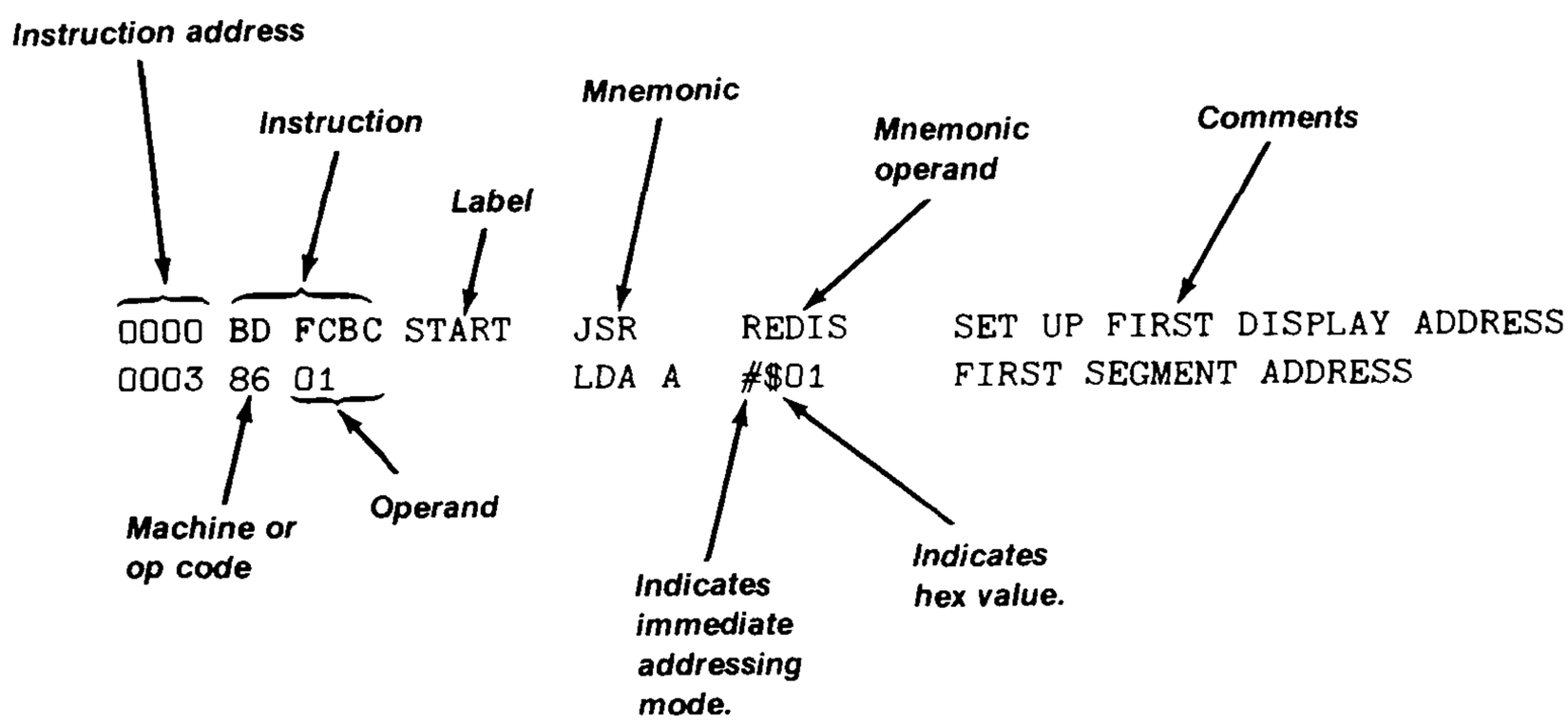
Operand: This is additional hexadecimal information required to perform the operation. It may be zero, one, or two bytes as determined by the addressing mode.

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.

Display is	→	Press	→	Enter	→	Display is	→	Press
0000XX		CHAN		BD		0000BD		FWD
0001XX		CHAN		FC		0001FC		FWD
0002XX		CHAN		BC		0002BC		FWD
0003XX		CHAN		86		000386		FWD
0004XX		CHAN		01		000401		FWD
0005XX		CHAN		etc.				

Auto

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

Display is	→	Enter
0000--		BD
0001--		FC
0002--		BC
0003--		86
0004--		01
0005--		etc.

Press RESET after last entry to exit the AUTO mode.

If Examine and Change is used, the last entry in sample program 1 (Page 67) 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 67) 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 RTI 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 RTI 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 RTI 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 RTI 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.

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

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.	0019 26
Press CC. The Z bit is set and the program should fall through the branch.	XXX1 XX
Press SS. It did.	001b 86
Press SS.	001d de
Press ACCA. (A) is loaded correctly.	Acca.01
Press SS.	001F 8C
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.	In.C10F
Press SS. Same conditional BRANCH.	0022 26
Press CC. Z is set and the program should fall through.	XXX1 XX
Press SS. It did.	0024 20
Press SS. Every instruction in the program has been run except for the conditional branch at 0022.	0000 bd
Press Reset. Clears the breakpoint at 0018	CPU UP
Press BR and enter 001F.	001F br
Press DO and enter 0000.	001F 8C
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.	In.C15F
Press SS. Conditional branch.	0022 26
Press CC. Z is clear and a branch to out should take place.	XXX0 XX
Press SS. It did.	000E bd

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 01          LDA A   #$01     RESET SEGMENT CODE
001D DE F0          LDX     DIGADD   NEXT DISPLAY
001F 8C C10F        CPX     #$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 8C C10F        CPX     #$C10F   LAST DISPLAY YET?
003C 26 F5          BNE     CLEAR
003E 8D 13          BSR     HOLD
0040 BD FCBC        JSR     REDIS    FIRST DISPLAY ADDRESS
0043 86 08          LDA A   #$08     HEX VALUE TO DISPLAY
0045 BD FE28 OUT    JSR     OUTHEX   OUTPUT CHARACTER
0048 DE F0          LDX     DIGADD   NEXT DISPLAY
004A 8C C10F        CPX     #$C10F   LAST DISPLAY YET?
004D 26 F6          BNE     OUT
004F 8D 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,$5B,$00,$00,$80  IS
007C 00
007D 30
007E 5B
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,$1C,$15,$15,$10,$15,$80  RUNNIN
00A0 1C
00A1 15
00A2 15
00A3 10
00A4 15
00A5 80
00A6 8D 03            BSR     HOLD     HOLD DISPLAY
00A8 7E 0060          JMP     START    DO AGAIN
00AB CE FF00 HOLD     LDX     #$FF00   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    0,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 FD          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 3B             FCB      $3B,$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,$5B,$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
0043 00
0044 7D
0045 15
```

```
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,$80
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 FDBB 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 38           BSR      CODE      CHANGE KEY TO FIND
0071 BD FDBB 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,$80  DP TO "C"
007D 00
007E 00
007F 00
0080 00
0081 80
              * 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,$0E,$7E,$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
```

```
00A1 00              FCB      $00,$00,$3B,$4F,$5B,$A0  YES!
00A2 00
00A3 3B
00A4 4F
00A5 5B
00A6 A0
00A7 20 B7           BRA      START    DO AGAIN
00A9 96 86   CODE    LDA A    KEY+1    CURRENT KEY VALUE
00AB 4C              INC A             NEXT KEY
00AC 97 86           STA A    KEY+1    KEY TO FIND
00AE 81 10           CMP A    #$10     CAN'T BE GREATER THAN F
00B0 26 03           BNE      GOOD
00B2 7F 0086         CLR      KEY+1    MAKE IT 0
00B5 39      GOOD    RTS
00B6 BD FD8D HIGH    JSR      OUTSTO   OUTPUT STRING
00B9 37              FCB      $37,$30,$00,$00,$00,$00,$80  HI
00BA 30
00BB 00
00BC 00
00BD 00
00BE 00
00BF 80
00C0 7E 0096         JMP      HOLD
```

SAMPLE 6

```
THIS IS A TWELVE-HOUR CLOCK PROGRAM.
THE ACCURACY DEPENDS ON THE MPU CLOCK
FREQUENCY AND THE TIMING LOOP AT START.
CHANGING THE VALUE AT 0006
CHANGES THE ACCURACY.
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 SUBROUTINES REDIS,DSPLAY.
NOTE:START THE PROGRAM AT 0004.

0001 00          HOURS   RMB     1
0002 00          MINUTE  RMB     1
0003 00          SECOND  RMB     1
0004 CE CEB3     START   LDX     #$CEB3    ADJUST FOR ACCURACY
0007 09          DELAY   DEX
0008 26 FD               BNE     DELAY     WAIT ONE SECOND
000A CE FFFF             LDX     #$FFFF    SET FIXED DELAY
000D 09          SETDEL  DEX
000E 26 FD               BNE     SETDEL
0010 C6 60               LDA B   #$60      SIXTY SECONDS, SIXTY MINUTES
0012 0D                  SEC               ALWAYS INCREMENT SECONDS
0013 8D 0F               BSR     INCS      INCREMENT SECONDS
0015 8D 10               BSR     INCMH     INCREMENT MINUTES IF NEEDED
0017 C6 13               LDA B   #$13      TWELVE HOUR CLOCK
0019 8D 0C               BSR     INCMH     INCREMENT HOURS IF NEEDED
001B BD FCBC             JSR     REDIS     RESET DISPLAY ADDRESS
001E C6 03               LDA B   #3        NUMBER OF BYTES TO DISPLAY
0020 8D 16               BSR     PRINT     DISPLAY HOURS, MINUTES, SECONDS
0022 20 E0               BRA     START     DO AGAIN
0024 CE 0003     INCS    LDX     #SECOND   POINT X AT TIME RMB
0027 A6 00       INCMH   LDA A   0, X      GET CURRENT TIME
0029 89 00               ADC A   #0        INCREMENT IF NECESSARY
002B 19                  DAA               FIX TO DECIMAL
002C 11                  CBA               TIME TO CLEAR?
002D 25 01               BCS     STORE     NO
002F 4F                  CLR A
0030 A7 00       STORE   STA A   0, X      STORE NEW TIME
0032 09                  DEX               NEXT TIME RMB
0033 07                  TPA
0034 88 01               EOR A   #1        COMPLEMENT CARRY BIT
0036 06                  TAP
0037 39                  RTS
0038 96 01       PRINT   LDA A   $01       WHAT'S IN HOURS?
003A 26 03               BNE     CONTIN    IF NOT ZERO
003C 7C 0001             INC     HOURS     MAKE HOURS ONE
003F 08          CONTIN  INX               POINT X AT HOURS
0040 7E FD7B             JMP     DSPLAY    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 C0 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
REDIS IHB OUTBYT OUTSTR

```
0000 BD FCBC START  JSR     REDIS    FIRST DISPLAY AT "H"
0003 BD FE09        JSR     IHB      INPUT BRANCH ADDRESS
0006 16             TAB              PUT IT IN B
0007 BD FE09        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 AND B
001B 8B 02          ADD A   #$02     ADJUST 2 BYTES
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.

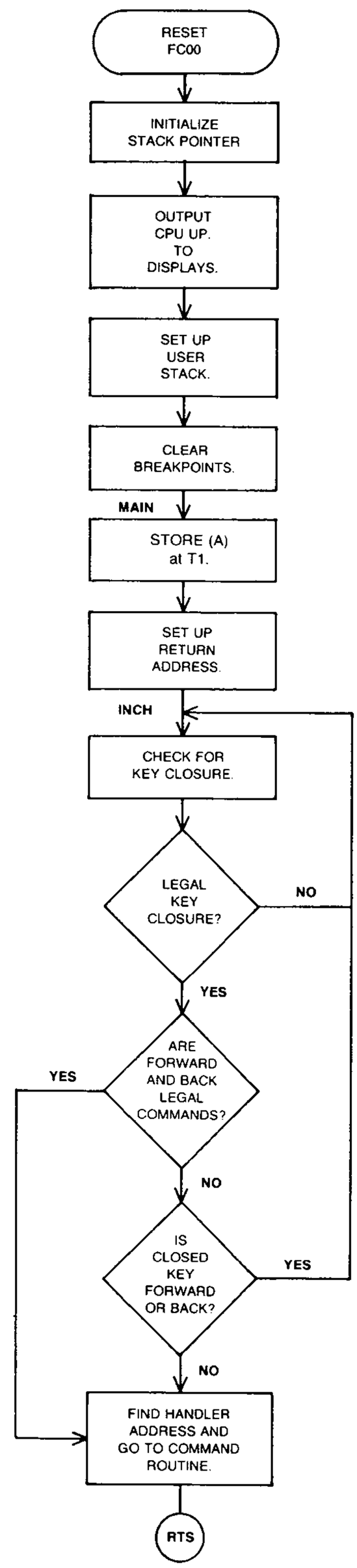
Reset first initializes the stack pointer to 00EB and outputs CPU UP, to the displays. The index register is set to 00CB (the start of the user's stack) and this value is stored in the user's stack pointer at location 00F2.

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.

RESET/MAIN ROUTINE



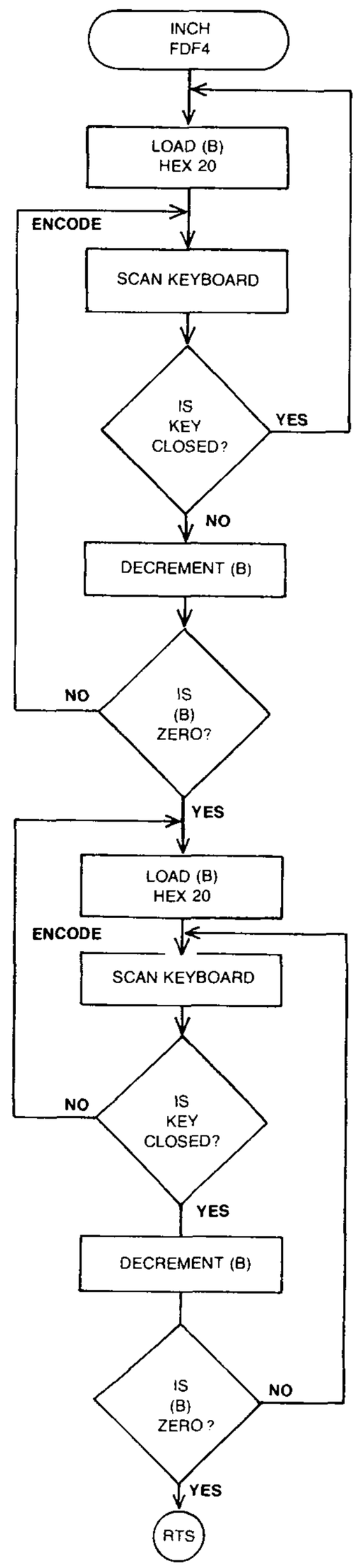
INCH Routine

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.

INCH ROUTINE

ENTRY: None.

EXIT: (A) contains hex value of closed key.



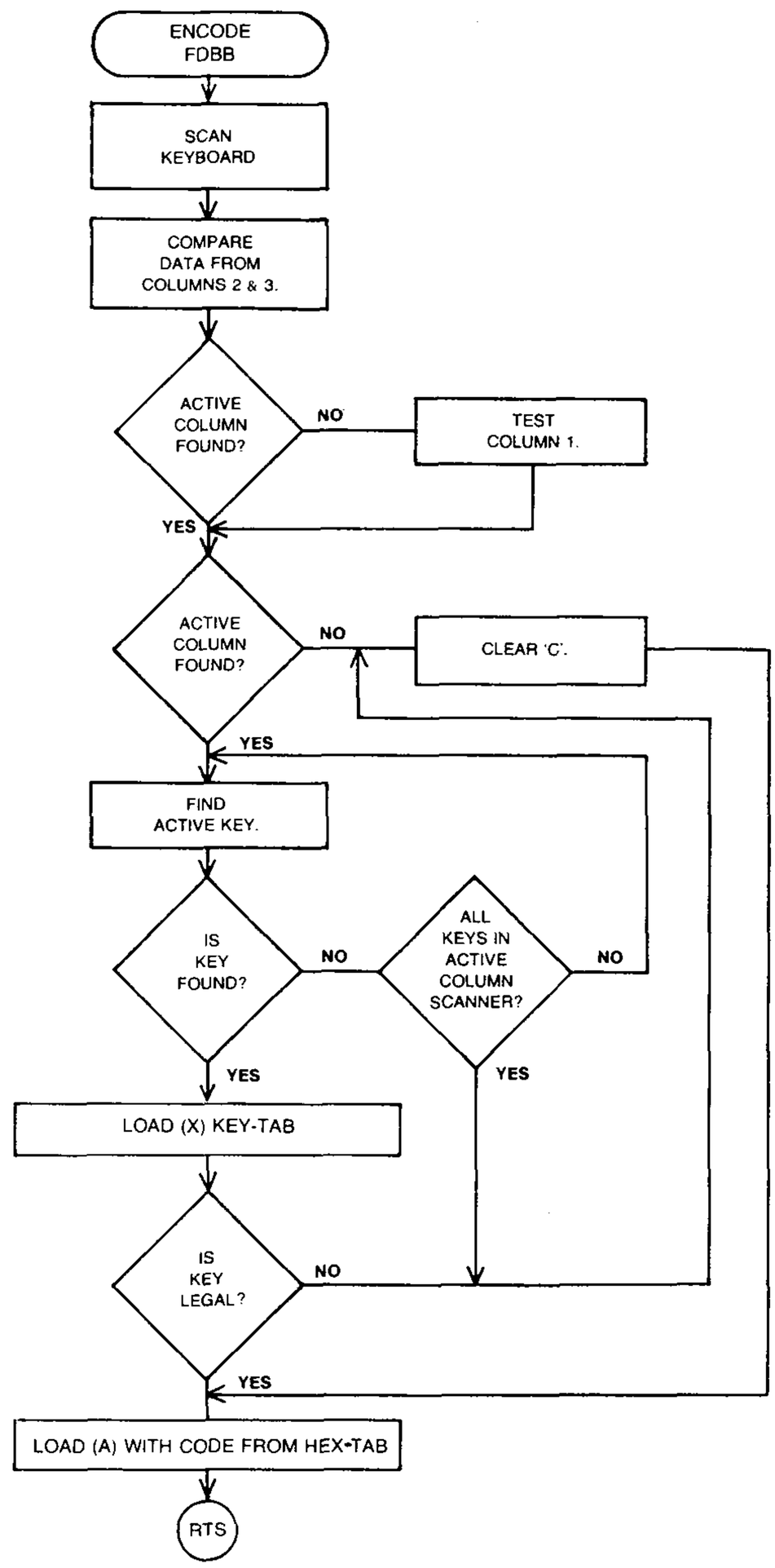
ENCODE Routine

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

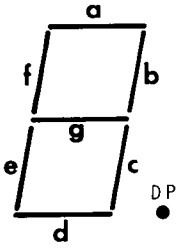


OUTCH Routine

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.

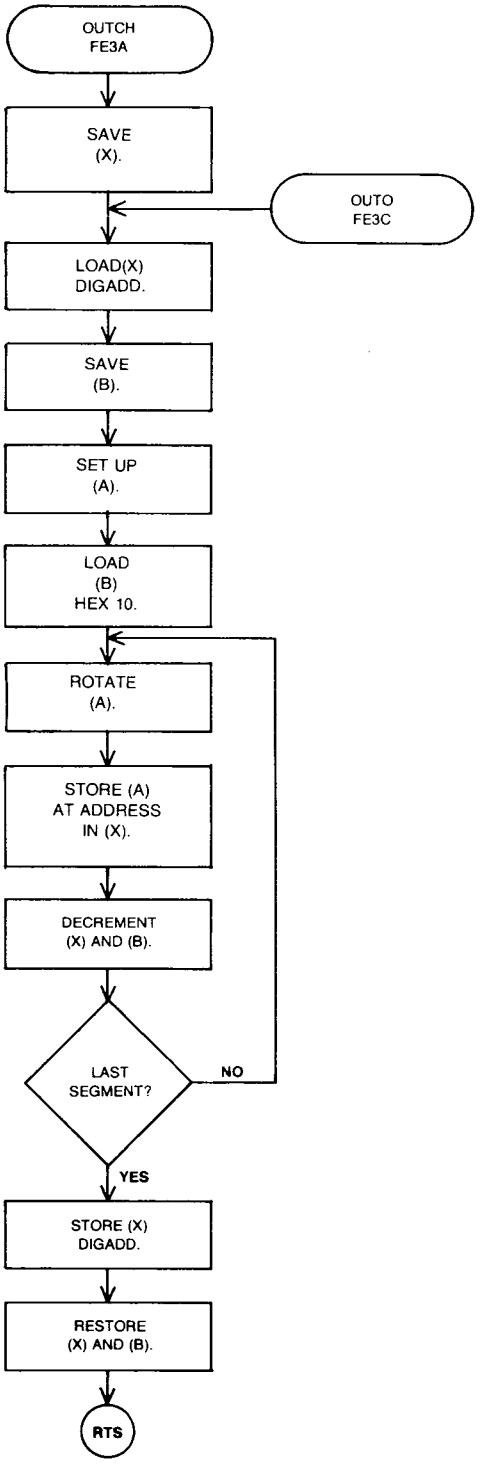
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SEGMENT	DP	a	b	c	d	e	f	g
HEX C 4E	0	1	0	0	1	1	1	0
LTR c OD	0	0	0	0	1	1	0	1



OUTCH ROUTINE

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.



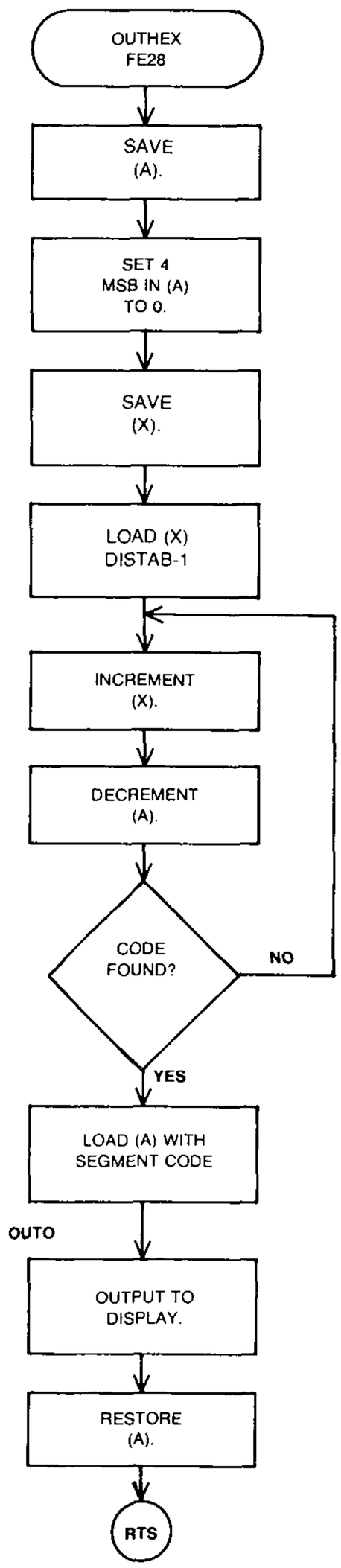
OUTHEX Routine

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.



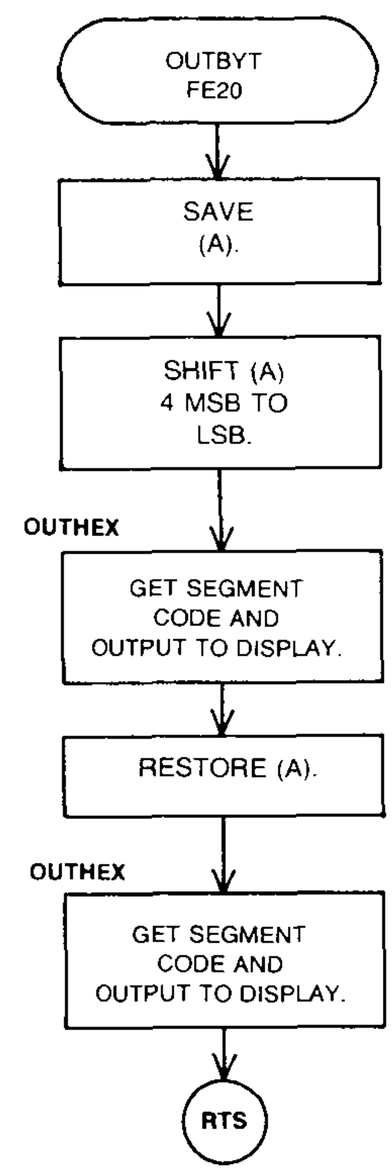
OUTBYT Routine

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.

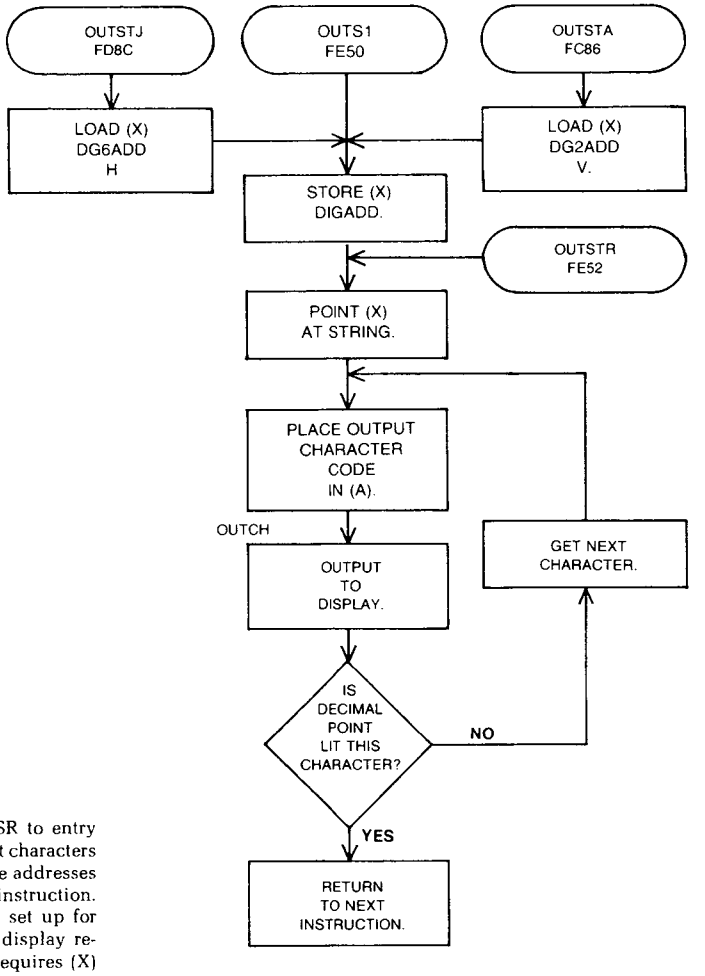


OUTST1 Routine

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

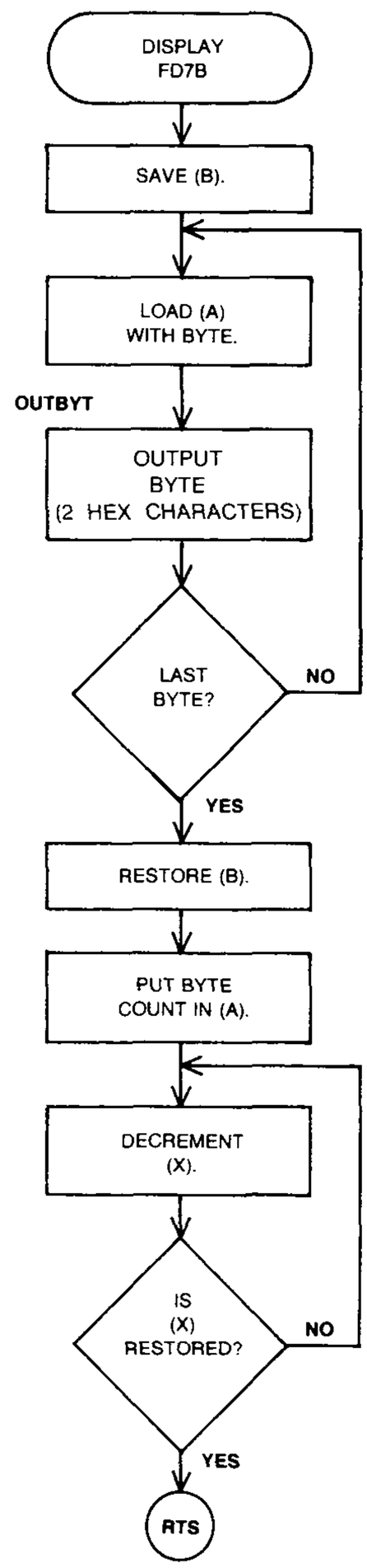
DSPLAY is called to output two or three bytes to the displays. The number of bytes to output is indicated by the contents of accumulator B. This routine could be called to output one byte, although OUTBYT would normally be called for this purpose.

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.
DIGADD contains address of digit.

EXIT: (X) and (B) unchanged.
Address of next digit to right contained in DIGADD.



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 accumualtor 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.

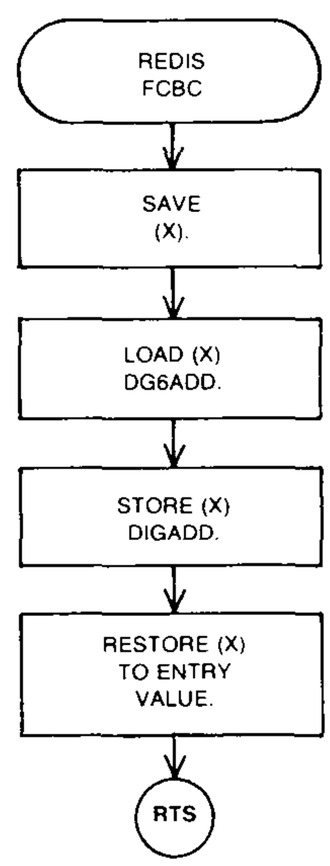
REDIS Routine

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

REDIS ROUTINE

ENTRY: None.

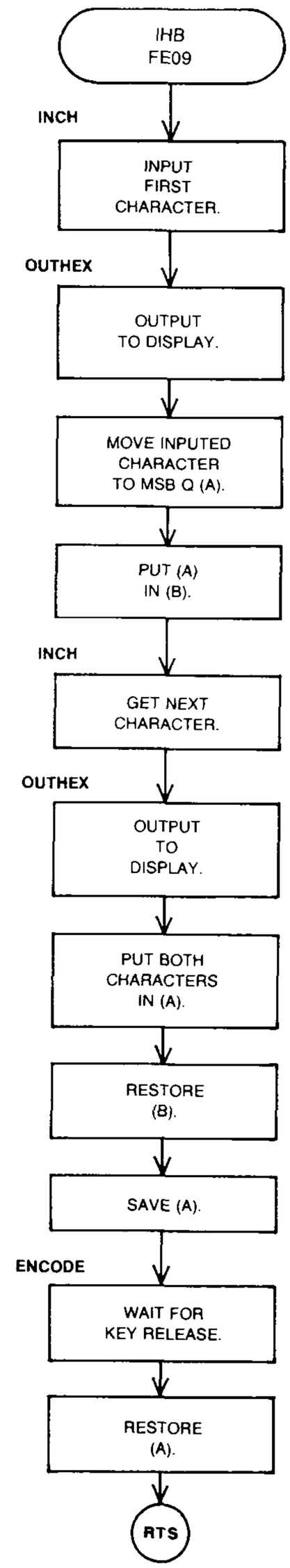
EXIT: DIGADD contains address of left-most digit.



IHB ROUTINE

ENTRY: None.

EXIT: ACCA contains byte value.
Digit address for next digit to right contained in DIGADD.



Monitor Listings

Your Trainer is controlled by U12, the "read only memory" (ROM). The following is a listing of the program stored in this IC.

Tables at the end of the listing show labels used in the program, keyboard and display addresses, segment codes for characters displayed by the program, and addresses in RAM that are reserved for use by the monitor program.

```
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,LTRF,LTRU,0,LTRU,LTRF+$80
FC0C  CE 00 CB              LDX     #USRSTK
FC0F  DF F2                 STX     USERS             SET UP FOR USER
FC11  86 FF                 LDA A   #$FF
FC13  C6 08                 LDA B   #2*NBR
FC15  36          RESE1     PSH A
FC16  5A                    DEC B
FC17  26 FC                 BNE     RESE1


                  **        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*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                          RETURN ONTO STACK
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   #$F
FC2B  27 F4                 BEQ     MAIN1                ILLEGAL NOW
FC2D  81 0B                 CMP A   #$B
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                    DEC A
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  DE EC                 LDX     T0                   RESTORE X
FC43  96 EE                 LDA A   T1
FC45  39          ZERO      RTS                          JUMP TO HANDLER
```

```
                  **        BKSET - ENTRY FOR BREAKPOINT TABLE
                  *
                  *         ENTRY:  NONE
                  *         EXIT    (B) = 2
                  *                 (X) = ADDRESS IN TABLE
                  *         USES: ALL,TO,11

FC46   CE 00 F2   BKSET     LDX     #BKTBL-2
FC49   86 FF                LDA A   #$FF
FC43   C6 04                LDA B   #ODR                FIND SPOT IN TABLE
FC4D   08         BKSE1     INX
FC4E   08                   INX
FC4F   A1 00                CMP A   0,X
FC51   26 04                BNE     BKSE2
FC53   A1 01                CMP A   1,X
FC55   27 0E                REC     BKSE3               EMPTY SPOT
FC57   5A         BKSE2     DEC B
FC58   26 F3                BNE     BKSE1               STILL HOPE

                  *         FULL UP

FC5A   BD FD 8D             JSR     OUTSTO
FC5D   00 47 3E 0E          FCB     0,LTRF,LTRU,LTRL,LTRL,$A0
FC63   4C                   INC A
FC64   39                   RTS

FC65   DF EE      BKSE3     STX     T1
FC67   8D 1D                BSR     OUTSTA
FC69   1F 85                FCC     LIBB,LTRR+$80
FC6B   8D 08                BSR     DOPM1
FC6D   4C                   INC A       (A) = 1
FC6E   39                   RTS
```

```
                    **      DOPMT - ACCEPT ADDRESS VALUE WITH 'DO' PROMPT
                    *
                    *       ENTRY:  (X) = ADDRESS TO STORE INPUTTED VALUE
                    *       EXIT:   (B) = 2
                    *               (X) UNCHANGED
                    *       USES:   ALL,T0,T1

FC6F   DF EE        DOPMT   STX     T1
FC71   8D 13                BSR     OUTSTA          OUTPUT PROMPT 'DO'
FC73   3D 90                FCC     LTRD,LTR0+$80
FC75   8D 45        DOPM1   BSR     REDIS           RESET DISPLAY
FC77   DE EE                LDX     T1              RESTORE X
FC79   C6 02                LDA B   #2
FC7B   7E FD 25             JMP     PROMPT          INPUT NEW VALUE


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

FC7E   DF EE        ADDR    STX     T1
FC80   8D 04                BSR     OUTSTA
FC82   77 BD                FCC     HEXA,LTRD+$80
FC84   20 EF                BRA     DOPM1


                    **      OUTSTA - OUTPUT STRING FOR ADDRESS PROMPT
                    *

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


                    **      DO - RESET USER PC AND RESUME
                    *
                    *       ENTRY:  NONE
                    *       EXIT:   TO 'RESUME'
                    *       USES:   ALL

FC8C   DE F2        DO      LDX     USERS
FC8E   08                   INX
FC8F   08                   INX
FC90   08                   INX
FC91   08                   INX
FC92   08                   INX
FC93   08                   INX                     X TO USER PC
FC94   8D D9                BSR     DOPMT
```

```
                  **      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
                  *       USES  ALL,T0,T1

FC96  8D 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     RES1
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
FCAD  36                  PSH A
FCAE  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     RES3
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
                  *       USES:   T0

FCBC  DF EC       REDIS   STX     T0
FCBE  CE C1 6F            LDX     #DG6ADD
FCC1  DF F0               STX     DIGADD
FCC3  DE EC               LDX     T0
FCC5  39                  RTS
```

```
                    **      BADDR - BUILD ADDRESS
                    *
                    *       ENTRY: NONE
                    *       EXIT   (X) = ADDRESS

FCC6  CE 00 EE      BADDR   LDX     #T1
FCC9  8D B3                 BSR     ADDR
FCCB  DE EE                 LDX     T1
FCCD  39                    RTS


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

FCCE  30            BKPT    TSX
FCCF  9F F2                 STS     USERS
FCD1  A6 06                 LDA A   6,X
FCD3  26 02                 BNE     BKP1                DECREMENT PC ON USERS STACK
FCD5  6A 05                 DEC     5,X
FCD7  4A            BKP1    DEC A
FCD8  A7 06                 STA A   6,X
FCDA  E6 05                 LDA B   5,X
FCDC  D7 EC                 STA B   TO                  SAVE FOR COMPARE
FCDE  97 ED                 STA A   TO+1

                    **      NOW CLEAR BREAKPOINTS

FCE0  0C                    CLC                         'C' IS HIT FLAG
FCE1  8E 00 D9      BKP2    LDS     #BKTBL-3-NBR-NBR
FCE4  C6 04                 LDA B   #NBR
FCE6  32            BKP3    PUL A
FCE7  32                    PUL A                       OLD OP CODE INTO A
FCE8  30                    TSX
FCE9  EE 08                 LDX     2*NBR,X
FCEB  9C EC                 CPX     TO                  DO WE HAVE A HIT?
FCED  26 01                 BNE     BKP4                   NO WE DO NOT
FCEF  0D                    SEC                               YES WE DO - SET FLAG
FCF0                BKP4    EQU     *
FCF0  A7 00                 STA A   0,X                 FIX USER CODE
FCF2  5A                    DEC B
FCF3  26 F1                 BNE     BKP3
FCF5  24 AC                 BCC     RES2                BREAKPOINT NOT HIT
FCF7  DE EC                 LDX     TO                  (X) = USER PC

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

FCF9  8D C1         MEM     BSR     REDIS               RESET DISPLAY
FCFB  DF EE                 STX     T1
FCFD  CE 00 EE              LDX     #T1
FD00  C6 02                 LDA B   #2
FD02  8D 03                 BSR     MEM2                DISPLAY ADDRESS
FD04  EE 00                 LDX     0,X
FD06  5A                    DEC B
FD07  7E FD 7B      MEM2    JMP     DSPLAY              OUTPUT DATA


                    **      AUTO - AUTO LOAD OF MEMORY
                    *
                    *       ENTRY: NONE
                    *       EXIT:  NO EXIT POSSIBLE
                    *       USES:  ALL,TO,T1
```

```
FD0A  8D BA     AUTO    BSR     BADDR           BUILD ADDRESS
FD0C  8D EB     AUT1    BSR     MEM
FD0E  8D 0B             BSR     REPLAC
FD10  08                INX
FD11  20 F9             BRA     AUT1            NO EXIT


                ***     EXAM - EXAMINE MEMORY
                *
                *       ENTRY:  NONE
                *       EXIT:   (X) = ADDRESS
                *               (B) = 0
                *               (A) = 0
                *       USES:   ALL,T0,T1

FD13  8D B1     EXAM    BSR     BADDR           BUILD ADDRESS
FD15  09                DEX

                ***     FOWD - DISPLAY NEXT BYTE
                *
                *       ENTRY:  (X) = OLD ADDRESS
                *       EXIT:   (X) = (XOLD) +1
                *               (B) = 1
                *               (A) = 0
                *       USES:   ALL,T0

FD16  08        FOWD    INX
FD17  08                INX


                ***     BACK - DISPLAY PREVIOUS BYTE
                *
                *       ENTRY:  (X) = ADDRESS
                *       EXIT:   (X) = (XOLD) + 1
                *               (B) = 1
                *               (A) = 0
                *       USES:   ALL,T0
FD18  09        BACK    DEX
FD19  20 DE             BRA     MEM             DISPLAY ADDRESS AND DATA


                ***     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

FD1B  5D        REPLAC  TST B
FD1C  27 06             BEQ     REPL1           NO BYTES
FD1E  36                PSH A
FD1F  8D 22             BSR     BKSP            BACKSPACE DISPLAYS
FD21  8D 02             BSR     PROMPT
FD23  32                PUL A
FD24  39        REPL1   RTS


                ***     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  58                  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  8D 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                    BUMP POINTER
FD39  5A                  DEC B
FD3A  26 F7               BNE     PROM2          MORE TO GO
FD3C  33                  PUL B
FD3D  17                  TBA                    DUPLICATE
FD3E  09          PROM3   DEX                    FIX X
FD3F  4A                  DEC A
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  8B 20       BKSP1   ADD A   #$20           BACKSPACE TWO PLACES
FD48  5A                  DEC B
FD49  26 FB               BNE     BKSP1
FD4B  97 F1               STA A   DIGADD+1
FD4D  33                  PUL B
FD4E  39                  RTS

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

FD4F  8D 3B       REGX    BSR     OUTSTJ         PRINT 'REGX'
FD51  30 95               FCC     LTRI,LTRN+$80
FD53  20 16               BRA     REGX1

FD55  8D 35       REGA    BSR     OUTSTJ         PRINT 'ACCA'
FD57  77 0D 0D            FCC     HEXA,LTRC,LTRC,LTRA+$80
FD5B  20 10               BRA     REGA1

FD5D  8D 2D       REGB    BSR     OUTSTJ         PRINT 'ACCB'
FD5F  77 0D 0D            FCC     HEXA,LTRC,LTRC,LTRB+$80
FD63  20 09               BRA     REGB1

FD65  8D 25       REGP    BSR     OUTSTJ         PRINT 'PC'
FD67  67 8D               FCC     LTRP,LTRC+$80

FD69  4C                  INC A                  (A) = OFFSET INTO STACK
FD6A  4C                  INC A                  (B) = #BYTES THIS REGISTER
FD6B  5C          REGX1   INC B
FD6C  4C                  INC A
FD6D  4C          REGA1   INC A
FD6E  5C          REGB1   INC B
FD6F  8B 02               ADD A   #2
```

```
FD71  DE F2              LDX     USERS
FD73  08         REG1    INX                     POINT X TO REGISTER
FD74  4A                 DEC A
FD75  26 FC              BNE     REG1
FD77  8D 02              BSR     DSPLAY
FD79  4C                 INC A
FD7A  39                 RTS


                 **      DISPLAY - DISPLAY INDEXED BYTES
                 *
                 *       ENTRY: (X) = ADDRESS OF BYTES TO OUTPUT
                 *              (B) = NUMBER OF BYTES TO DISPLAY
                 *       EXIT: X,B UNCHANGED
                 *              (DIGADD) UPDATED
                 *       USES:  ALL, TO

FD7B  37         DSPLAY  PSH B
FD7C  A6 00      DIS1    LDA A   0,X             GET BYTE
FD7E  BD FE 20           JSR     OUTBYT          DISPLAY BYTE
FD81  08                 INX
FD82  5A                 DEC B
FD83  26 F7              BNE     DIS1
FD85  33                 PUL B
FD86  17                 TBA                     DUPLICATE BYTE COUNT
FD87  09         DIS2    DEX                     RESTORE X
FD88  4A                 DEC A
FD89  26 FC              BNE     DIS2
FD8B  39                 RTS


                 *       CLEAR B AND JUMP TO OUTSTR

FD8C  5F         OUTSTJ  CLR B
FD8D  CE C1 6F   OUTSTO  LDX     #DG6ADD
FD90  7E FE 50           JMP     OUTST1


                 **      CONDX - DISPLAY CONDITION CODES
                 *
                 *       ENTRY:  DIGADD INITIALIZED
                 *       EXIT:   (B) = 0
                 *       USES:   ALL,TO

FD93  BD FC BC   CONDX   JSR     REDIS           RESET DISPLAYS
FD96  DE F2              LDX     USERS
FD98  C6 20              LDA B   #$20
FD9A  4F         CONDO   CLR A
FD9B  E5 01              BIT B   1,X             MASK DESIRED BIT

FD9D  27 01              BEQ     COND1           IS A ZERO
FD9F  4C                 INC A                   IS A ONE
FDA0  BD FE 28   COND1   JSR     OUTHEX
FDA3  56                 ROR B
FDA4  26 F4              BNE     CONDO           MORE TO GO
FDA6  4C                 INC A
FDA7  39                 RTS


                 **      STKPTR - OUTPUT USER STACK POINTER
                 *
                 *       ENTRY: (DIGADD) INITIALIZED
                 *       EXIT:  (B) = 0
                 *       USES:   ALL, TO

FDA8             REGS    EQU     *
FDA8  8D E2      STKPTR  BSR     OUTSTJ
FDAA  5B E7              FCC     LTRS,LTRP+$80
FDAC  D6 F3              LDA B   USERS+1
FDAE  CB 07              ADD B   #7
```

```
FDB0  99 F2                ADC A   USERS            CLEAN UP FOR USER
FDB2  8D 6C                BSR     OUTBYT
FDB4  17                   TBA
FDB5  5F                   CLR B
FDB6  8D 68                BSR     OUTBYT
FDB8  86 01                LDA A   #1
FDBA  39                   RTS


                  **       ENCODE - SCAN AND ENCODE KEYBOARD
                  *
                  *        ENTRY:  NONE
                  *        EXIT:   (A) = HEX VALUE OF KEY PRESSED
                  *                'C' SET FOR VALID CONDITION
                  *        USES:   A,C,TO

FDBB  37          ENCODE   PSH B
FDBC  F6 C0 03             LDA B   COL1             GET KEYBOARD DATA
FDBF  B6 C0 06             LDA A   COL3
FDC2  48                   ASL A                    LEFT JUSTIFY DATA
FDC3  48                   ASL A
FDC4  48                   ASL A
FDC5  59                   ROL B
FDC6  48                   ASL A
FDC7  59                   ROL B
FDC8  48                   ASL A
FDC9  59                   ROL B
FDCA  37                   PSH B
FDCB  F6 C0 05             LDA B   COL2             GET LAST DATA
FDCE  C4 1F                AND B   #$1F             MASK ANY GARBAGE
FDD0  1B                   ABA
FDD1  33                   PUL B
FDD2  43                   COM A
FDD3  53                   COM B

                  *        (BA) IS NOW KEYBOARD PATTERN

FDD4  DF EC                STX     TO
FDD6  CE FF A5             LDX     #HEXTAB-1        TABLE OF POSSIBLE OUTPUTS
FDD9  11                   CBA                      FIND ACTIVE ACCUMULATOR
FDDA  27 11                BEQ     ENC3             ILLEGAL OR NO KEY
FDDC  24 06                BCC     ENC1             A ACTIVE
FDDE  36                   PSH A                    B ACTIVE
FDDF  17                   TBA                          INTERCHANGE B,A
FDE0  33                   PUL B
FDE1  CE FF AD             LDX     #HEXTAB+7
FDE4  5D          ENC1     TST B                    B SHOULD BE ZERO
FDE5  26 06                BNE     ENC3             ILLEGAL
FDE7  08          ENC2     INX                      SCAN FOR ACTIVE BIT
FDE8  48                   ASL A
FDE9  22 FC                BHI     ENC2             NOT ACTIVE BIT
FDEB  27 01                BEQ     ENC4             LEGAL CHARACTER
FDED  0C          ENC3     CLC                      ILLEGAL RETURNS 'C' CLEAR
FDEE  A6 00       ENC4     LDA A   0,X              GET HEX FROM TABLE
FDF0  DE EC                LDX     TO
FDF2  33                   PUL B                    CLEAN UP
FDF3  39                   RTS                        AND RETURN


                  **       INCH - INPUT CHARACTER FROM KEYBOARD
                  *
                  *          '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,TO

FDF4  37          INCH     PSH B
FDF5  C6 7F       INC1     LDA B   #TIME            VIOLATION COUNT
```

```
FDF7  8D C2     INC2    BSR     ENCODE          WAIT FOR ILLEGAL INTERVAL
FDF9  25 FA             BCS     INC1            STILL LEGAL
FDFB  5A                DEC B
FDFC  26 F9             BNE     INC2            NOT A FELONY

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

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

                **      IHB - INPUT HEX BYTE AND DISPLAY ON LEDS
                *
                *       ENTRY: NONE
                *       EXIT:  (A) = BYTE VALUE
                *              (DIGADD) UPDATED
                *       USES:  A,TO,C

FE09  8D E9     IHB     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     IHB1    BSR     ENCODE          WAIT FOR KEY RELEASE
FE1C  25 FC             BCS     IHB1
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,TO

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,TO

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     OUTO            ALTERNATE ENTRY FOR 'OUTCH'
```

```
FE38  32                  PUL A
FE39  39                  RTS


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

FE3A  DF EC       OUTCH   STX     TO
FE3C  DE F0       OUTO    LDX     DIGADD          **ALTERNATE ENTRY** FROM 'OUTHEX
FE3E  37                  PSH B
FE3F  49                  ROL A
FE40  49                  ROL A                   PRE-ROTATE A
FE41  C6 10               LDA B   #$10            TO GET TO NEXT DIGIT
FE43  49          OUT1    ROL A                   HERE WE MAKE TWO PASSES AT
FE44  A7 00               STA A   0,X                LIGHTING DIGITS---
FE46  09                  DEX                           KING'S X ON FIRST PASS!
FE47  5A                  DEC B
FE48  26 F9               BNE     OUT1
FE4A  DF F0               STX     DIGADD          UPDATE 'DIGADD'
FE4C  DE EC               LDX     TO              RESTORES X
FE4E  33                  PUL B
FE4F  39                  RTS


                  **      OUTSTR--OUTPUT IMBEDDED CHARACTER STRING
                  *          CALLING CONVENTION:
                  *               JSR     OUTSTR
                  *               FIRST CHARACTER
                  *                    *
                  *                    *
                  *               LAST CHARACTER (HAS D.P. LIT)
                  *               NEXT INSTRUCTION
                  *
                  *       ENTRY:  NONE
                  *       EXIT:   TO 'NEXT INSTRUCTION'
                  *               (A) = 0
                  *       USES:   A,X,TO

FE50  DF F0       OUTST1  STX     DIGADD          **ALTERNATE ENTRY** SETS UP DIGA
FE52  30          OUTSTR  TSX                     POINT 'X' AT STRING
FE53  EE 00               LDX     0,X
FE55  31                  INS
FE56  31                  INS
FE57  A6 00       OUTST3  LDA A   0,X             GET CHARACTER
FE59  8D DF               BSR     OUTCH           OUTPUT IT TO DISPLAYS
FE5B  08                  INX
FE5C  4D                  TST A                   LAST CHARACTER IS NEGATIVE
FE5D  2A F8               BPL     OUTST3
FE5F  4F                  CLR A
FE60  6E 00               JMP     0,X             RETURN TO 'NEXT INST.'

                  **      STEP - STEP USER CODE
                  *
                  *       ENTRY:  NONE
                  *       EXIT:   (B) = 1
                  *               (X) = USER P.C.
                  *               (A) = 0
                  *       USES:   ALL,TO,T1
FE62  8D 07       STEP    BSR     SSTEP           STEP USER CODE
FE64  DE F2               LDX     USERS           DISPLAY INSTRUCTION
FE66  EE 06               LDX     6,X
FE68  7E FC F9            JMP     MEM


                  **      SSTEP - PERFORM SINGLE STEP.
                  *
```

```
FE6B  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 ONTO 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  86 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  81 30              CMP A    #$30          CHECK FOR BRANCH
FE99  24 04              BCC      BYT3
FE9B  81 20              CMP A    #$20
FE9D  24 14              BCC      BYT5          IT IS A BRANCH
FE9F  81 60      BYT3    CMP A    #$60
FEA1  25 11              BCS      BYT6          IT IS ONE BYTE
FEA3  81 8D              CMP A    #$8D
FEA5  27 0C              BEQ      BYT5          IT IS BSR
FEA7  84 BD              AND A    #$BD
FEA9  81 8C              CMP A    #$8C
FEAB  27 04              BEQ      BYT4          IS X OR SP IMMEDIATE
FEAD  84 30              AND A    #$30          CHECK FOR THREE BYTES
FEAF  81 30              CMP A    #$30
FEB1  C2 FF      BYT4    SBC B    #$FF
FEB3  5C         BYT5    INC B
FEB4  5C         BYT6    INC B
FEB5  27 70      BYT7    BEQ      BSTFD
FEB7  30                 TSX
FEB8  25 02              BCS      STEP1
FEBA  E7 01              STA B    1,X           BRANCH OFFSET TO 2
FEBC  86 01      STEP1   LDA A    #1
FEBE  C1 02              CMP B    #2
FEC0  2E 06              BGT      STEP3
FEC2  27 02              BEQ      STEP2         TWO BYTES
FEC4  A7 01              STA A    1,X           FOR ONE BYTERS
FEC6  A7 02      STEP2   STA A    2,X           NOT FOR THREE BYTERS
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
FECD  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

FED1  DE F2      SRCHOP  LDX      USERS
FED3  A7 06              STA A    6,X
FED5  E7 07              STA B    7,X           UPDATE PC ON USER STACK
FED7  C6 06              LDA B    #6
FED9  32                 PUL A
FEDA  36                 PSH A                  GET COPY OF OPCODE
```

```
FEDB  84 CF                AND A   #$CF            IS THIS A SUBROUTINE CALL?
FEDD  81 8D                CMP A   #$8D
FEDF  32                   PUL A
FEE0  27 48                BEQ     BSRH
FEE2  81 6E                CMP A   #$6E
FEE4  27 5B                BEQ     JPXH            IT IS INDEXED JUMP
FEE6  81 7E                CMP A   #$7E
FEE8  27 5E                BEQ     JMPH            IT IS EXTENDED JUMP
FEEA  81 39                CMP A   #$39
FEEC  27 62                BEQ     RTSH            IT IS RTS
FEEE  81 3B                CMP A   #$3B
FEF0  27 6C                BEQ     RTIH            IT IS RTI
FEF2  81 3F                CMP A   #$3F
FEF4  27 6E                BEQ     SWIH            IT IS SWI
FEF6  AF 06                STS     6,X             AIM USER PC AT SCRATCH AREA
FEF8  36                   PSH A                   REPLACE OPCODE
FEF9  CE FF 05             LDX     #SSRET

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

FEFC  86 7E       SWIVE1   LDA A   #$7E            JUMP OP CODE
FEFE  97 F4                STA A   SYSSWI
FF00  DF F5                STX     SYSSWI+1
FF02  9E F2                LDS     USERS
FF04  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

FF05  30          SSRET    TSX                     GET SWI HIT LOCATION INTO X
FF06  EE 05                LDX     5,X
FF08  08                   INX
FF09  4F                   CLR A
FF0A  5F                   CLR B
FF0B  9C EE                CPX     TEMP
FF0D  26 0C                BNE     BCHNTK

                  *        ADD THE BRANCH OFFSET TO THE USER PC

FF0F  09                   DEX                     X WILL NOW POINT AT USERPC
FF10  EE 00                LDX     0,X             SAVED VALUE OF PC INTO X
FF12  09                   DEX                     PREPARE TO FETCH BRANCH OFFSET
FF13  E6 00                LDA B   0,X
FF15  2A 01                BPL     PLUS
FF17  43                   COM A                   A IS SIGN EXTENSION OF B
FF18  30          PLUS     TSX                     LO COST WAY TO POINT TO USERPC
FF19  EE 05                LDX     5,X
FF1B  EB 01       BCHNTK   ADD B   1,X             ADD BRANCH OFFSET OR ZERO TO PC
FF1D  A9 00                ADC A   0,X
FF1F  30                   TSX                     PLACE NEW USERPC ONTO STACK
FF20  A7 05                STA A   5,X
FF22  E7 06                STA B   6,X
FF24  09                   DEX                     NOW X AND SP ARE EQUAL
FF25  DF F2       STOX     STX     USERS
FF27  9E EE       BSTRD    LDS     TEMP
FF29  39                   RTS                     RETURN TO CALLING ROUTINE


                  **       SPECIAL HANDLERS


                  **       BSR HANDLER

FF2A  81 8D       BSRH     CMP A   #$8D            IS IT BSR
FF2C  26 02                BNE     JSRH
FF2E  86 5F                LDA A   #$5F            THIS CONVERTS BSR'S TO BRA'S
```

```
                  **      JSR HANDLER

FF30  80 3F       JSRH    SUB A   #$3F              JSR'S TO JUMPS
FF32  36                  PSH A                   CORRECTED OPCODE ONTO STACK
FF33  09                  DEX
FF34  09                  DEX
FF35  DF F2               STX     USERS
FF37  A6 03       JSRH1   LDA A   3,X
FF39  A7 01               STA A   1,X             MOVE USER REGISTERS
FF3B  08                  INX
FF3C  5A                  DEC B
FF3D  2A F8               BPL     JSRH1
FF3F  20 90               BRA     SRCHOP          NOW EXECUTE JUMP INSTRUCT

                  **      JPXH - 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             CPX#: ONE BYTE BRA NEWPC

                  **      JMP HANDLER

FF48  32          JMPH    PUL A
FF49  33                  PUL B
FF4A  A7 06       NEWPC   STA A   6,X
FF4C  E7 07               STA B   7,X
FF4E  20 D5               BRA     STOX            RETURN TO CALLER

                  **      RTS HANDLER

FF50  08          RTSH    INX
FF51  08                  INX
FF52  DF F2               STX     USERS           NET PULL OF TWO BYTES
FF54  A6 03       RTS1    LDA A   3,X             MOVE FIVE BYTES
FF56  A7 05               STA A   5,X
FF58  09                  DEX
FF59  5A                  DEC B
FF5A  2E F8               BGT     RTS1
FF5C  20 C9               BRA     BSTRD

                  **      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      SET INTERUPT MASK
FF6E  A7 01               STA A   1,X
FF70  C6 FA               LDA B   #-USWI/256*256+USWI    USWI LO ORDER
FF72  86 00               LDA A   #USWI/256
FF74  20 D4               BRA     NEWPC           PATCH IN UIRQ
```

```
                    **      OPTAB - LEGAL OP-CODE LOOKUP TABLE

FF76  9C 00 3C  OPTAB   FDB     $9C00,$3CAF,$4000,$00AC,$6412,$6412,$6410,$6410
FF86  11 01 10          FDB     $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  CMDTAB  FDB     ZERO,REGA,REGB,REGP,REGX,CONDX,REGS,RESUME,STEP
FFC8  FC 46 FD          FDB     BKSET,AUTO,BACK,REPLAC,DO,EXAM,FOWD



FFF8                    ORG     $FFF8

                    **      INTERRUPT VECTORS.

FFF8  00 F7             FDB     UIRQ            USER IRQ HANDLER
FFFA  00 F4             FDB     SYSSWI          SYSTEM SWI HANDLER
FFFC  00 FD             FDB     UNMI            USER NMI HANDLER
FFFE  FC 00             FDB     RESET

0000                    END



STATEMENTS =970

FREE BYTES =24298

NO ERRORS DETECTED
```

SYMBOLIC REFERENCE TABLE.

ADDR	FC7E	DIGADD	00F0	MEM	FCF9	REG1	FD73
AUTO	FD0A	DISTAB	FF96	MEM2	FD07	REPLAC	FD1B
AUT1	FD0C	DIS1	FD7C	MONSTK	00E3	REPL1	FD24
BACK	FD18	DIS2	FD87	NEWPC	FF4A	RESET	FC00
BADDR	FCC6	DO	FC8C	OPTAB	FF76	RESE1	FC15
BCHNTK	FF1B	DOPMT	FC6F	OUTBYT	FE20	RESUME	FC96
BKPT	FCCE	DOPMI	FC75	OUTCH	FE3A	RES1	FC9B
BKP1	FCD7	DSPLAY	FD7B	OUTHEX	FE28	RES2	FCA3
BKP2	FCE1	ENCODE	FDBB	OUTH1	FE30	RES3	FCA8
BKP3	FCE6	ENC1	FDE4	OUTSTA	FC86	RTIH	FF5E
BKP4	FCF0	ENC2	FDE7	OUTSTJ	FD8C	RTSH	FF50
BKSET	FC46	ENC3	FDED	OUTSTR	FE52	RTS1	FF54
BKSE1	FC4D	ENC4	FDEE	OUTSTO	FD8D	SRCHOP	FED1
BKSE2	FC57	EXAM	FD13	OUTST1	FE50	SSRET	FF05
BKSE3	FC65	FOWD	FD16	OUTST3	FE57	SSTEP	FE6B
BKSP	FD43	HEXTAB	FFA6	OUTO	FE3C	STEP	FE62
BKSP1	FD46	IHB	FE09	OUT1	FE43	STEP1	FEBC
BSRH	FF2A	IHB1	FE1A	PLUS	FF18	STEP2	FEC6
BSTRD	FF27	INCH	FDF4	PROMPT	FD25	STEP3	FEC8
BYTCNT	FE83	INC1	FDF5	PROM1	FD29	STKPTR	FDA8
BYT1	FE88	INC2	FDF7	PROM2	FD33	STOX	FF25
BYT2	FE8F	INC3	FDFE	PROM3	FD3E	SWIH	FF64
BYT3	FE9F	INC4	FE00	REDIS	FCBC	SWIVE1	FEFC
BYT4	FEB1	JMPH	FF48	REGA	FD55	SYSSWI	00F4
BYT5	FEB3	JPXH	FF41	REGA1	FD6D	ZERO	FC45
BYT6	FEB4	JSRH	FF30	REGB	FD5D		
BYT7	FEB5	JSRH1	FE37	REGB1	FD6E		
CMDTAB	FFB6	MAIN	FC19	REGP	FD65		
CONDX	FD93	MAIN1	FC21	REGS	FDA8		
CONDO	FD9A	MAIN2	FC31	REGX	FD4F		
COND1	FDA0	MAIN3	FC36	REGX1	FD6B		

```
        ASSEMBLY CONSTANT TABLE

        KEYBOARD LOCATIONS

COL1    EQU     $0003           RIGHTMOST COLUMN
COL2    EQU     $0005
COL3    EQU     $0006           LEFTMOST COLUMN

        MISC. CONSTANTS

TIME    EQU     32

NBR     EQU     4               NUMBER BREAKPOINTS
```

```
*
*       DISPLAY ADDRESSES
DG6ADD  EQU     $C16F           LEFTMOST DIGIT
DG5ADD  EQU     $C15F
DG4ADD  EQU     $C14F
DG3ADD  EQU     $C13F
GD2ADD  EQU     $C12F
DG1ADD  EQU     $C11F           RIGHTMOST DIGIT
```

```
*       DISPLAYED CHARACTER SEGMENT CODES
*
HEX0    EQU     $7E     HEX8    EQU     $7F     LTRA    EQU     $7D     LTRH    EQU     $37
HEX1    EQU     $30     HEX9    EQU     $7B     LTRB    EQU     $1F     LTRD    EQU     $3D
HEX2    EQU     $6D     HEXA    EQU     $77     LTRC    EQU     $0D     LTRG    EQU     $5E
HEX3    EQU     $79     HEXB    EQU     $1F     LTRF    EQU     $47     LTRO    EQU     $1D
HEX4    EQU     $33     HEXC    EQU     $4E     LTRN    EQU     $15     LTRR    EQU     $05
HEX5    EQU     $5B     HEXD    EQU     $3D     LTRI    EQU     $30     LTRU    EQU     $3E
HEX6    EQU     $5F     HEXE    EQU     $4F     LTRP    EQU     $67     LTRY    EQU     $3B
HEX7    EQU     $70     HEXF    EQU     $47     LTRL    EQU     $0E     DASH    EQU     $08
```

Reserved Memory Bytes in RAM

```
00CB            USRSTK          EQU     *-6
00D1                            RMB     19
00E3            MONSTK          EQU     *-1
00E4            BKTBL           RMB     2*NBR
00EC            TO              RMB     2               TEMPORARY
00EE            TEMP            RMB     2               USED BY SINGLE STEPPER
00F0            DIGADD          RMB     2               DISPLAY POINTER
00F2            USERS           RMB     2               USER STACK POINTER
00EF            T1              EQU     TEMP
00F4            SYSSWI          RMB     3               SYSTEM SWI VECTOR
00F7            UIRQ            RMB     3               USER IRQ VECTOR
00FA            USWI            RMB     3               USER SWI VECTOR
00FD            UNMI            RMB     3               USER NMI VECTOR
```

Memory

Memory Map

The memory is organized as shown below.

Monitor ROM	FFFF FC00
Not usable	C1FF C170
Displays	C16F * C110
Not usable	C10F C00F
Keyboard	C00E * C003
Not usable	C002 C000
256 bytes of user RAM	01FF 0100
59 bytes RAM (reserved for monitor)	00FF 00C5
197 bytes of user RAM	00C4 0000

Memory Decoding

		A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
ROM U12	FFXX FCXX	1	1	1	1	1	1	X	X	X	X	X	X	X	X	X	X
RAM U14, U15	01XX	0	0	0	0	0	0	0	1	X	X	X	X	X	X	X	X
RAM U14, U15	00XX	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X
KEYBOARD	C0 - X	1	1	0	0	0	0	0	0	—	—	—	—	—	X	X	X
DISPLAYS	C1XX	1	1	0	0	0	0	0	1	—	X	X	X	—	X	X	X

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

* Not fully decoded.

Keyboard And Display Functioning Addresses

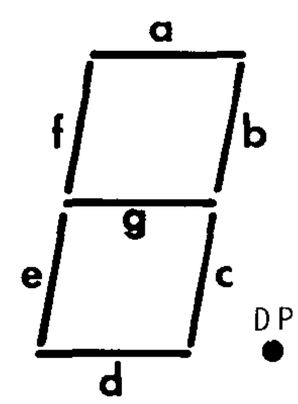
KEYBOARD

Keys	BINARY				HEX
	A_3	A_2	A_1	A_0	
3, 6, 9, C, F	—	0	1	1	3
2, 5, 8, B, E	—	1	0	1	5
0, 1, 4, 7, A, D	—	1	1	0	6

DISPLAY

LED	BINARY			HEX
	A_6	A_5	A_4	
H	1	1	0	6
I	1	0	1	5
N	1	0	0	4
Z	0	1	1	3
V	0	1	0	2
C	0	0	1	1

LED SEGMENTS



LED SEGMENT	BINARY				HEX
	A_3	A_2	A_1	A_0	
a	—	1	1	0	E or 6
b	—	1	0	1	D or 5
c	—	1	0	0	C or 4
d	—	0	1	1	B or 3
e	—	0	1	0	A or 2
f	—	0	0	1	9 or 1
g	—	0	0	0	8 or 0
DP	—	1	1	1	F or 7

— = DOES NOT CARE

Heathkit®

Instruction Set*

ACCUMULATOR AND MEMORY OPERATIONS	MNEMONIC	IMMED OP	IMMED ~	IMMED #	DIRECT OP	DIRECT ~	DIRECT #	INDEX OP	INDEX ~	INDEX #	EXTND OP	EXTND ~	EXTND #	INHER OP	INHER ~	INHER #	BOOLEAN/ARITHMETIC OPERATION (All register labels refer to contents)	COND. CODE REG. 5 H	4 I	3 N	2 Z	1 V	0 C
Add	ADDA	8B	2	2	9B	3	2	AB	5	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	5	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add with Carry	ADCA	89	2	2	99	3	2	A9	5	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	5	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	5	2	B4	4	3				A • M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	5	2	F4	4	3				B • M → B	•	•	↕	↕	R	•
Bit Test	BITA	85	2	2	95	3	2	A5	5	2	B5	4	3				A • M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	5	2	F5	4	3				B • M	•	•	↕	↕	R	•
Clear	CLR							6F	7	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	5	2	B1	4	3				A − M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	5	2	F1	4	3				B − M	•	•	↕	↕	↕	↕
Compare Acmltrs	CBA													11	2	1	A − B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	7	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	7	2	70	6	3				00 − M → M	•	•	↕	↕	①	②
	NEGA													40	2	1	00 − A → A	•	•	↕	↕	①	②
	NEGB													50	2	1	00 − B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts Binary Add. of BCD Characters into BCD Format	•	•	↕	↕	↕	③
Decrement	DEC							6A	7	2	7A	6	3				M − 1 → M	•	•	↕	↕	④	•
	DECA													4A	2	1	A − 1 → A	•	•	↕	↕	④	•
	DECB													5A	2	1	B − 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	5	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	5	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	7	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Acmltr	LDAA	86	2	2	96	3	2	A6	5	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	5	2	F6	4	3				M → B	•	•	↕	↕	R	•
Or, Inclusive	ORAA	8A	2	2	9A	3	2	AA	5	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	5	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	4	1	SP + 1 → SP, M_{SP} → A	•	•	•	•	•	•
	PULB													33	4	1	SP + 1 → SP, M_{SP} → B	•	•	•	•	•	•
Rotate Left	ROL							69	7	2	79	6	3				M	•	•	↕	↕	⑥	↕
	ROLA													49	2	1	A	•	•	↕	↕	⑥	↕
	ROLB													59	2	1	B	•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	7	2	76	6	3				M	•	•	↕	↕	⑥	↕
	RORA													46	2	1	A	•	•	↕	↕	⑥	↕
	RORB													56	2	1	B	•	•	↕	↕	⑥	↕
Shift Left, Arithmetic	ASL							68	7	2	78	6	3				M	•	•	↕	↕	⑥	↕
	ASLA													48	2	1	A	•	•	↕	↕	⑥	↕
	ASLB													58	2	1	B	•	•	↕	↕	⑥	↕
Shift Right, Arithmetic	ASR							67	7	2	77	6	3				M	•	•	↕	↕	⑥	↕
	ASRA													47	2	1	A	•	•	↕	↕	⑥	↕
	ASRB													57	2	1	B	•	•	↕	↕	⑥	↕
Shift Right, Logic.	LSR							64	7	2	74	6	3				M	•	•	R	↕	⑥	↕
	LSRA													44	2	1	A	•	•	R	↕	⑥	↕
	LSRB													54	2	1	B	•	•	R	↕	⑥	↕
Store Acmltr.	STAA				97	4	2	A7	6	2	B7	5	3				A → M	•	•	↕	↕	R	•
	STAB				D7	4	2	E7	6	2	F7	5	3				B → M	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	5	2	B0	4	3				A − M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	5	2	F0	4	3				B − M → B	•	•	↕	↕	↕	↕
Subract Acmltrs.	SBA													10	2	1	A − B → A	•	•	↕	↕	↕	↕
Subtr. with Carry	SBCA	82	2	2	92	3	2	A2	5	2	B2	4	3				A − M − C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	5	2	F2	4	3				B − M − C → B	•	•	↕	↕	↕	↕
Transfer Acmltrs	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test, Zero or Minus	TST							6D	7	2	7D	6	3				M − 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A − 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B − 00	•	•	↕	↕	R	R

INDEX REGISTER AND STACK		IMMED			DIRECT			INDEX			EXTND			INHER				5	4	3	2	1	0
POINTER OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#	BOOLEAN/ARITHMETIC OPERATION	H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2	BC	5	3				$(X_H/X_L) - (M/M+1)$	•	•	⑦	↕	⑧	•
Decrement Index Reg	DEX													09	4	1	$X - 1 \rightarrow X$	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	4	1	$SP - 1 \rightarrow SP$	•	•	•	•	•	•
Increment Index Reg	INX													08	4	1	$X + 1 \rightarrow X$	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	4	1	$SP + 1 \rightarrow SP$	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	6	2	FE	5	3				$M \rightarrow X_H, (M+1) \rightarrow X_L$	•	•	⑨	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	6	2	BE	5	3				$M \rightarrow SP_H, (M+1) \rightarrow SP_L$	•	•	⑨	↕	R	•
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3				$X_H \rightarrow M, X_L \rightarrow (M+1)$	•	•	⑨	↕	R	•
Store Stack Pntr	STS				9F	5	2	AF	7	2	BF	6	3				$SP_H \rightarrow M, SP_L \rightarrow (M+1)$	•	•	⑨	↕	R	•
Indx Reg → Stack Pntr	TXS													35	4	1	$X - 1 \rightarrow SP$	•	•	•	•	•	•
Stack Pntr → Indx Reg	TSX													30	4	1	$SP + 1 \rightarrow X$	•	•	•	•	•	•

JUMP AND BRANCH		RELATIVE			INDEX			EXTND			INHER				5	4	3	2	1	0
OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	BRANCH TEST	H	I	N	Z	V	C
Branch Always	BRA	20	4	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	4	2										C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	4	2										C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	4	2										Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	4	2										$N \oplus V = 0$	•	•	•	•	•	•
Branch If > Zero	BGT	2E	4	2										$Z + (N \oplus V) = 0$	•	•	•	•	•	•
Branch If Higher	BHI	22	4	2										$C + Z = 0$	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	4	2										$Z + (N \oplus V) = 1$	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	4	2										$C + Z = 1$	•	•	•	•	•	•
Branch If < Zero	BLT	2D	4	2										$N \oplus V = 1$	•	•	•	•	•	•
Branch If Minus	BMI	2B	4	2										N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	4	2										Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	4	2										V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	4	2										V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	4	2										N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	8	2										See Special Operations	•	•	•	•	•	•
Jump	JMP				6E	4	2	7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR				AD	8	2	BD	9	3					•	•	•	•	•	•
No Operation	NOP										01	2	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI										3B	10	1		⑩					
Return From Subroutine	RTS										39	5	1	See special Operations	•	•	•	•	•	•
Software Interrupt	SWI										3F	12	1		•	S	•	•	•	•
Wait for Interrupt	WAI										3E	9	1		•	⑪	•	•	•	•

CONDITIONS CODE REGISTER		INHER				5	4	3	2	1	0
OPERATIONS	MNEMONIC	OP	~	#	BOOLEAN OPERATION	H	I	N	Z	V	C
Clear Carry	CLC	0C	2	1	$0 \rightarrow C$	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	$0 \rightarrow I$	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	$0 \rightarrow V$	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	$1 \rightarrow C$	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	$1 \rightarrow I$	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	$1 \rightarrow V$	•	•	•	•	S	•
Acmltr A → CCR	TAP	06	2	1	$A \rightarrow CCR$	⑫					
CCR → Acmltr A	TPA	07	2	1	$CCR \rightarrow A$	•	•	•	•	•	•

CONDITION CODE REGISTER NOTES:

(Bit set if test is true and cleared otherwise)

① (Bit V) Test: Result = 10000000?

② (Bit C) Test: Result = 00000000?

③ (Bit C) Test: Decimal value of most significant BCD Character greater than nine? (Not cleared if previously set.)

④ (Bit V) Test: Operand = 10000000 prior to execution?

⑤ (Bit V) Test: Operand = 01111111 prior to execution?

⑥ (Bit V) Test: Set equal to result of $N \oplus C$ after shift has occurred.

⑦ (Bit N) Test: Sign bit of most significant (MS) byte of result = 1?

⑧ (Bit V) Test: 2's complement overflow from subtraction of LS bytes?

⑨ (Bit N) Test: Result less than zero? (Bit 15 = 1)

⑩ (All) Load Condition Code Register from Stack. (See Special Operations)

⑪ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.

⑫ (ALL) Set according to the contents of Accumulator A.

LEGEND:

OP	Operation Code (Hexadecimal);	00	Byte = Zero;
~	Number of MPU Cycles;	H	Half-carry from bit 3;
#	Number of Program Bytes;	I	Interrupt mask
+	Arithmetic Plus;	N	Negative (sign bit)
−	Arithmetic Minus;	Z	Zero (byte)
•	Boolean AND;	V	Overflow, 2's complement
M_{SP}	Contents of memory location pointed to be Stack Pointer;	C	Carry from bit 7
		R	Reset Always
		S	Set Always
✚	Boolean Inclusive OR;	↕	Test and set if true, cleared otherwise
⊕	Boolean Exclusive OR;	•	Not Affected
$\bar{M}$	Complement of M;	CCR	Condition Code Register
→	Transfer Into;	LS	Least Significant
0	Bit = Zero;	MS	Most Significant

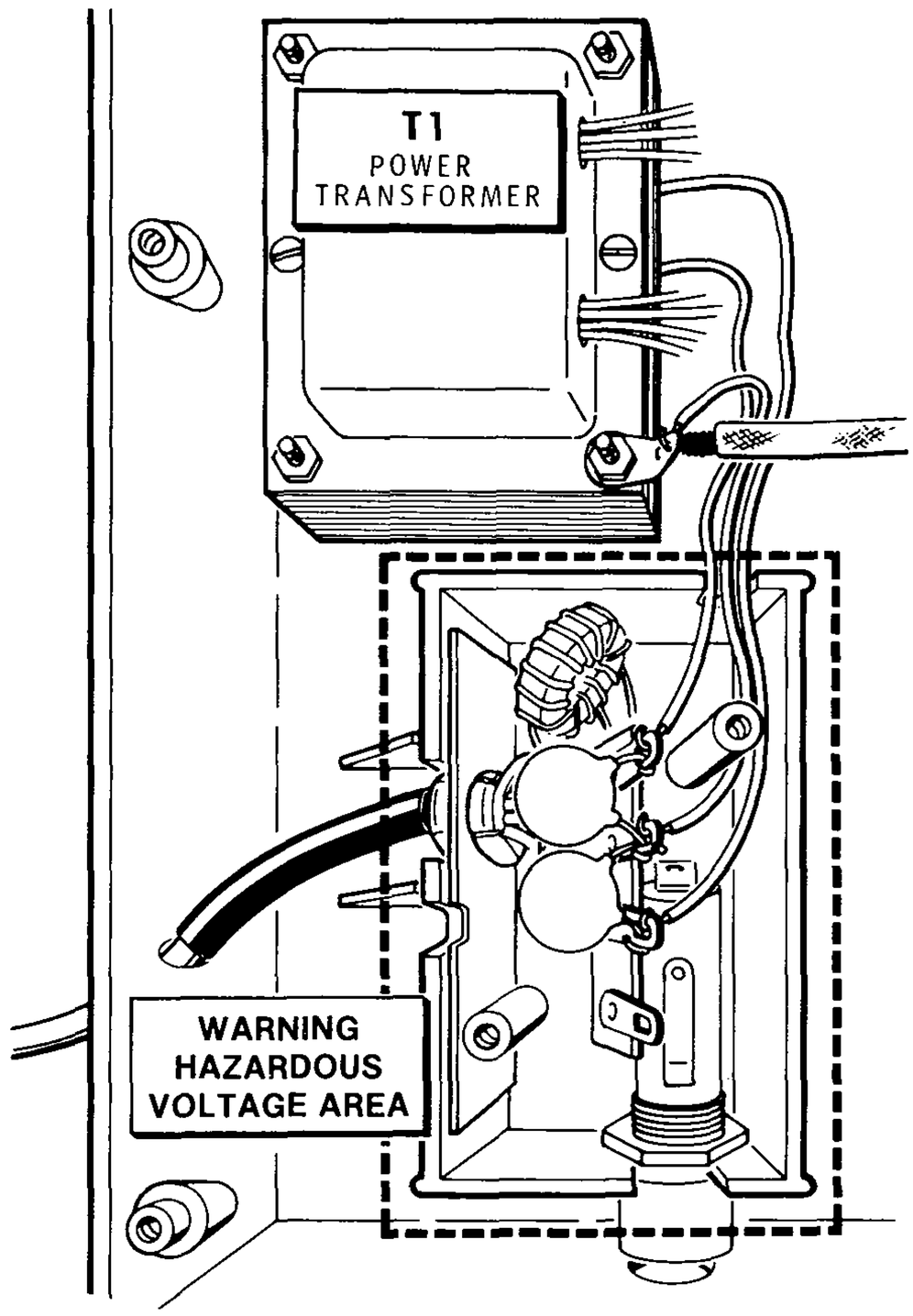
IN CASE OF DIFFICULTY

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: "General," "Troubleshooting Chart," and "Detailed Troubleshooting." Use the "General" part 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 trouble is still not located after you complete the "Visual Tests," check voltage readings against those shown in the "Schematic." NOTE: All voltage readings were taken with a high impedance voltmeter (10 MΩ or greater). Refer to "Circuit Board X-Ray Views" (Illustration Booklet, Page 14) for the physical location of parts.

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 Waranty is located inside the front cover.



PICTORIAL 9-1

General

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. Therfore, 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 U23 through U28 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. If a particular part is mentioned as a possible cause, check the part and other components connected to it to see that they are installed and/or wired correctly. Also check for solder bridge and poor connections in the surrounding area. It is also possible, on rear occasions, for a part to be faulty and require replacement.

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 in 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 de-soldering tool makes part removal easier. Check the foils around the component to be sure the foil is still intact.

4. Logic 0 refers to voltage levels less than .8 volt. Logic 1 refers to voltage levels greater than 2 volts.

5. When you apply +5 volts to LED's D9 through D16 in some of the following tests, the LED's may stay on when you remove the +5 volts. To turn the LED off, touch the test lead to ground.

POWER SUPPLIES

DIFFICULTY	POSSIBLE CAUSE
No +5 V, +12 V, and −12 V supplies. LED D3 not lit.	1. Fuse F1. 2. Transformer T1 primary wiring. 3. Line cord wiring.
No +5 V supply, in Standby or On position. LED D3 not lit.	1. Transformer T1 secondary wiring (green and green-yellow leads). 2.* Regulator U31. 3. short circuit on 5 V line.
No +5 V supply in On position.	1. Switch SW1. 2. Short circuit on main 5 V line.
No +12 V supply in On position.	1.* Regulator U29.
No −12 V supply in On position.	1.* Regulator U30.
Have +5 VS in Standby position. No + 5 V in On position.	1. Shorted main +5 V line. 2. Switch SW1.
Have +5 VS in Standby position. No +5 V to LED's in On position.	1. Open main +5 V line. 2. Switch SW1.

*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.

7-SEGMENT LED's

DIFFICULTY	POSSIBLE CAUSE
No LED's light when "Segment Test" is shorted.	1. +5 V not supplied to LED's.
All seven segments of one LED do not light when "Segment Test" is shorted.	1. +5 V not supplied to this LED.
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. 2. Jumper wire at terminal blocks A11/A12.
All LED's light dimly when "Segment Test" is shorted.	1. +5 V not supplied to IC's U23 through U28.
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 pushed to Standby.	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 LED D3 next to the Power switch being lit, you can troubleshoot your Trainer without using test equipment.

Push 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 does not seem to 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 bottom of the circuit board.

B. Unplug the line cord, remove U1 from its socket, and plug the line cord back in.

Remove the yellow wire from the +5 connector block and insert it into the GND connector block.

With the free end of the yellow wire, one at a time, touch the eight indicated pins of socket U1. 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 U1. Use a DC voltmeter to check the voltage between this wire and ground (GND). You should measure 5 volts. IF it does not, 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.

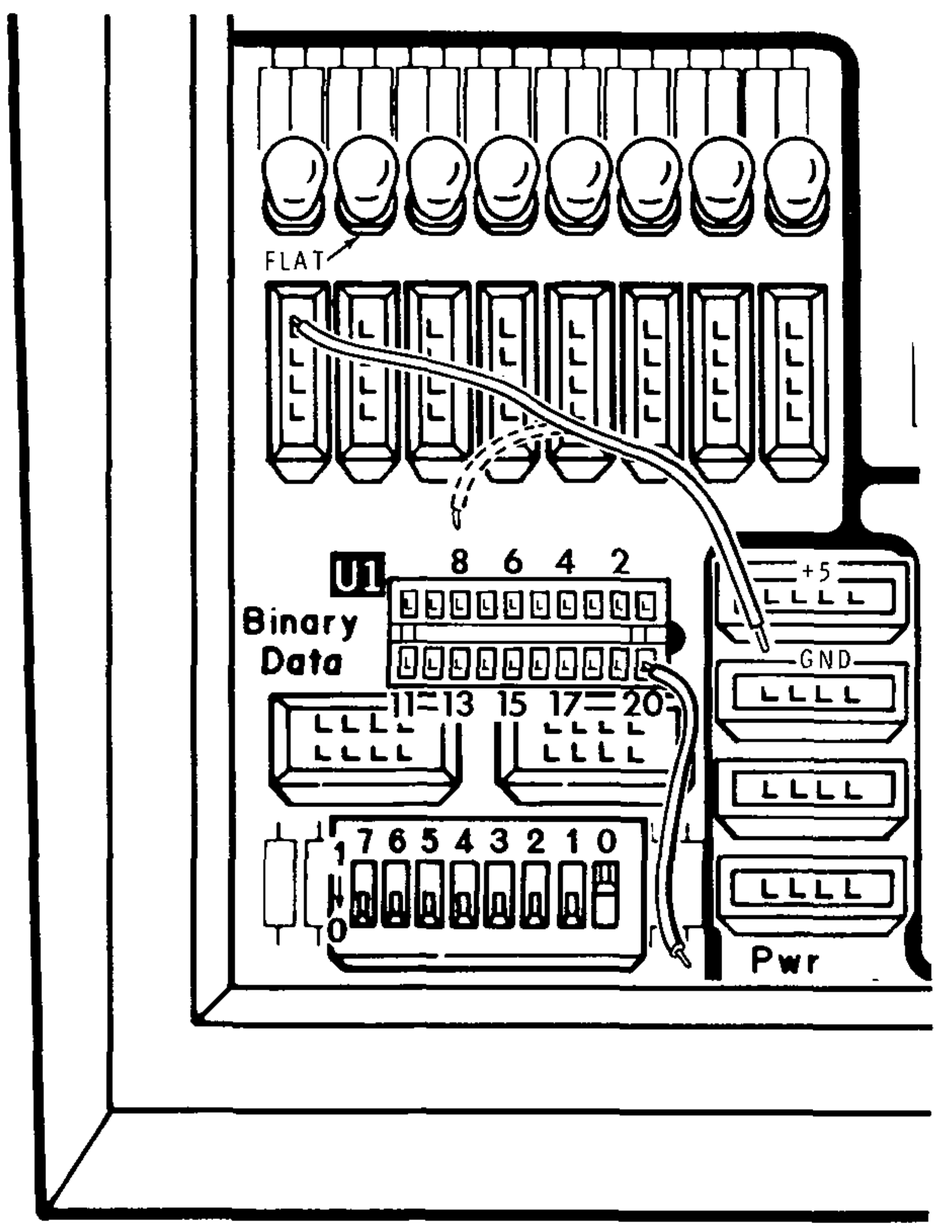
D. Unplug the line cord.

Be sure the pins of U1 are straight and then properly reinstall the IC in its socket.

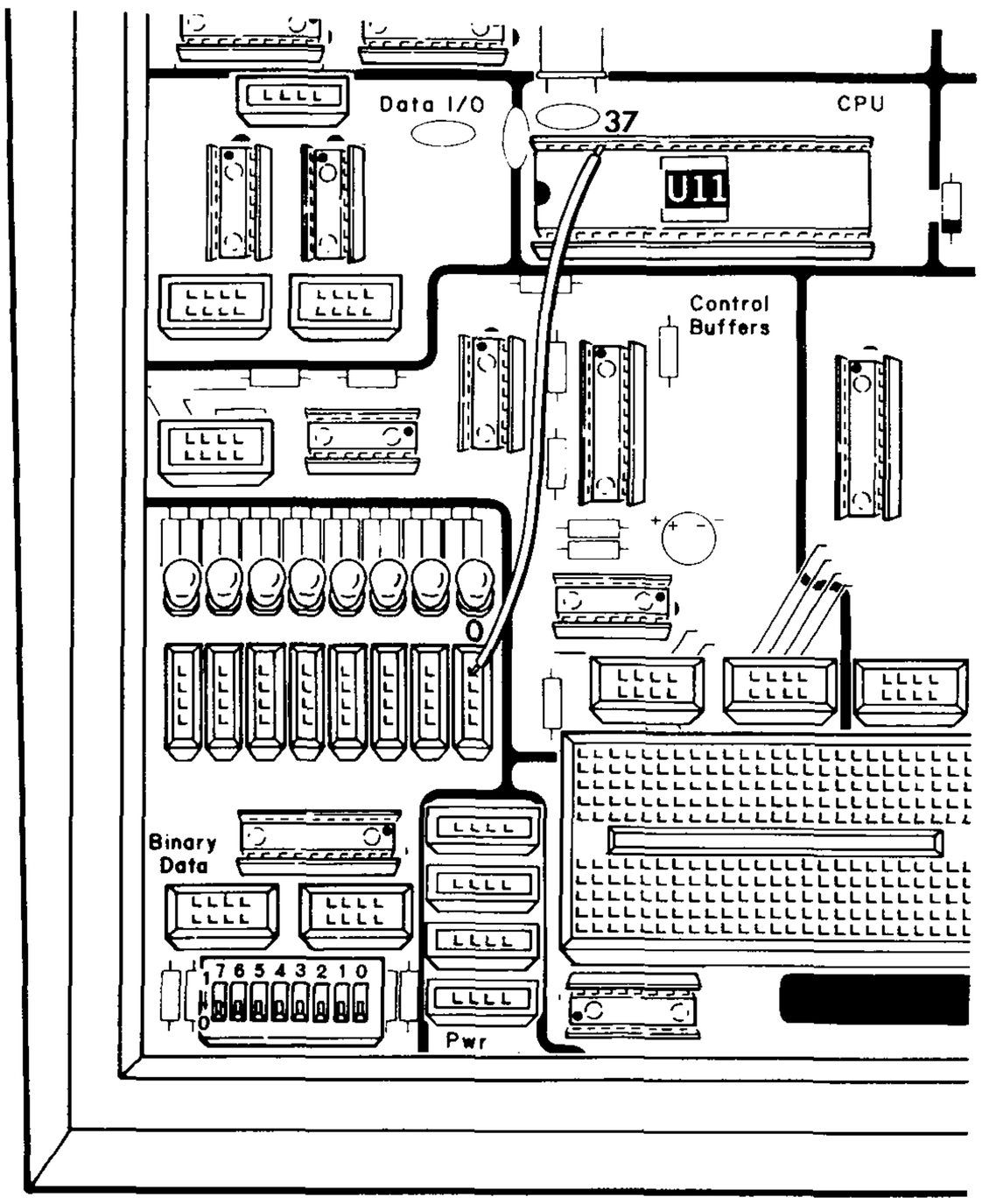
Reconnect the yellow wire to LED connector block 7 (or to the connector block of the LED that did not seem to work earlier) and the +5 connector block. The data LED should light. If it does not light, replace U1.

Clock

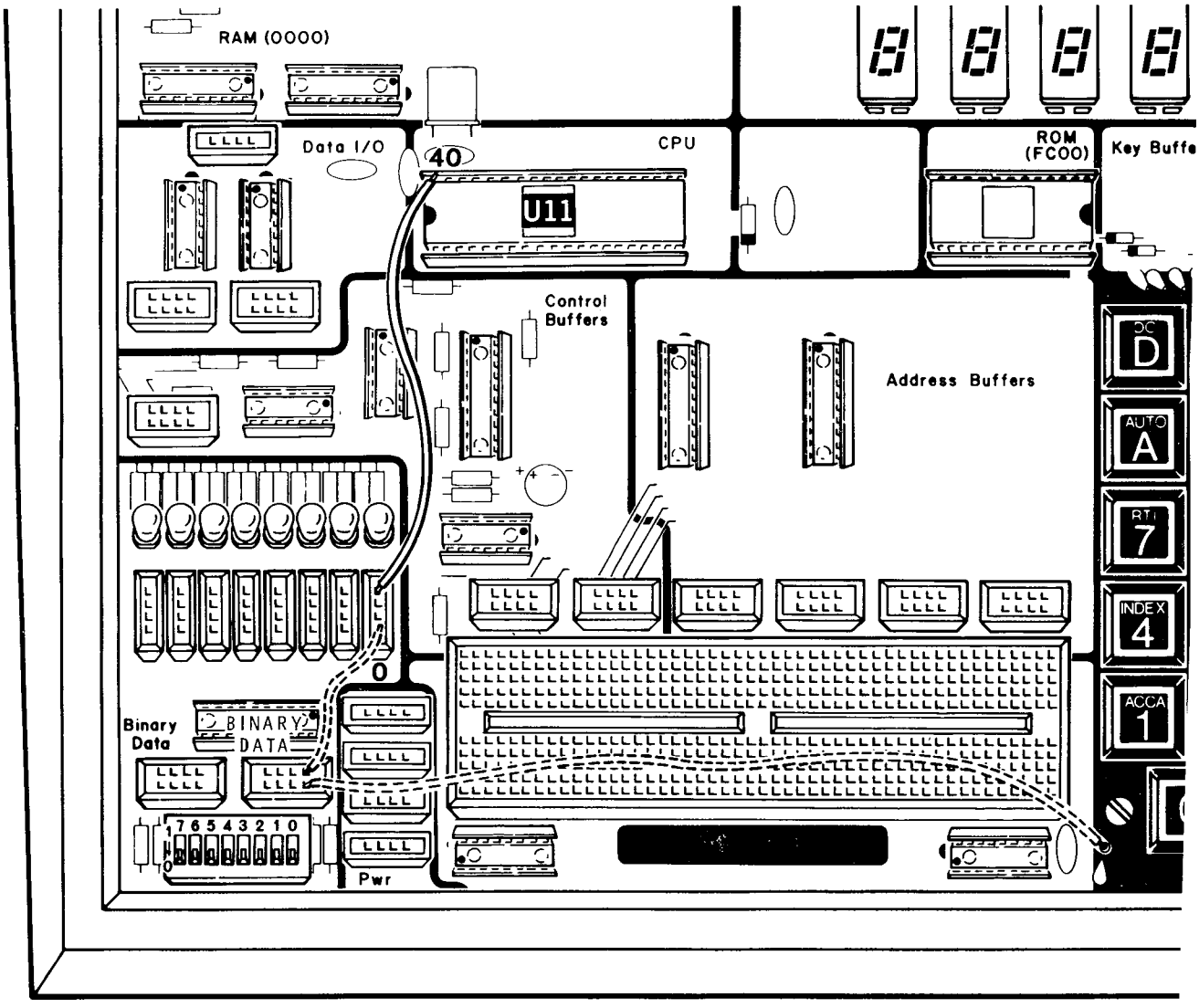
Usually, the outputs of a clock that has failed will assume a logic 0 state. To test the clock, use a wire and connect data LED block 0 to U11 pin 37. If the clock is working, the LED will light but it will be noticeably dimmer than the same LED connected to +5 volts. This is due to the 50% duty cycle of the clock. See Pictorial 9-3.



PICTORIAL 9-2



PICTORIAL 9-3



PICTORIAL 9-4

Reset

Refer to Pictorial 9-4 and connect a wire from a data LED connector block to U11 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 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 10) and unsolder and disconnect the indicated lead of resistor R27 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 D9 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:

WIRES	LENGTH
3	4"
2	8"

Refer to Pictorial 9-5 and connect a 4" wire from the ground connector block to $\overline{TSC}$.

Touch test wire 6 to the +5 connector block to test the LED. LED D10 should light.

Touch test wire 6 to U7 pin 1 and then to U8 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 117.

Touch test wire 7 to U7 pin 19 and then to U8 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_0-A_{15}) LED D9 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 prossible damage from a static charge. (See Page 45 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 D16 to A_0, and an 8" wire from data LED D15 to A_1. LED D16 should be on and LED D15 should be out. If there is a short between lines A_0 and A_1, 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 A_0 (U7 pin 12, U12 pin 8, 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 A_1 to A_2. Then move the wire at A_0 to A_1.

As before, LED D16 should be on and LED D15 should be out. If both LED's are out, this time check for a short between lines A_1 and A_2.

Connect test wire 6 to all the IC pins indicated in Test Chart A as being connected to A_1. The LED should light as each pin is touched.

Continue moving the two wires towards A_{15}, 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

BUFFERED ADDRESS LINES	BUFFERS		ROM U12	RAM U14 U15	DISPLAY LATCHES U23 U24 U25 U26 U27 U28	ADDRESS DECODING				
	U7 PIN	U8 PIN	PIN	PIN	PIN	U2 PIN	U3 PIN	U20 PIN	U22 PIN	U16 PIN
A_0	12		8	5	1					
A_1	14		7	6	2					
A_2	16		6	7	3					
A_3	18		5	4						
A_4	9		4	3					15	
A_5	7		3	2					14	
A_6	5		2	1					13	
A_7	3		1	17						
A_8		12	23					15		
A_9		14	22					14		
A_{10}		16	19				15			
A_{11}		18					14			11
A_{12}		3	21				12			
A_{13}		5				15				
A_{14}		7				14				
A_{15}		9				13				

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 U11 from its socket.

Remove U16 and bend pin 6 out slightly. Then reinstall the IC so that pin 6 is not in the socket.

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

Follow chart below and reconnect the wires.

WIRE	FROM	TO
4″	Data switch 0	LED block 0
4″	Data switch 1	LED block 1
4″	+5	$\overline{BA}$
8″	LED block 0	A_0
8″	LED block 1	A_1

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 U7 and U8. Pin 1 is logic 0 (LED D10 off) and pin 19 is logic 1 (LED D10 on).

Address lines A_0 and A_1 should be logic 1, indicated by LED D16 and LED D15 being lit. Remove the 12″ test wire from LED block 7 and insert one end in the GND connector.

Touch the free wire end to U7 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 buffer inputs.

Follow Test Chart B to check all the address lines.

TEST CHART B

8″ WIRE FROM LED BLOCK 0 TO:	8″ WIRE FROM LED BLOCK 1 TO:	GND IC	GND PIN	TURNS OFF LED
A_0	A_1	U7	8	D16
A_2	A_1	U7	6	D15
A_2	A_3	U7	4	D16
A_4	A_3	U7	2	D15
A_4	A_5	U7	11	D16
A_6	A_5	U7	13	D15
A_6	A_7	U7	15	D16
A_8	A_7	U7	17	D15
A_8	A_9	U8	8	D16
A_{10}	A_9	U8	6	D15
A_{10}	A_{11}	U8	4	D16
A_{12}	A_{11}	U8	2	D15
A_{12}	A_{13}	U8	17	D16
A_{14}	A_{13}	U8	15	D15
A_{14}	A_{15}	U8	13	D16
A_{14}	A_{15}	U8	11	D15

Properly replace U11 and U16 into their sockets.

Remove all the wires except test wires 6 and 7.

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.

WIRES	LENGTH
3	8″
3	4″

If not already done, refer to Page 111 and prepare the two test wires as instructed there.

Refer to Pictorial 9-6 (Illustration Booklet, Page 10) and install a 4″ wire between GND and $\overline{HALT}$ 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 U9 and U10. The LED should light, indicating that the buffers are in the right state.

Touch the test lead to each of the data connectors (D_0 — D_7). 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.

WIRE	FROM	TO
4″	Data switch 0	LED block 0
4″	Data switch 1	LED block 1
8″	LED block 0	D_0
8″	LED block 1	D_1
8″	GND	$\overline{RE}$

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 D_0 is connected. The LED should light. If the LED does not light, there is a short between the D_0 and D_1 lines. Visually check for shorts. Remove the IC's connected to the D_0 and D_1 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 D_0 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 D_0 also goes to the display latches and is inserted at U21 pins 4 and 5. The inverted D_0 output, U21 pin 6 and U6 pin 2, is connected from the output of U6 (pin 18) to pin 13 of U23 through U28.)

Move the wire that is at D_1 to D_2. Then move the wire at D_0 to D_1.

TEST CHART C

	BUFFERS				CPU	ROM	RAM		KEYBOARD BUFFER
	CPU SIDE		CONNECTOR SIDE						
	U9 PIN	U10 PIN	U9 PIN	U10 PIN	U11 PIN	U12 PIN	U15 PIN	U14 PIN	U13 PIN
D_0		8		6	33	9	14		3
D_1		9		5	32	10	13		5
D_2		10		4	31	11	12		9
D_3		11		3	30	13	11		7
D_4	8		6		29	14		14	11
D_5	9		5		28	15		13	13
D_6	10		4		27	16		12	
D_7	11		3		26	17		11	

Remove all the wires except the test wires.

Control Lines

If not already done, refer to Page 111 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 $\overline{RESET}$, $\overline{HALT}$, $\overline{IRQ}$, and $\overline{NMI}$. 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. (Pin 2 for $\overline{HALT}$, pin 6 for $\overline{NMI}$, and pin 4 for $\overline{IRQ}$.) 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, U6 may be defective.

To check the five remaining control lines (R/$\overline{W}$, TSC, BA, VMA, and DBE) plus $\overline{VMA\phi 2}$, 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.

TOUCH GROUND WIRE TO CONNECTOR	TSC U4 PIN 12	R/$\overline{W}$ U11 PIN 34	VMA U11 PIN 5	BA U11 PIN 7	U4 PINS 1, 4, 10	U4 PIN 13	$\overline{VMA\phi 2}$	$\overline{BA}$	$\overline{TSC}$	R/$\overline{W}$
———	0	1	1	0	1	1	1 [1]	1	1	1 [2]
$\overline{HALT}$	0	0	0	1	0	0	1	0	1	1
$\overline{TSC}$	1	0	0	0	0	1	1	1	0	1

[1] The $\overline{VMA\phi 2}$ 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 $\overline{Halt}$ or TSC states.

[2] Although the R/$\overline{W}$ line changes, the output connector does not change because the buffer is tri-stated when R/$\overline{W}$ is low.

Remove all the wires except test wires 6 and 7.

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/$\overline{W}$ is high. In addition, the CPU puts the first address of the reset sequence on the address line. This address is FFFE. Therefore, press the RESET button and test all the address lines with test wire 6. They will all be logic 1 except for A_0, 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.

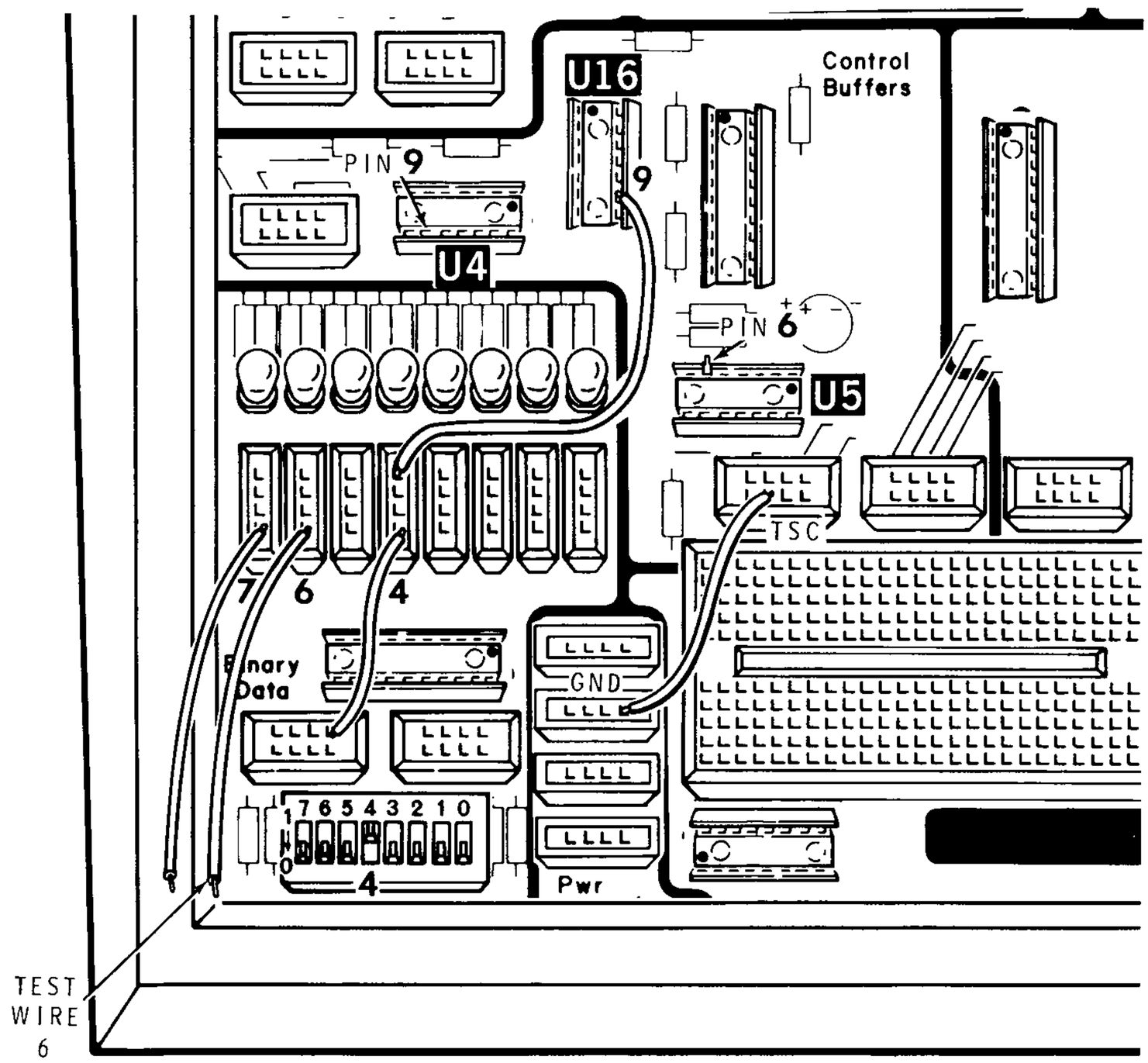
If not already done, refer to Page 111 and prepare the two test wires as instructed there.

$\overline{VMA\phi 2}$, for internal operation, is taken from the line connecting U5 pin 6 to U16 pin 9. To perform tests on the decoding section, you will need to pull this line to logic 0. To avoid damaging U5 when you do this, refer to Pictorial 9-7, carefully remove U5 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. Temporarily solder the free end of this wire to pin 8 of U16.

Install a 4" wire from LED 4 to data switch 4. Data switch 4 will now determine the logic level of $\overline{VMA\phi 2}$, and LED 4 will display the level. 1 is ON, 0 is OFF.

Install a 4" wire from GND to $\overline{TSC}$ 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 11 and 12 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 $\overline{VMA\phi 2}$ is doing its job, switch D4 between logic 1 and logic 0. The chart "End Result" should only be obtained when $\overline{VMA\phi 2}$ 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 FF — Remove the wire installed between A_9 and A_{10}, 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_{13}, and install it between A_9 and A_{10}. Address lines A_3 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_0 through D_5). 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, D_0 through D_5. 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 D_0 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 D_0 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 D_0 should be logic 0. If, for example, lines D_0 and D_1 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 D_0 is now logic 0, the problem exists between keys F and C. If D_0 and D_1 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 D_0 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 A_8-A_9. Move the GND wire from A_8 to A_9. Then connect a wire from A_8 to +5. Check for the proper end result indicated in the Main Decoding Chart. Move the 8″ wires installed at A_0, A_1, and A_2 to A_4, A_5, and A_6. 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 segement CIXX, move the 8″ wires from A_4, A_5, and A_6 to A_0, A_1, and A_2 respectively. Now use data switch 0, 1, and 2 to address the desired segment. Move the wire soldered to VMAϕ2 from LED 4 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 A_4, A_5, and A_6. (NOTE: The data switches are one number from the corresponding address line so LED's D10 and D11 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 D_0 data line controls the state of the LED segment when the segment is addressed and VMAϕ2 is logic 0. If D_0 is logic 1, the segment will light and if D_0 is logic 0, the segment will be off. The D_0 data line is connected through U21 and U6 to the D data input (pin 13) of decoder latch IC's.

The "D_0 Logic Level Chart" shows the levels at the various connections on the D_0 segment control line. To control the logic level on the D_0 data line, connect an 8″ wire from $\overline{RE}$ to GND. Connect another 8″ wire to the D_0 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 D_0. 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 D_0 is logic 1, the segment should light and the logic probe should indicate logic 0. The reverse is true if D_0 is logic 0.

If you wish to check different LED's or segments, inset the D_0 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 $\overline{VMA\phi 2}$ from GND to LED block 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 D_0 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 $\overline{VMA\phi 2}$ is moved to logic 1.

Remove the wires from your Trainer, properly replace U5, and then reconnect and resolder the free lead of resistor R27.

SPECIFICATIONS

CPU (Central Processing Unit)	8-bit parallel, NMOS, bus oriented 6808.
ROM (Read Only Memory)	NMOS, 1024 bytes.
RAM (Random Access Memory)	NMOS, 512 accessible bytes.
Clock Frequency	1 MHz (approximately).
Display	Six 7-segment LED digits.
Keyboard	Hexadecimal (0-F and Reset). 1 through F are dual-function keys and also enter commands.
Input Switches	Eight miniature switches in a dual-in-line package. (For experiments.)
LED Monitor Lights	Eight red LED's with separate input terminals. (For experiments.)
Power Supplies	+5 volts at 1.5 amperes (.5A available for breadboard at output terminal.)
	+12 volts, and −12 volts at 50 milliamperes at output terminals.
Power Requirements	105-130 volts rms, 50-60 Hz. 30 watts maximum.
Fuse	3/8-ampere, slow-blow.
Dimensions	12-1/8″ wide × 11-3/4″ deep × 3-1/2″ high. (30.8 × 29.8 × 8.9 cm.)
Net Weight	4 lbs (1.8 kg).

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 13) and the Schematic (fold-in).

The operation of the CPU (microprocessor U11) 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 M6808 application manual or the Heath Microprocessor course EE-3401.

Many lines are connected to front panel connectors, as shown in 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 capacitor C17 and resistor R53, which produce a proper reset pulse. This pulse is applied through tri-state buffer U6 to the reset input (pin 40) of the CPU.

The $\phi 2$ clock output is used for external peripheral timing and is connected through U4C to a connector block.

The VMA line from the CPU is buffered by U6G, ANDed by U5B with $\phi 2$ clock, and then inverted by U16D to produce $\overline{VMA\phi 2}$. This signal is then applied to the address decoding circuits.

The CPU R/$\overline{W}$ line is coupled through U4A to the R/$\overline{W}$ inputs of RAM and is also connected to a front panel connector for use by peripherals.

The input signal $\overline{TSC}$ is applied through inverter U16B. $\overline{TSC}$ is normally logic 1 and after inversion is connected through U4D to the input of U16A and to $\overline{G}$ of the address buffers U7 and U8. Line BA is normally connected through U16C to the control line of the TSC portion of U4D. The output of U16A is logic 1 and is connected to the control lines in U4 for R/$\overline{W}$, and $\overline{VMA\phi 02}$; keeping these sections enabled. The output of U16A is also connected to the enable input of the address buffers U7 and U8.

When the TSC input is logic 0, the input to U16A and $\overline{G}$ on the address buffers become logic 1. The output of U16A and, therefore, the G inputs to address buffers U7 and U8 and the control lines for the other two sections of U4 become logic 0. The address, R/$\overline{W}$, and $\overline{VMA\phi 2}$ buffers are all tri-stated. In this state, the R/$\overline{W}$ and VMA02 lines are held at logic 1 by pull-up resistors. When BA goes to logic 0, the TSC section of U4 is tri-stated, TSC does not control the output, and the output is held at logic 1 by a pull-up resistors which tri-states the address buffers, R/$\overline{W}$, and $\overline{VMA\phi 2}$ as described above.

The address lines are buffered by U7 and U8. 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; $\overline{VMA\phi 2}$ 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 U14 and U15, and its 256 bytes of RAM memory may be addressed by lines A_0 — A_7. R/$\overline{W}$ 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 256 bytes of RAM at that address.

With the high-order byte FC, FD, FE, FF; the address decoder places a logic 0 in $\overline{CS1}$ of the ROM. Address lines A_{10} and A_{12} place logic 1 on CS0 and CS3; and lines A_0 through A_9 can address the 1024 bytes of read only memory.

Buffer U13 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 A_0, A_1, and A_2 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 U22. U22 is a 4 to 9 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, U23 through U28. If the D input to U22 (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 A_4, A_5, and A_6 and will determine which output line will be logic 0 by their logic levels. A hex 6 or BCD 110 on lines A_6, A_5, and A_4 will cause the enable line for the left-most latch driver (U23 and DISPLAY LED H) to be logic 0. Hex 1 or BCD 001 enables U28 and DISPLAY LED C.

Address lines A_0, A_1, and A_2 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 and 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 D_0 data line is inverted by U21B and applied to the latch driver D inputs through U6A. Therefore, if line D0 is logic 1, and D input is logic 0 and the addressed segment will be lit. If D0 is logic 0, and 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 U9 and U10 are bus transceivers. They are wired to normally provide output from the data lines to the data terminals. Connecting $\overline{RE}$ 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 inverting buffer U1. 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 150 ohm current-limiting resistor to the switched 5V power supply. When the inputs to the buffer are logic 0, the outputs are logic 1 and the LED is off. When the inupt rises to logic 1, the output also drops to logic 0 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 C4, and regulated by U31 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 D4 and D6, filtered by C11, and regulated by U30 to provide a −12-volt supply. This same winding is rectified by diodes D5 and D7, filtered by C9, and regulated by U29 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.

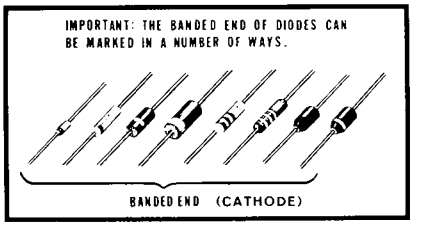
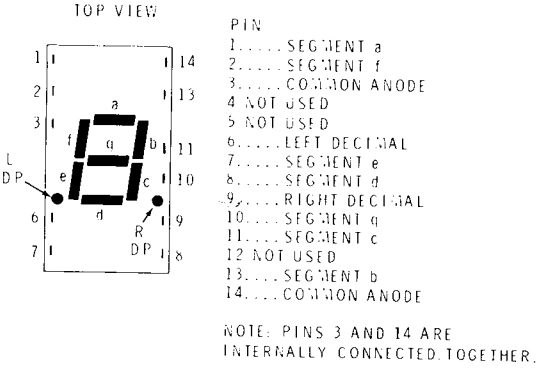
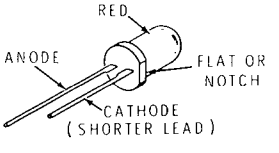
SQUARE WAVE OUTPUTS

The AC voltage at the anode of diode D7 is coupled by R5 and R6 to a section of voltage comparator U18. Diode D8 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.

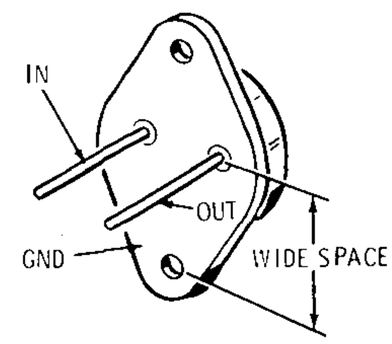
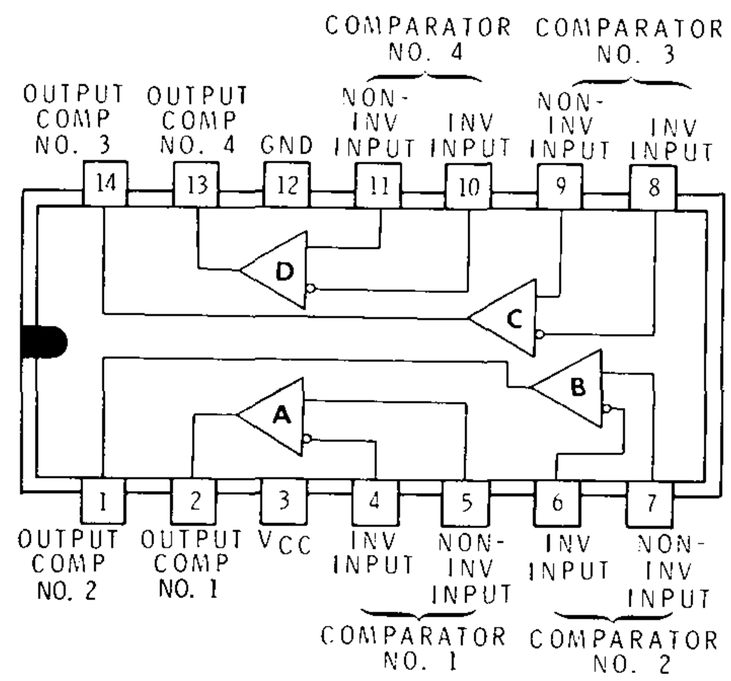
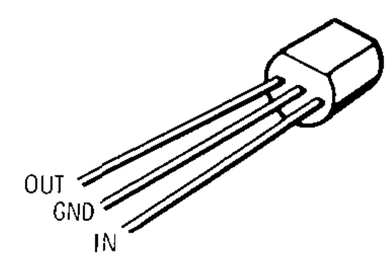
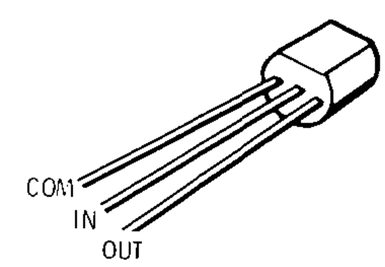
A second section of U18 is used as an oscillator to produce a TTL compatible square wave at approximately 1 Hz. The symmetry and frequency of the square wave are determined by C16, R15, and R16.

SEMICONDUCTOR IDENTIFICATION CHARTS

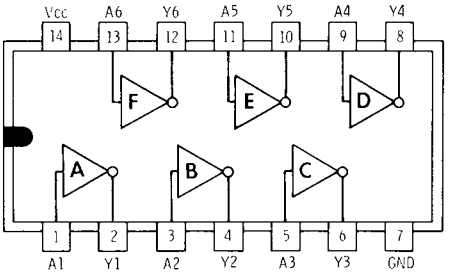
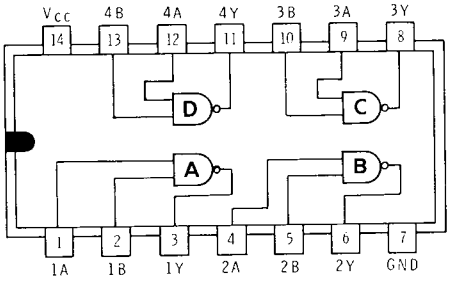
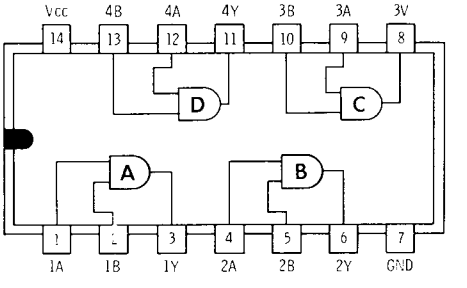
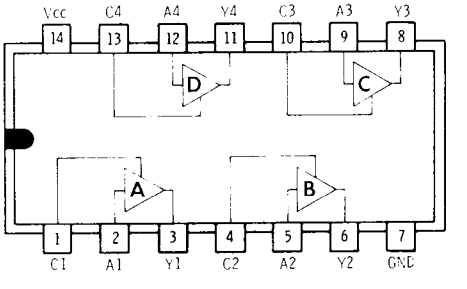
DIODES

COMPONENT	HEATH PART NUMBER	MAY BE REPLACED WITH	IDENTIFICATION
D8, D18, D19, D20	56-56	1N4149	
D1, D2	57-42	3A1	
D4, D5, D6, D7	57-65	1N4002	
V1, V2, V3, V4, V5, V6	411-831	TIL312	
D3, D9, D10, D11, D12, D13, D14, D15, D16	412-640	FLV117	

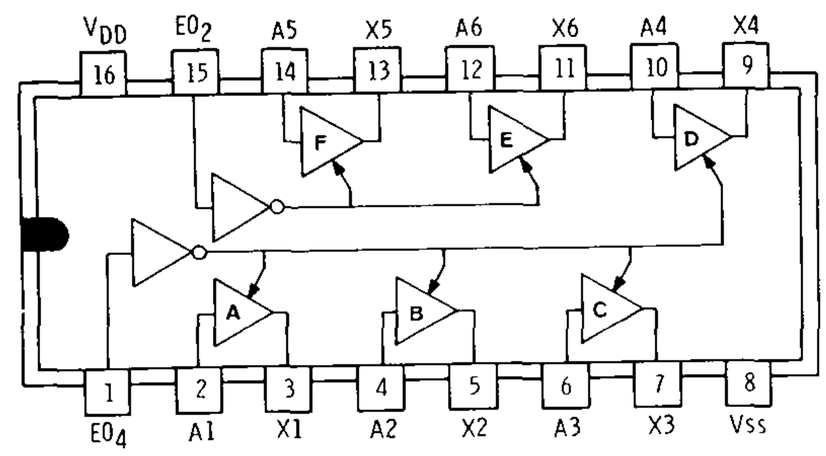
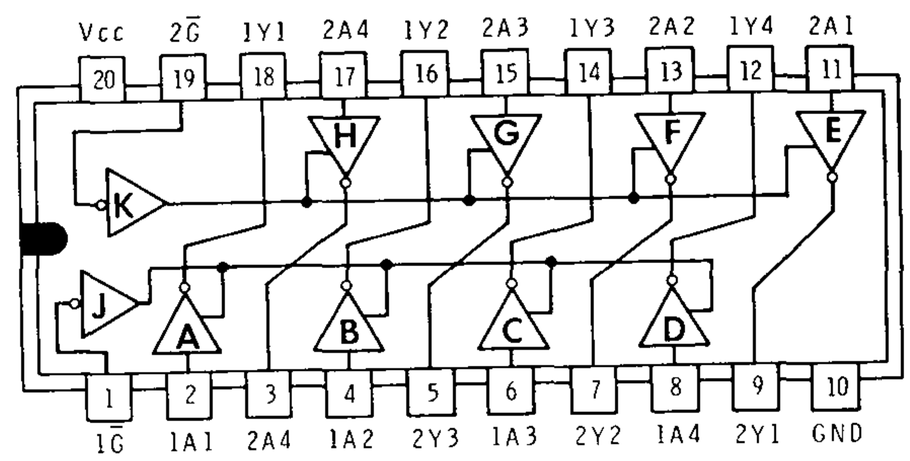
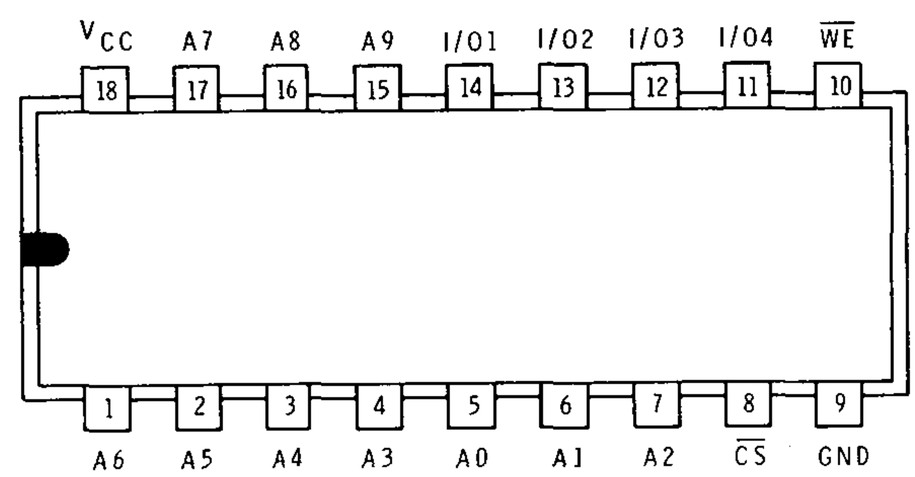
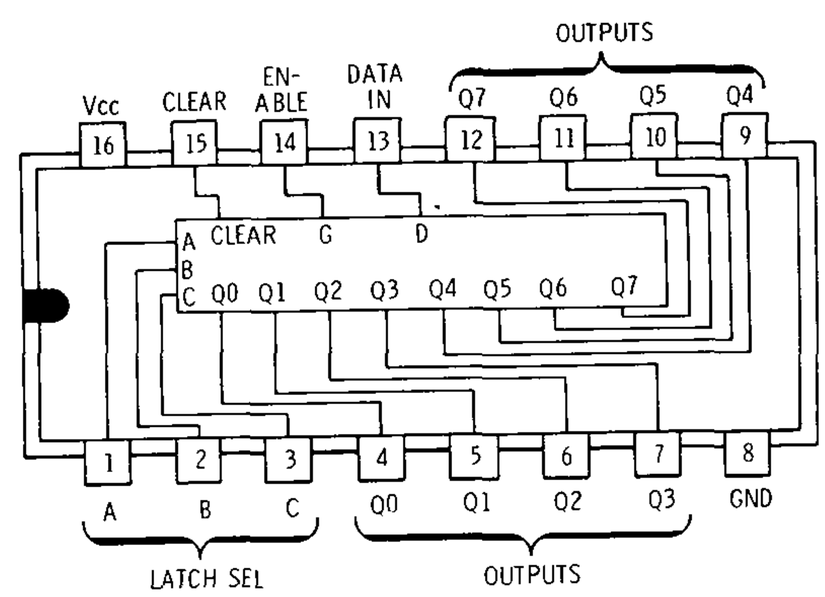
INTEGRATED CIRCUITS

COMPONENT	HEATH PART NUMBER	MAY BE REPLACED WITH	IDENTIFICATION
U31	442-30	μA309K	
U18	442-616	LM3302N, LM2901N, or μA775 (selected)	
U29	442-644	78L12	
U30	442-646	79L12	

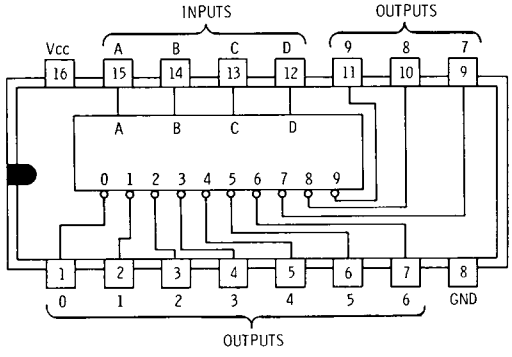
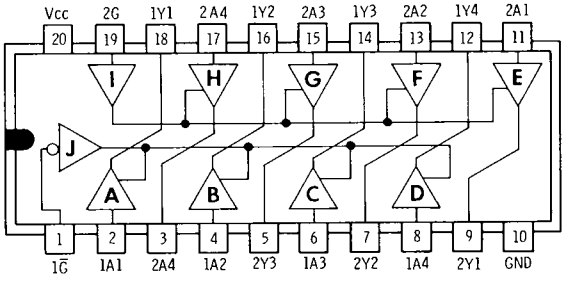
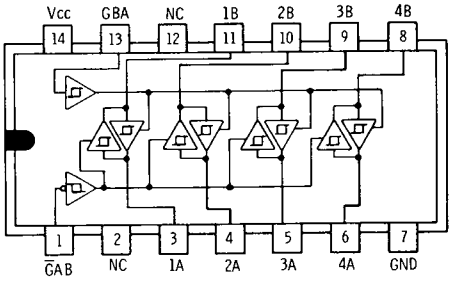
Integrated Circuits (cont'd.)

COMPONENT	HEATH PART NUMBER	MAY BE REPLACED WITH	IDENTIFICATION
U16	443-897	74S04	
U21	443-26	74S00	
U5	443-45	7408	
U4	443-717	74126	 FUNCTION TABLE INPUTS: A, C; OUTPUT: Y A C Y H H H L H L X L Hi-Z Y = A Hi - Z = HIGH IMPEDANCE

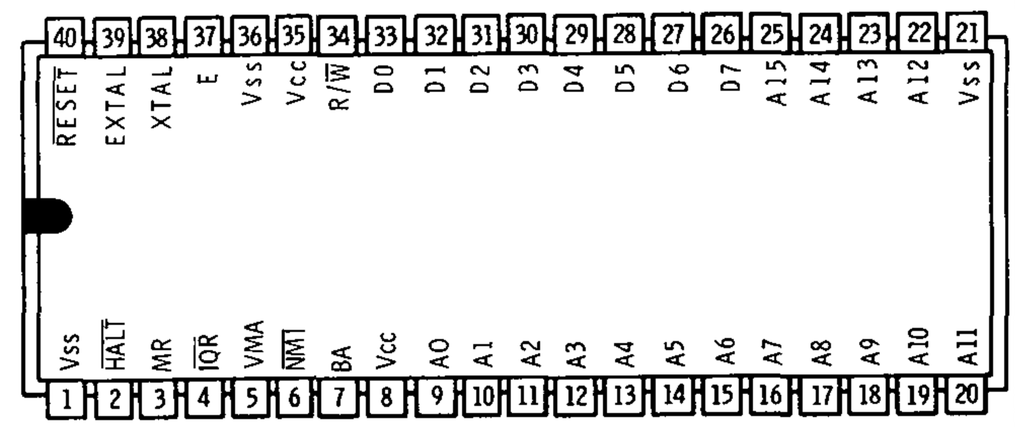
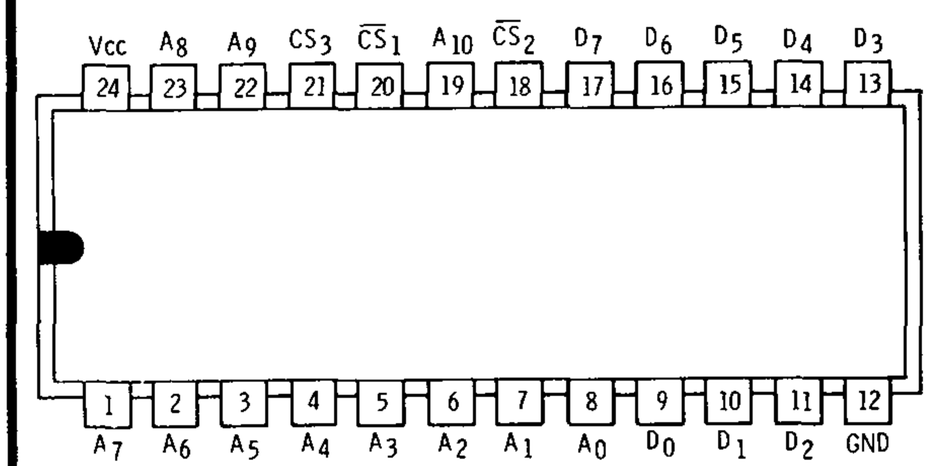
Integrated Circuits (cont'd.)

COMPONENT	HEATH PART NUMBER	MAY BE REPLACED WITH	IDENTIFICATION
U13	443-720	40097	
U1	443-754	74LS240	
U14, U15	443-764	2114	
U23, U24, U25, U26, U27, U28	443-804	74LS259	

Integrated Circuits (cont'd.)

COMPONENT	HEATH PART NUMBER	MAY BE REPLACED WITH	IDENTIFICATION
U2, U3, U20, U22	443-807	74LS42	
U6, U7, U8	443-824	74LS241	
U9, U10	443-839	74LS243	

Integrated Circuits (cont'd.)

COMPONENT	HEATH PART NUMBER	MAY BE REPLACED WITH	IDENTIFICATION
U11	443-939	MC6808	
U12	444-76	ROM*	

*Must be mask programmed from the listing in this Manual.

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

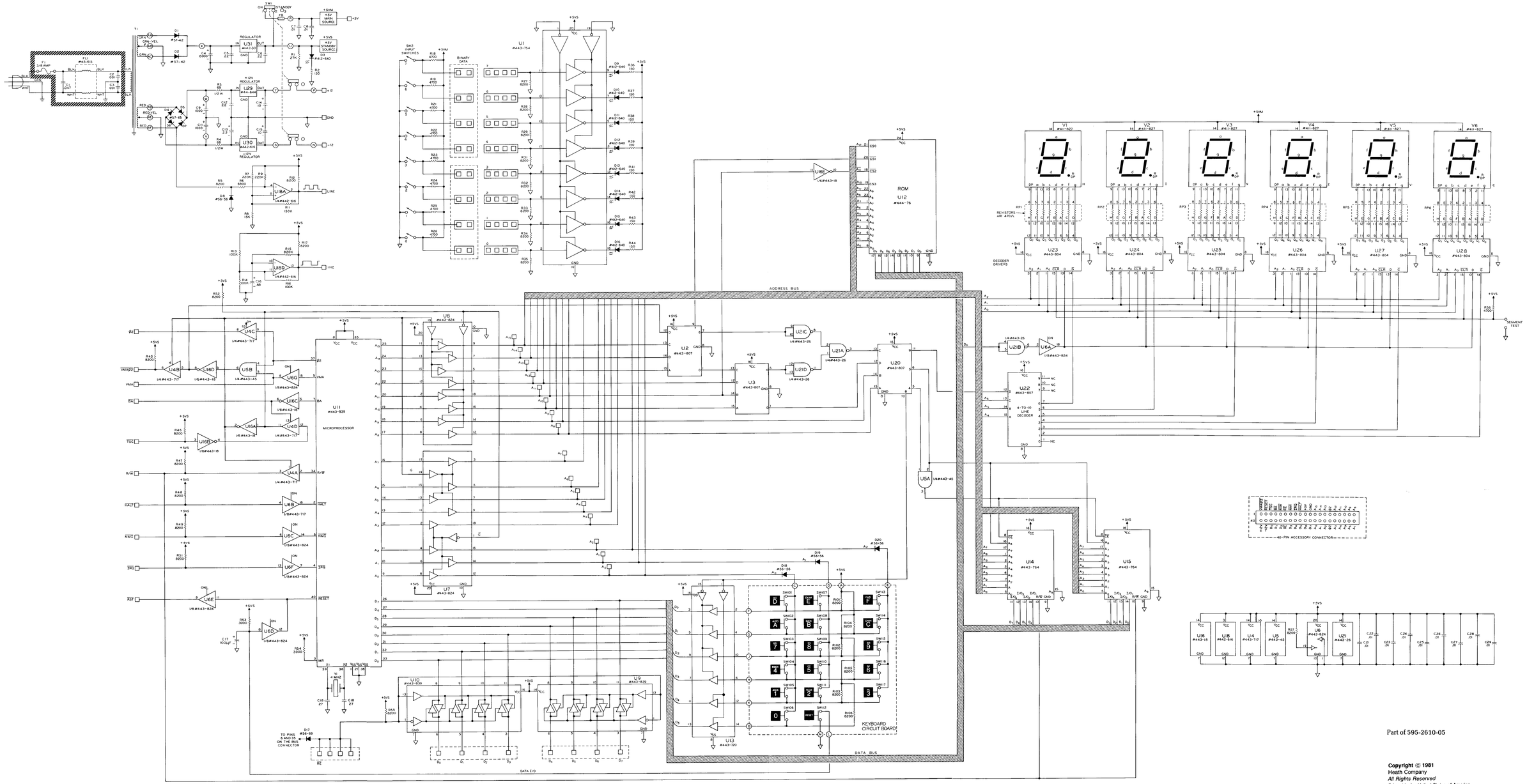
LITHO IN U.S.A.

SCHEMATIC OF THE HEATHKIT® MICROCOMPUTER LEARNING SYSTEM

Model ET-3400A

NOTES:

1. ALL RESISTORS ARE RATED AT 1/4-WATT AND HAVE A 5% TOLERANCE UNLESS NOTED OTHERWISE. RESISTOR VALUES ARE IN OHMS (K=1000).
2. CAPACITOR VALUES LESS THAN 1 ARE IN µF (MICROFARADS); CAPACITOR VALUES GREATER THAN 1 ARE IN pF (PICOFARADS) UNLESS OTHERWISE NOTED.
3. ⬭ THIS SYMBOL INDICATES A DC VOLTAGE TAKEN WITH A HIGH IMPEDANCE INPUT VOLTMETER FROM THE POINT INDICATED TO CHASSIS GROUND. VOLTAGES MAY VARY ±20%.
4. ▽ THIS SYMBOL INDICATES CIRCUIT BOARD GROUND.
5. ☐ THIS SYMBOL INDICATES A CONNECTOR IN A CONNECTOR BLOCK.
6. COMPONENTS IN THE SHADED AREA ARE CRITICAL TO CONTINUED PRODUCT SAFETY. REPLACE THEM ONLY WITH PARTS OF THE SAME RATING OR WITH PROPER HEATH REPLACEMENT PARTS.
7. IF YOU INSTALL THE 40-PIN ACCESSORY CONNECTOR, USE JUMPER WIRES TO CONNECT THE DATA LINES.



Part of 595-2610-05

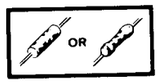
Copyright © 1981
Heath Company
All Rights Reserved
Printed in the United States of America

ILLUSTRATION BOOKLET

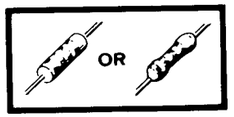
Part of 595-2610-07

PARTS PICTORIAL

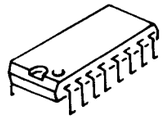
A1



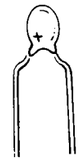
A2



A3

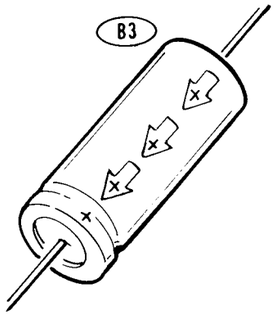


B1

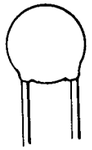


B2

B3

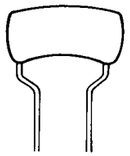


B4

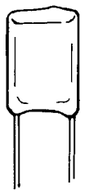


B5

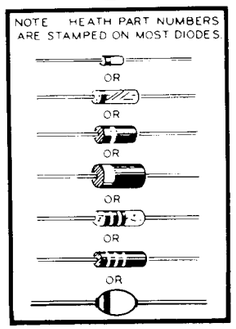
B6



B7



C1



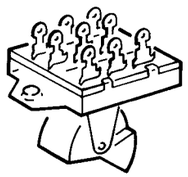
D1

D2

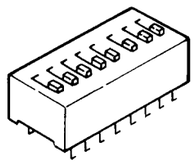
D3

ALWAYS IDENTIFY AN I.C. BY THE PART NUMBER OR TYPE NUMBER
NOTE: THE STYLE MAY BE SLIGHTLY DIFFERENT THAN SHOWN.

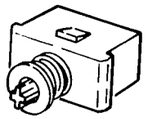
E1



E2

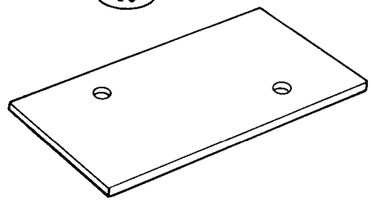


E3

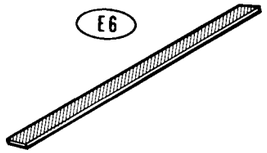


E4

E5



E6



F1

F2

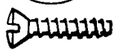
F3

F4

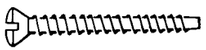
F5

F6

F7



F8



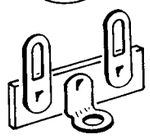
F9

F10

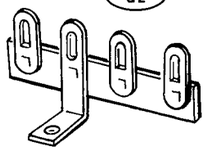
F11

F12

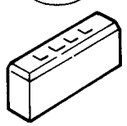
G1



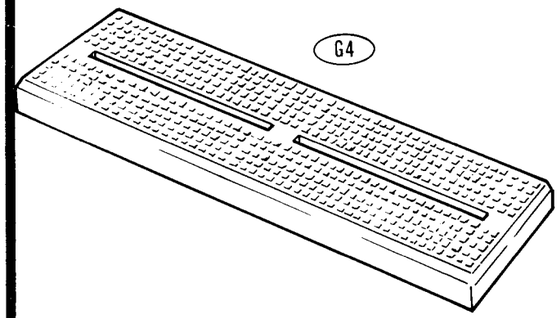
G2



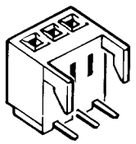
G3



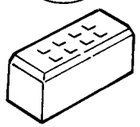
G4



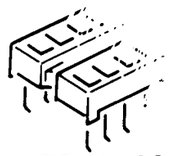
G5



G6

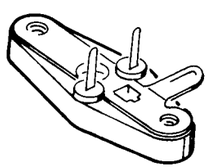


G7



ALWAYS IDENTIFY AN I.C. SOCKET BY COUNTING ITS PINS.
NOTE: THE STYLE MAY BE SLIGHTLY DIFFERENT THAN SHOWN.

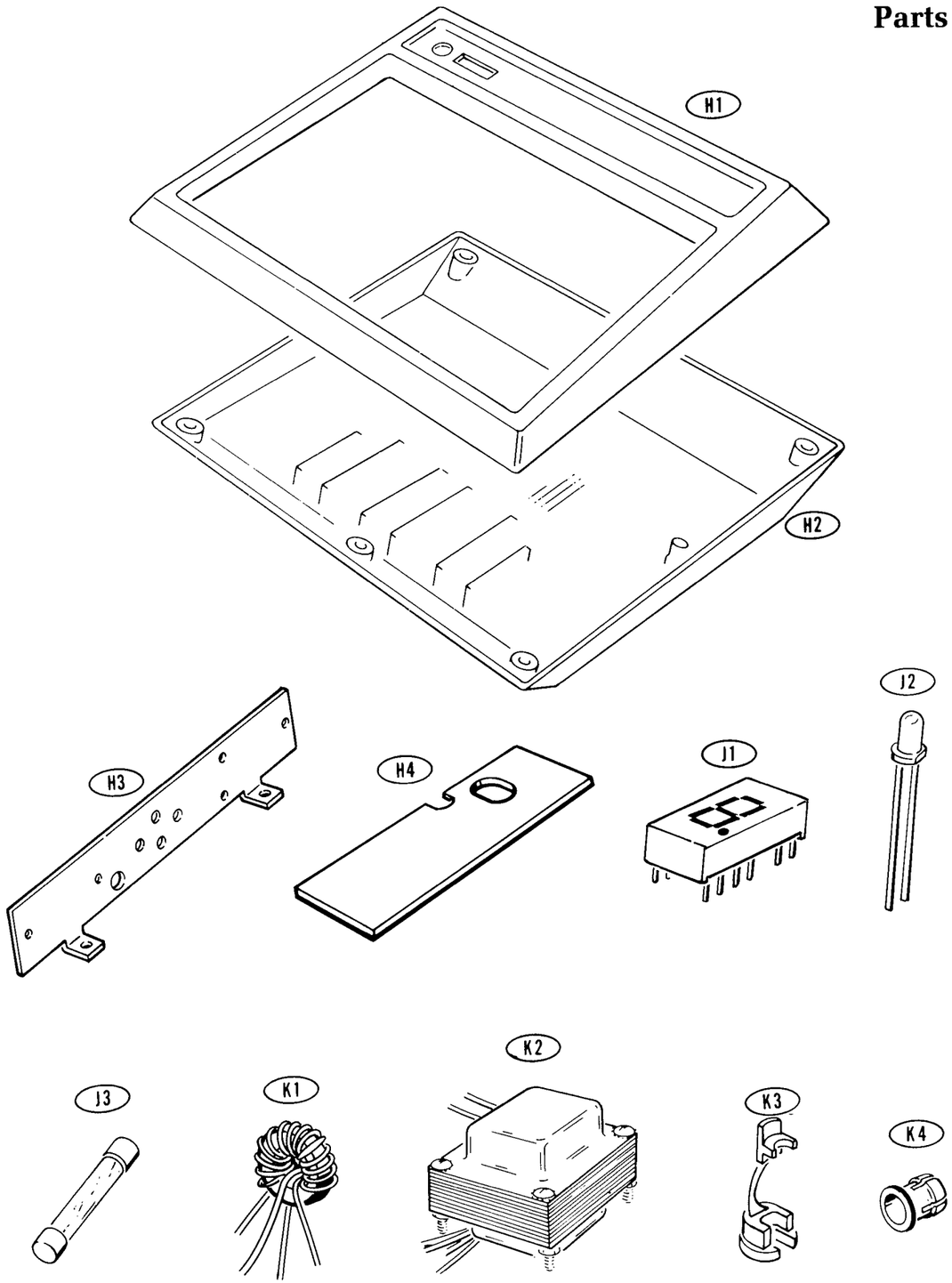
G8

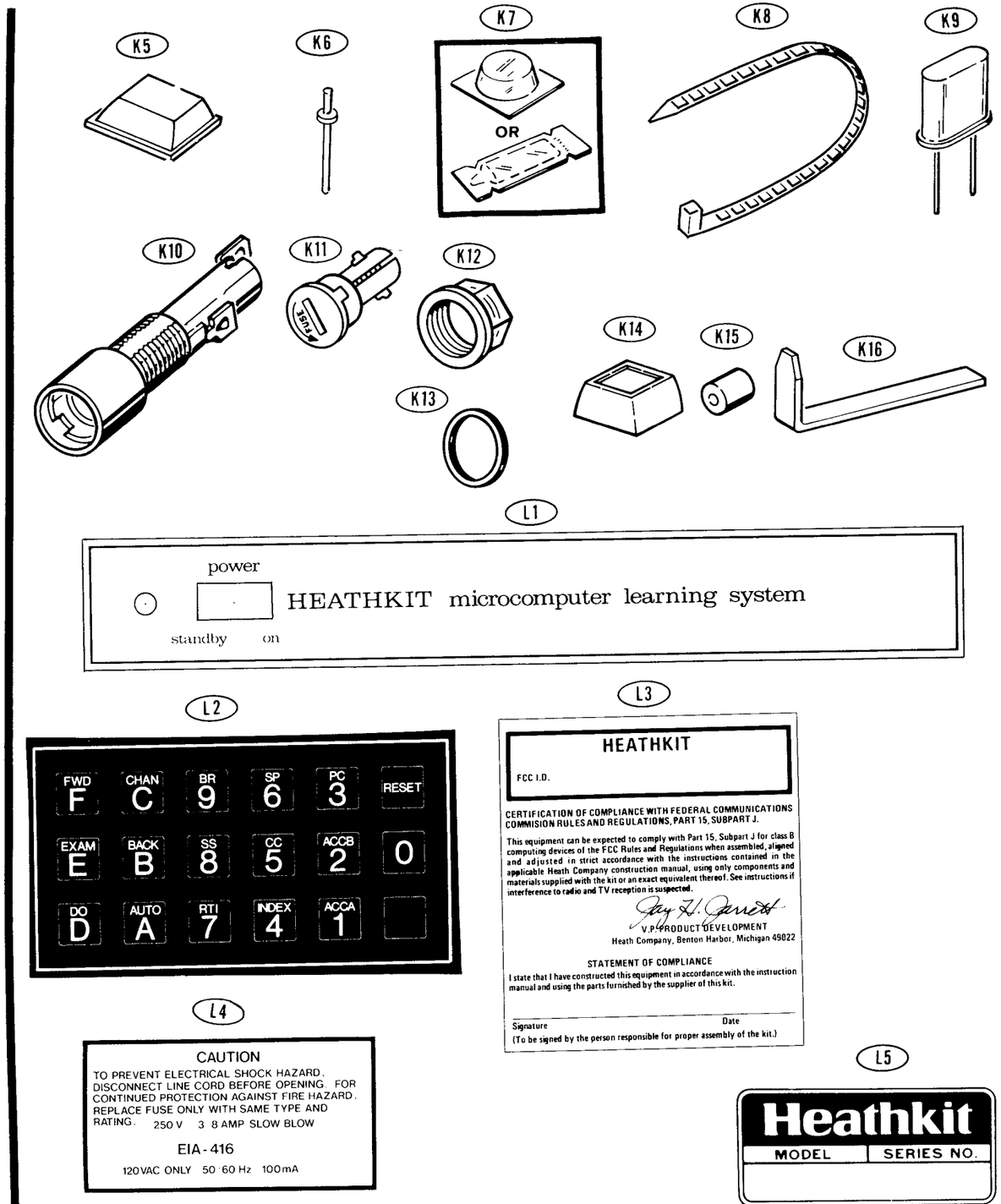


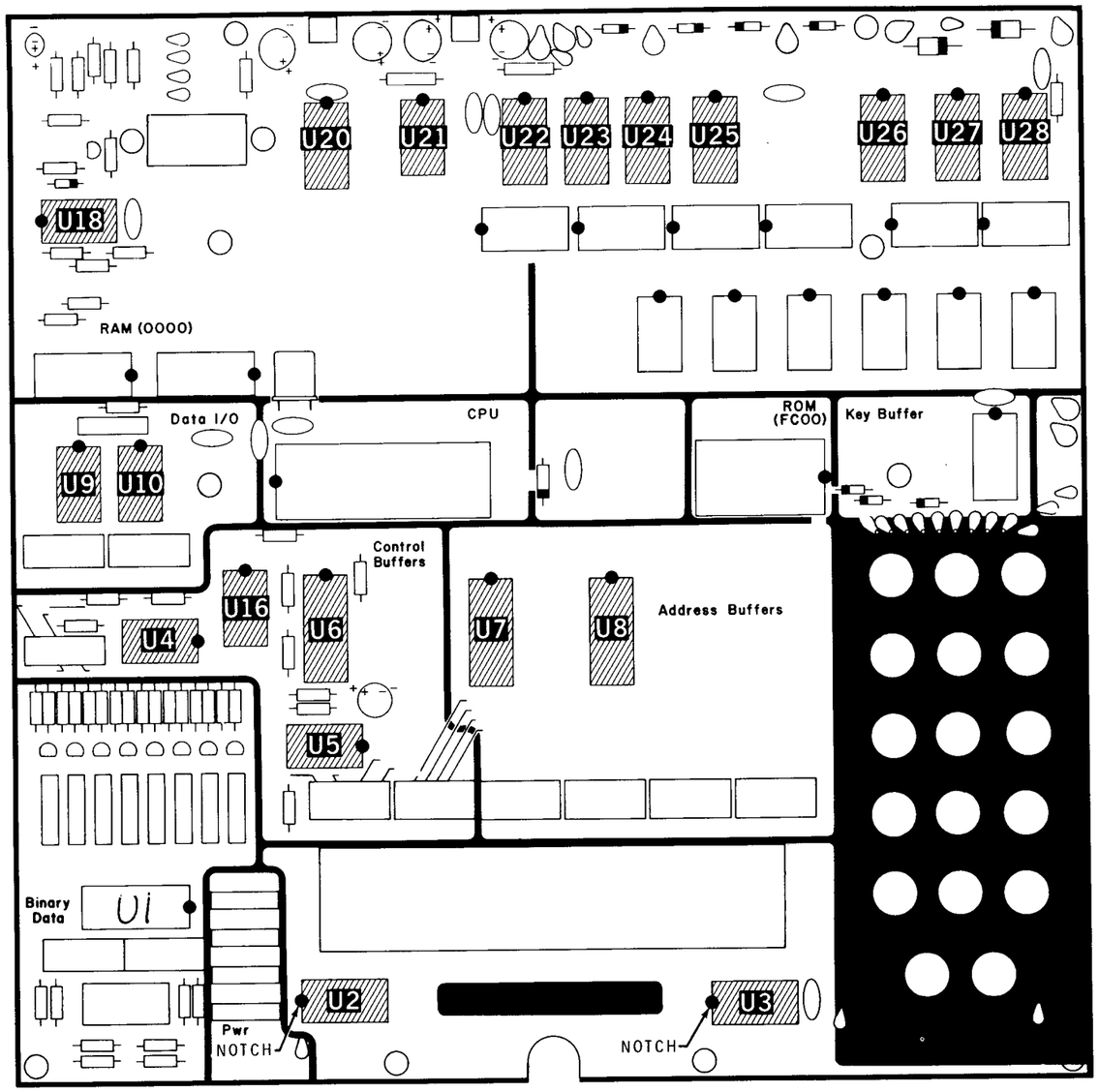
Model ET-3400A

Copyright © 1981
Heath Company
All Rights Reserved
Printed in the United States of America

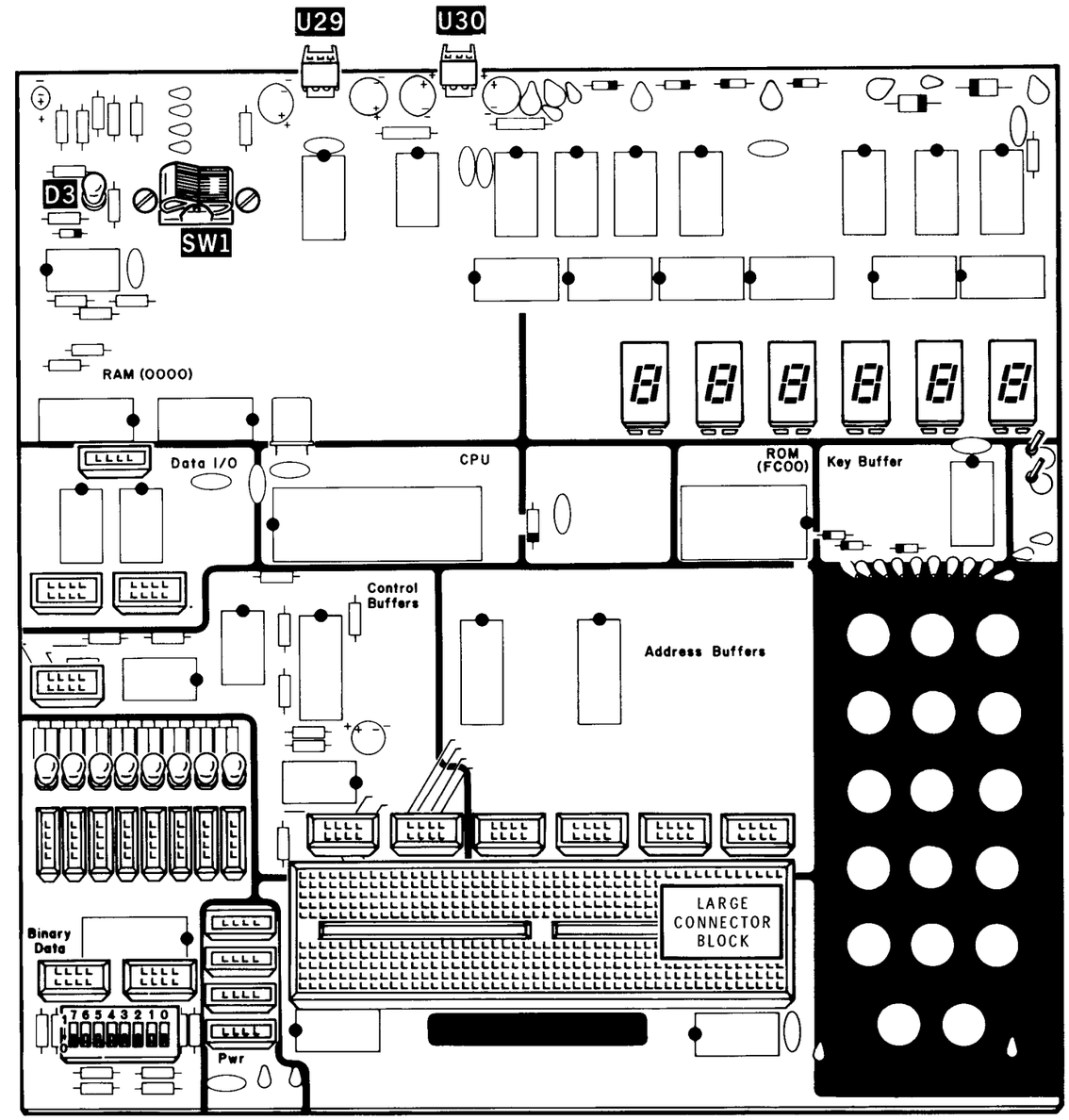
Parts Pictorial (cont'd.)



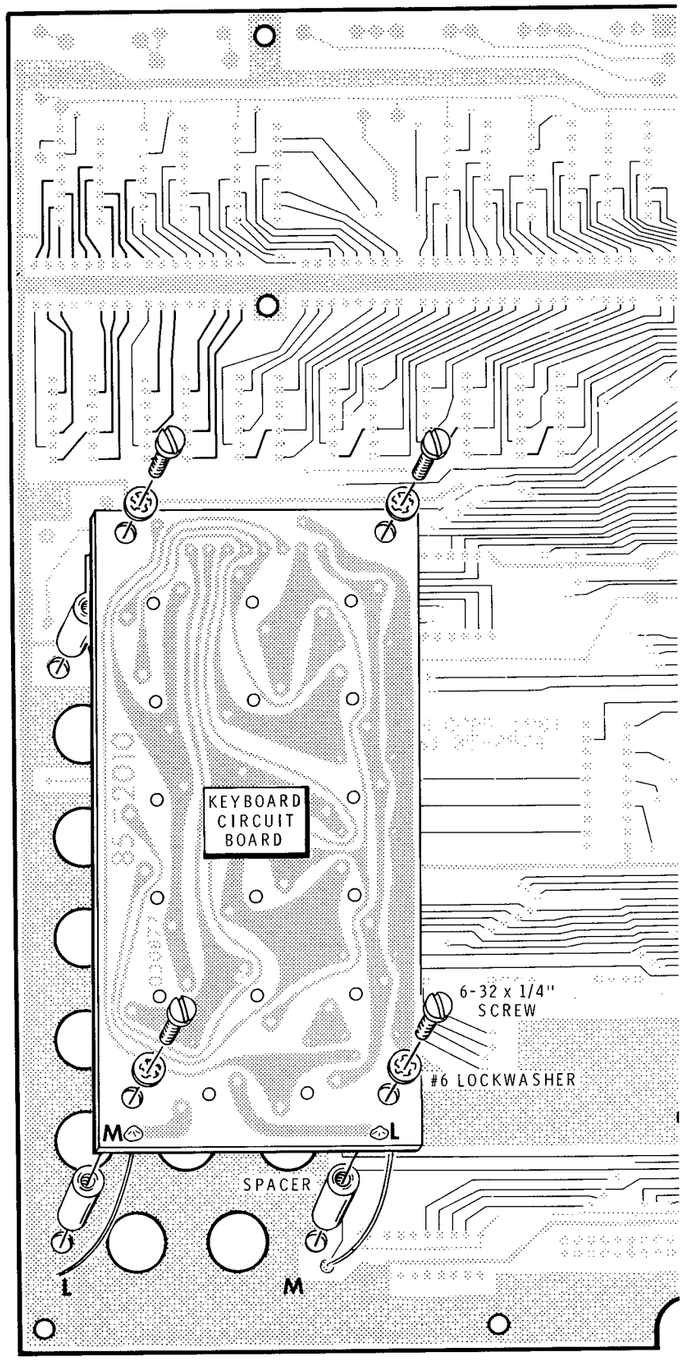




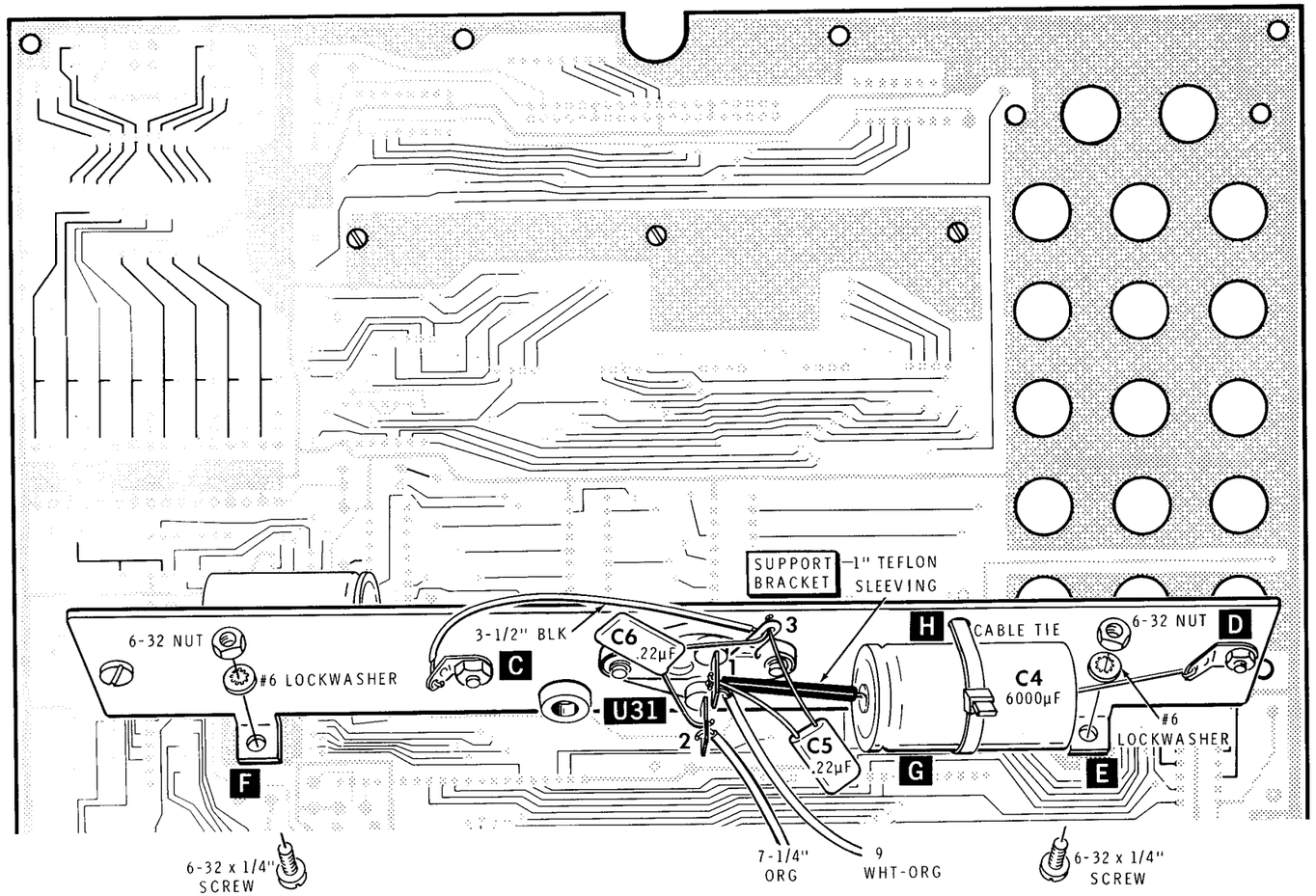
PICTORIAL 1-13



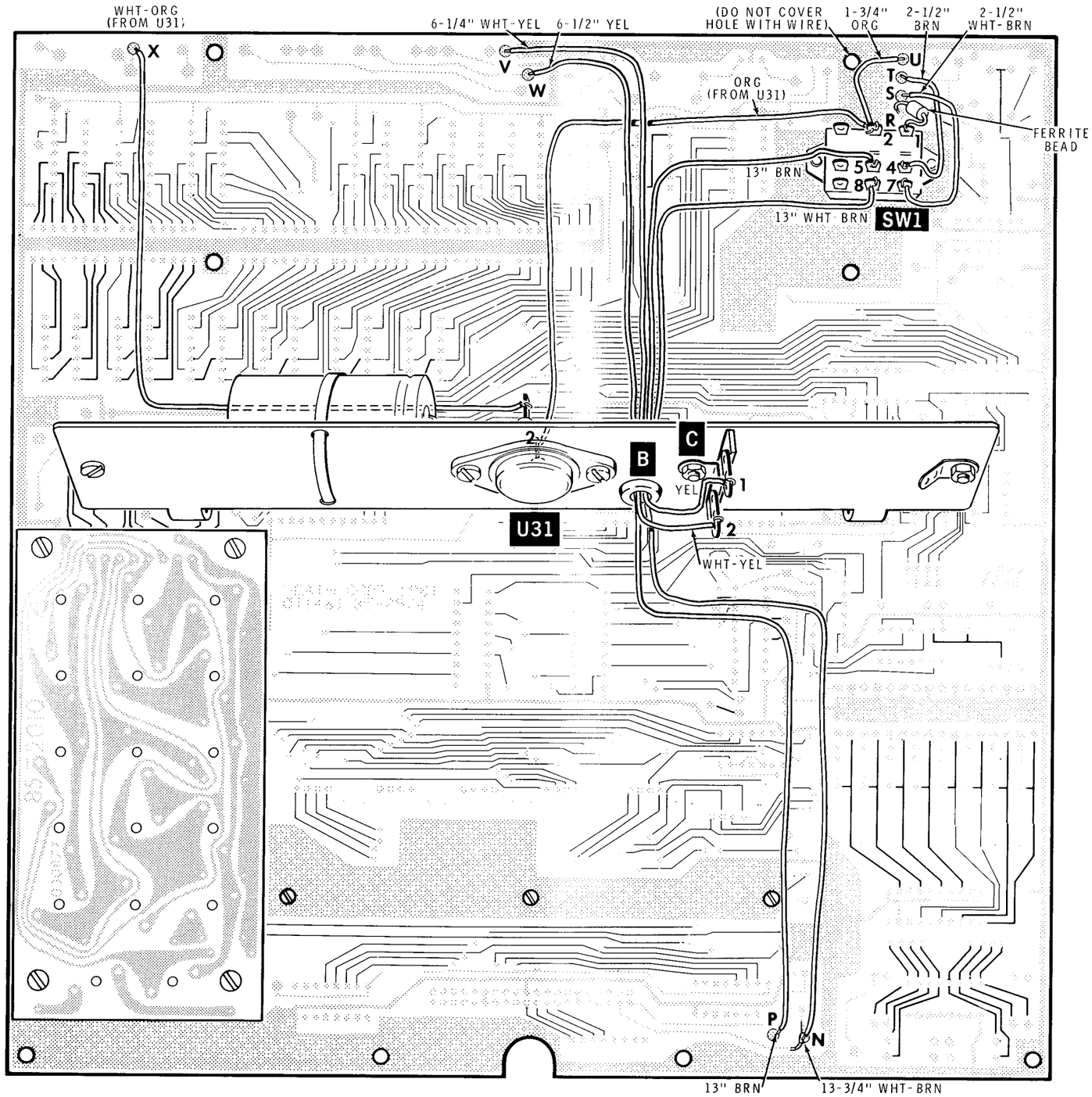
PICTORIAL 1-14



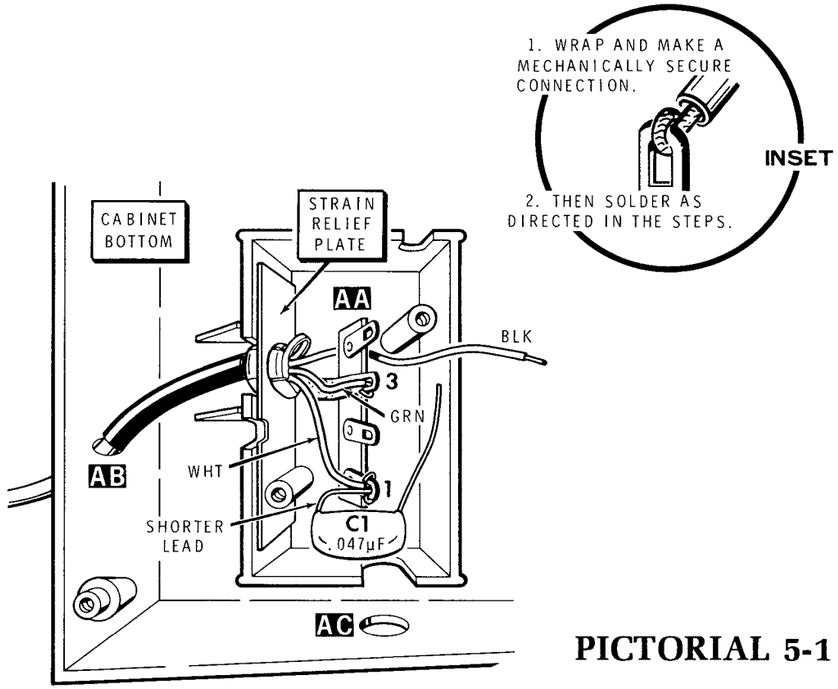
PICTORIAL 3-1



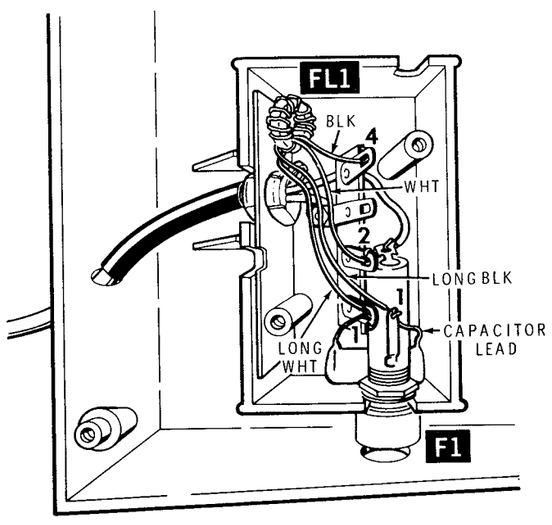
PICTORIAL 4-2



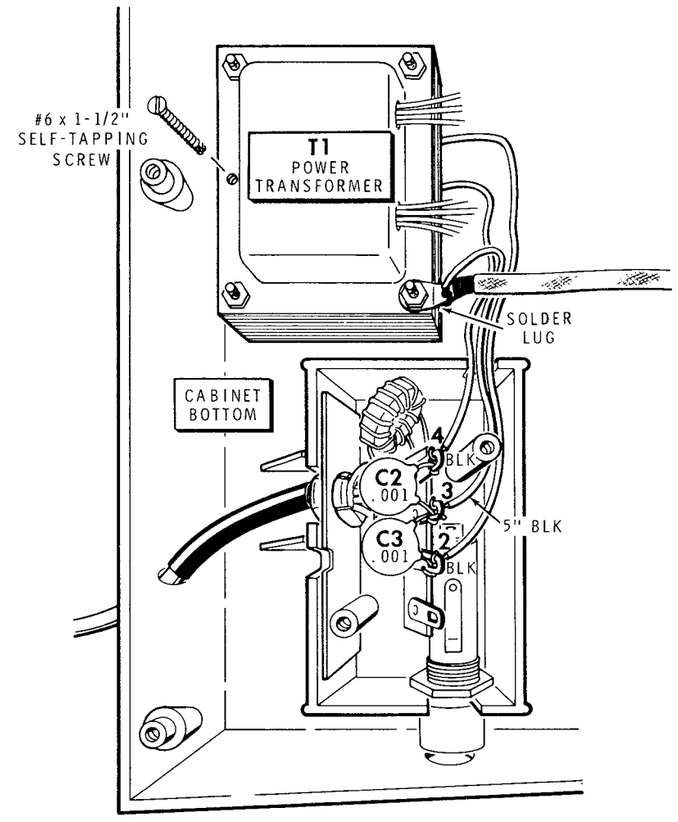
PICTORIAL 4-3



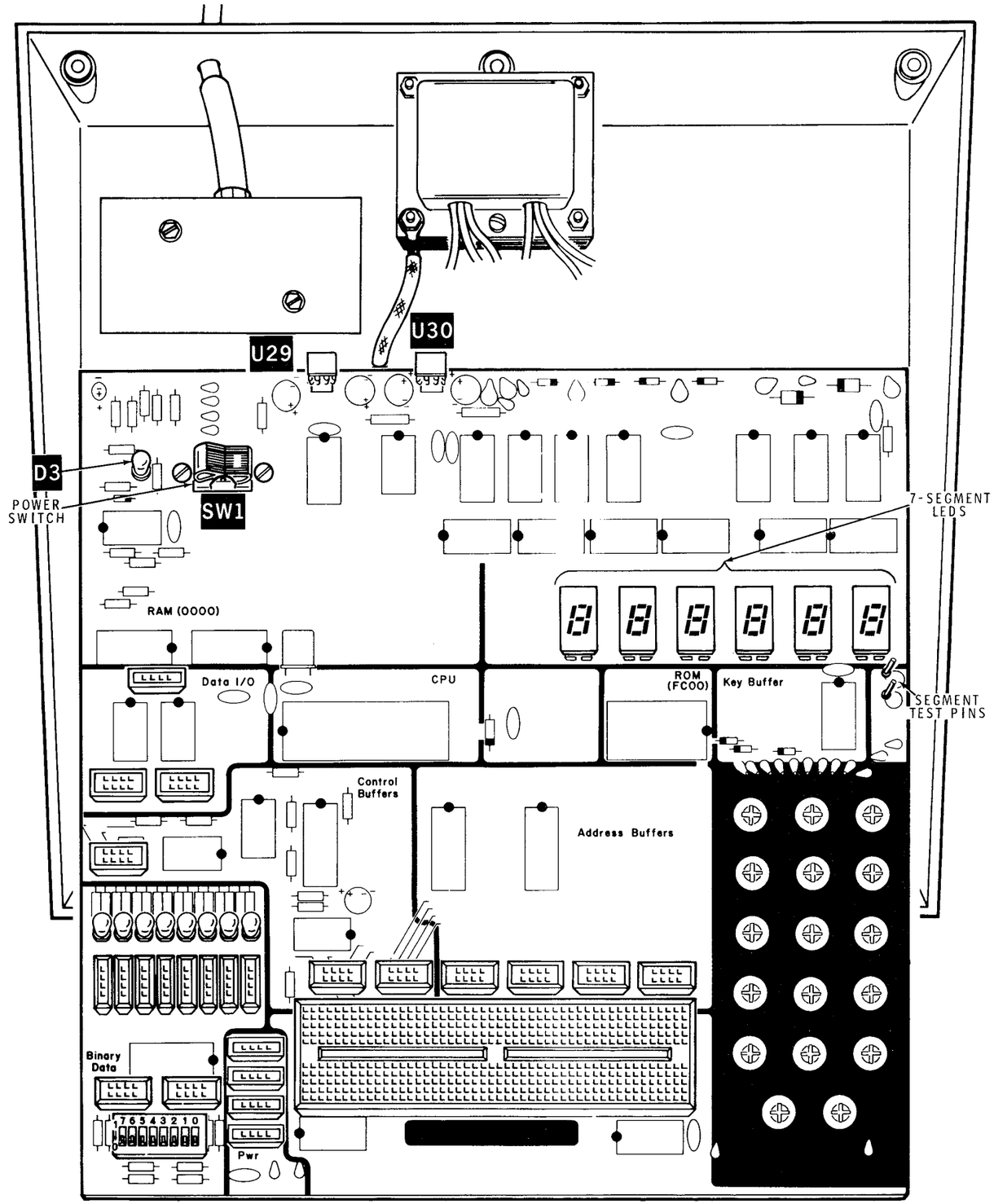
PICTORIAL 5-1



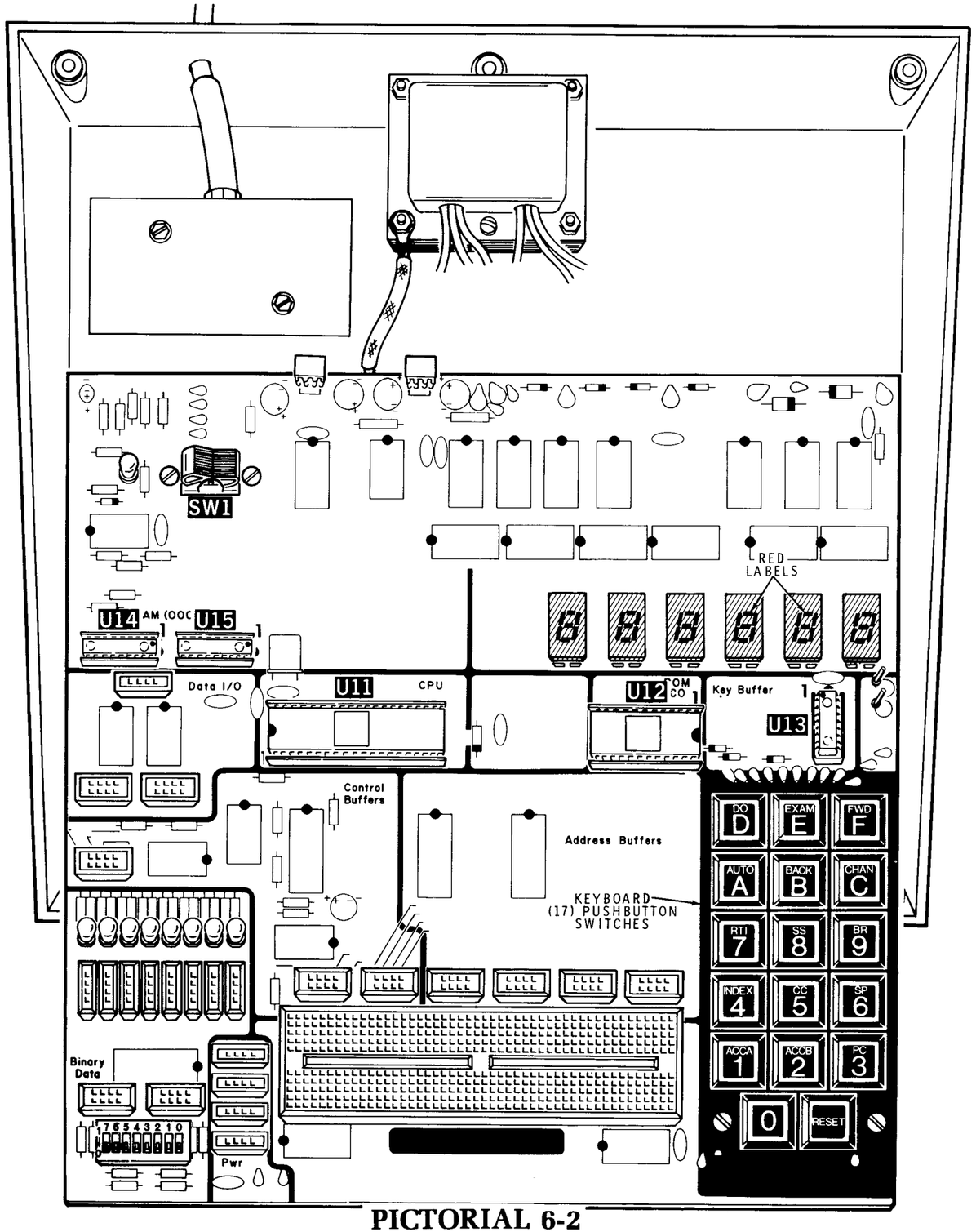
PICTORIAL 5-2



PICTORIAL 5-3

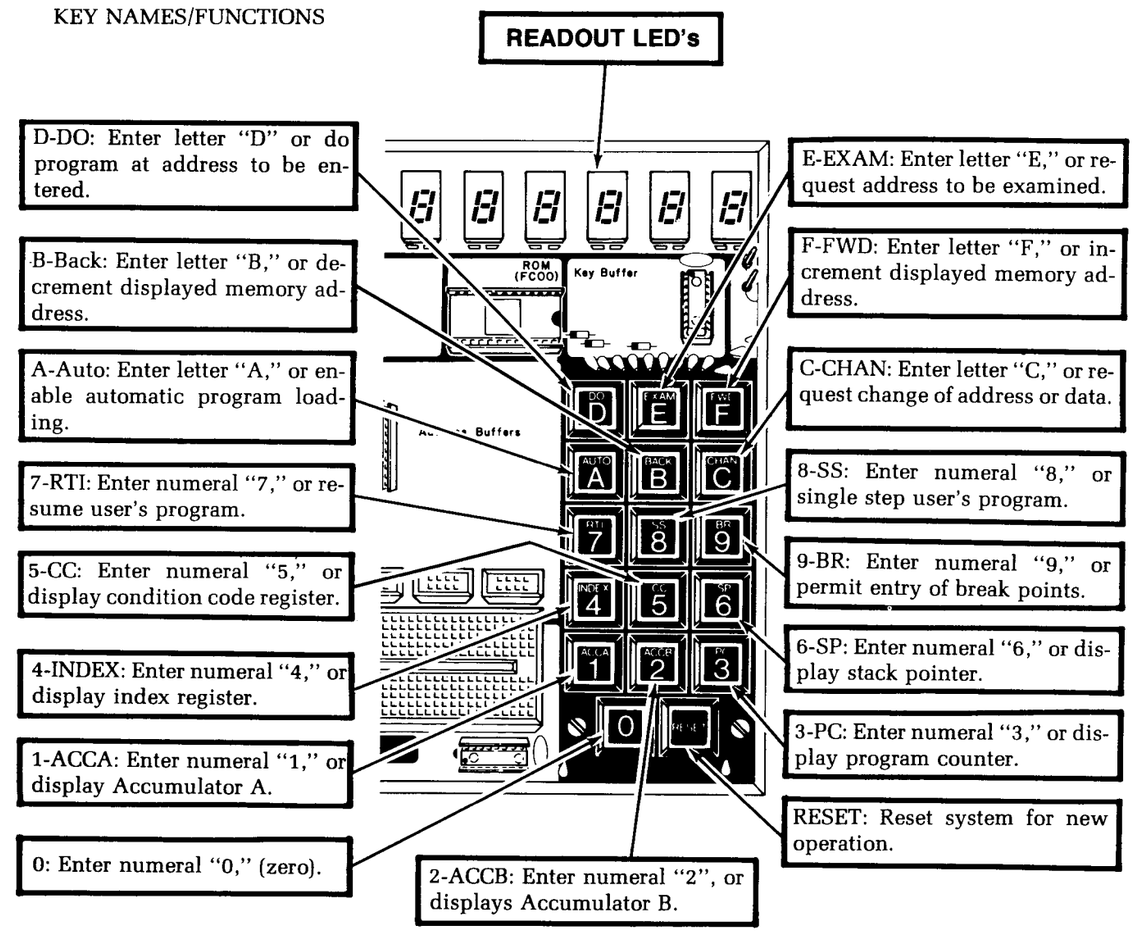


PICTORIAL 6-1

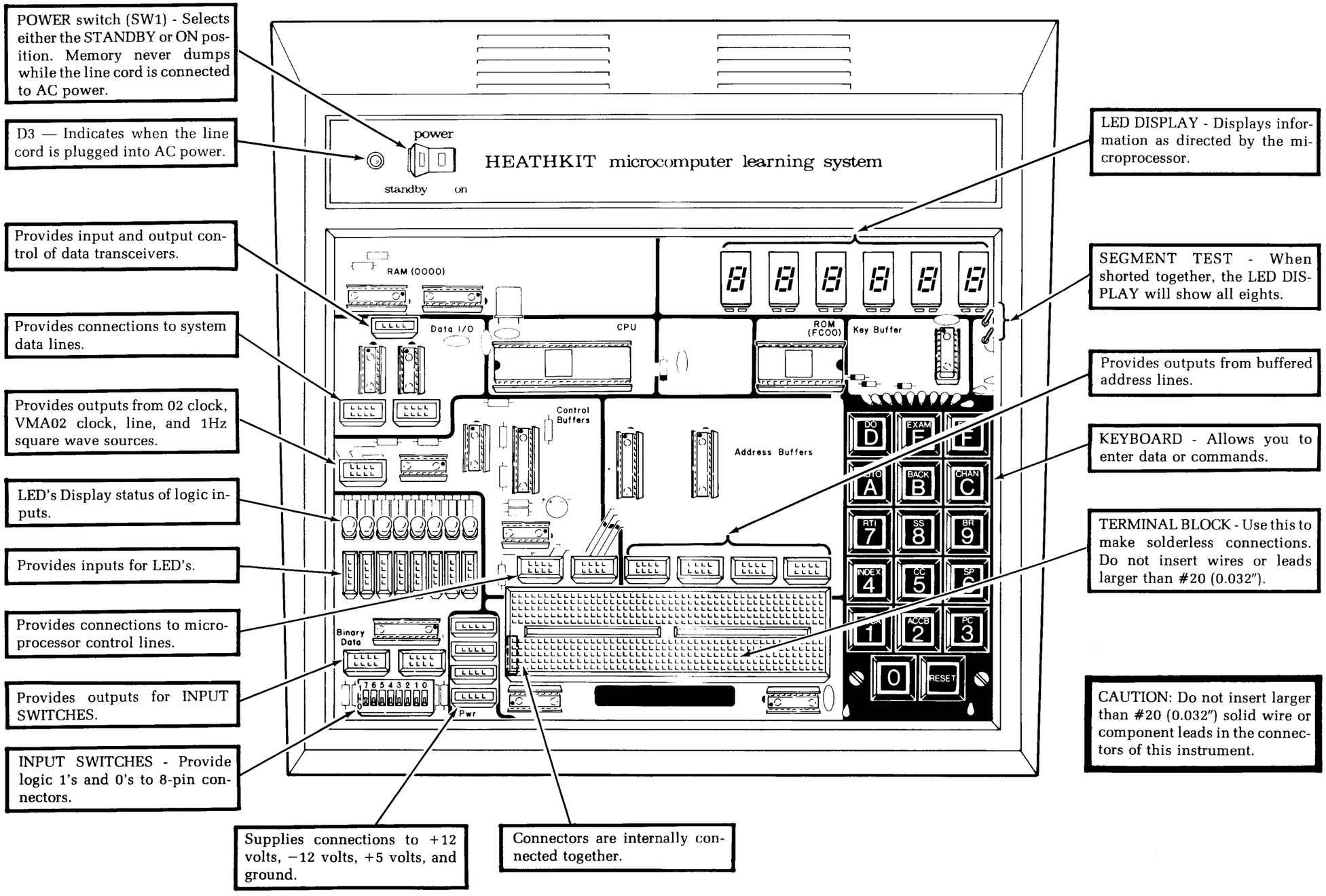


PICTORIAL 6-2

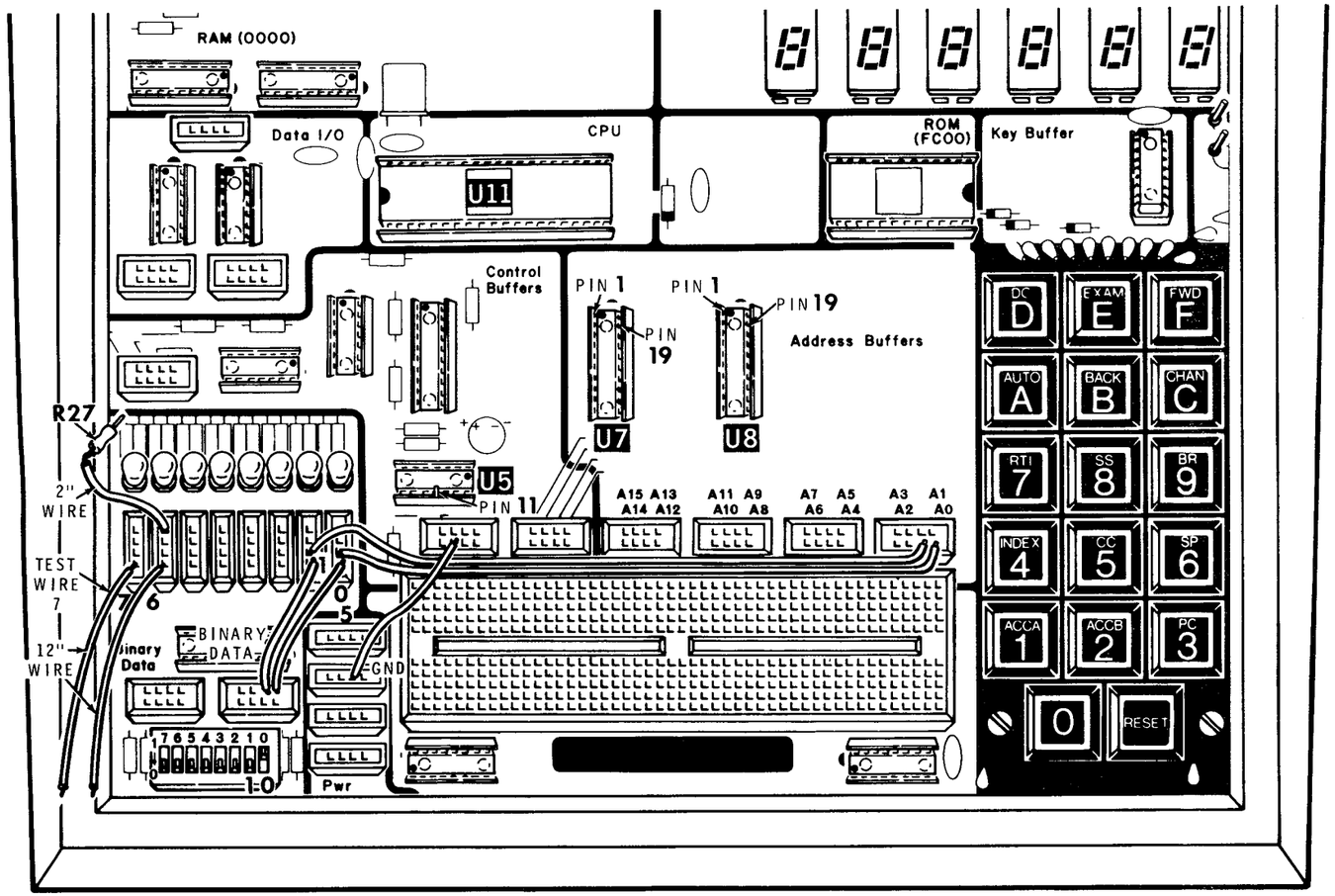
KEY NAMES/FUNCTIONS



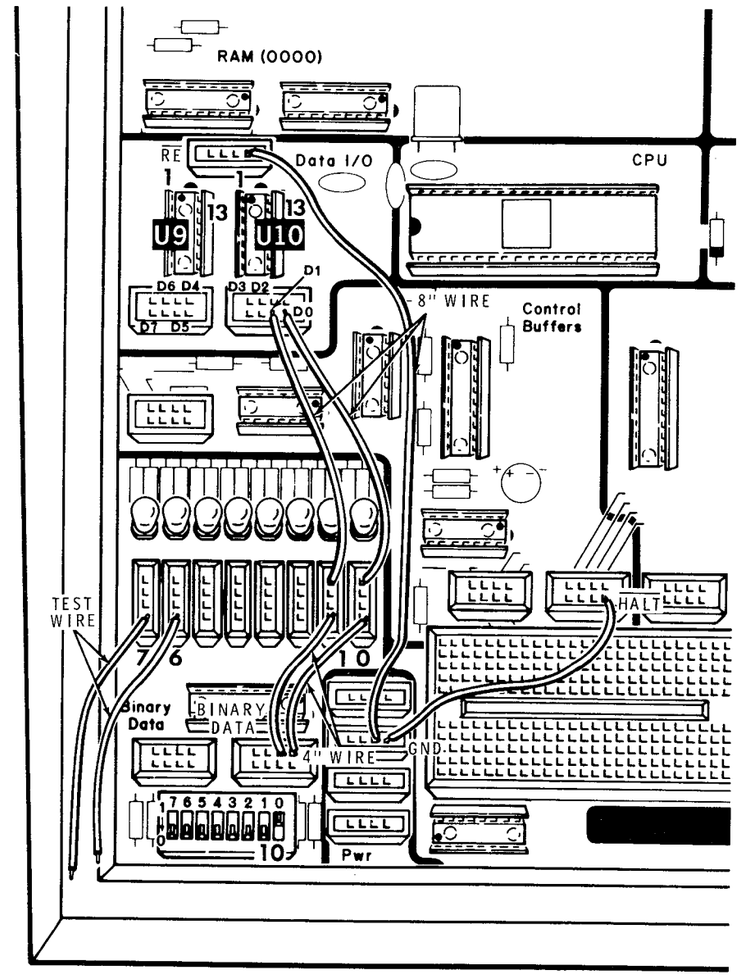
PICTORIAL 6-3



PICTORIAL 8-1



PICTORIAL 9-5



PICTORIAL 9-6

MAIN DECODING CHART

	U2							U3						U21 PIN					IC20								END RESULT
	INPUT PIN				OUTPUT PIN			INPUT PIN				OUTPUT PIN		9,	8,	12,	11,		INPUT PIN				OUTPUT PIN				
	12	13	14	15	1	7	9	12	13	14	15	1	5	10	2	13	1	3	12	13	14	15	5	6	10	11	
RAM 00XX	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	1	0	1	0	1	0	0	0	1	1	1	U14 and U15, Pin 8 is 0.
RAM 01XX	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	1	0	1	0	1	0	1	1	0	1	1	U14 and U15, Pin 8 is 0.
FFXX ROM FFXX	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	1	0	0	1	1	1	1	U12 pins 19 and 21 are 1. Pins 18 and 20 are 0.
KEYBOARD C0-X	0	1	1	0	1	0	1	0	1	0	0	1	0	0	1	0	1	0	1	0	0	0	1	1	0	1	U13 pins 1 and 15 are 0.
DISPLAY LED's C1XX	0	1	1	0	1	0	1	0	1	0	0	1	0	0	1	0	1	0	1	0	0	1	1	1	1	0	U22 pin 12 is 0.

0 = LOGIC 0

1 = LOGIC 1

– = DOES NOT CARE

X = FUNCTIONING ADDRESS

KEYBOARD COLUMN ADDRESS DECODING CHART

COLUMN ADDRESS	X				PRESS KEY	LOGIC 0 ON DATA LINE
	A_3	A_2	A_1	A_0		
CO – 3	—	0	1	1	F	D_0
					C	D_1
					9	D_2
					6	D_3
					3	D_4
CO – 5	—	1	0	1	E	D_0
					B	D_1
					8	D_2
					5	D_3
					2	D_4
CO – 6	—	1	1	0	D	D_0
					A	D_1
					7	D_2
					4	D_3
					1	D_4
					0	D_5

DISPLAY LED CHART

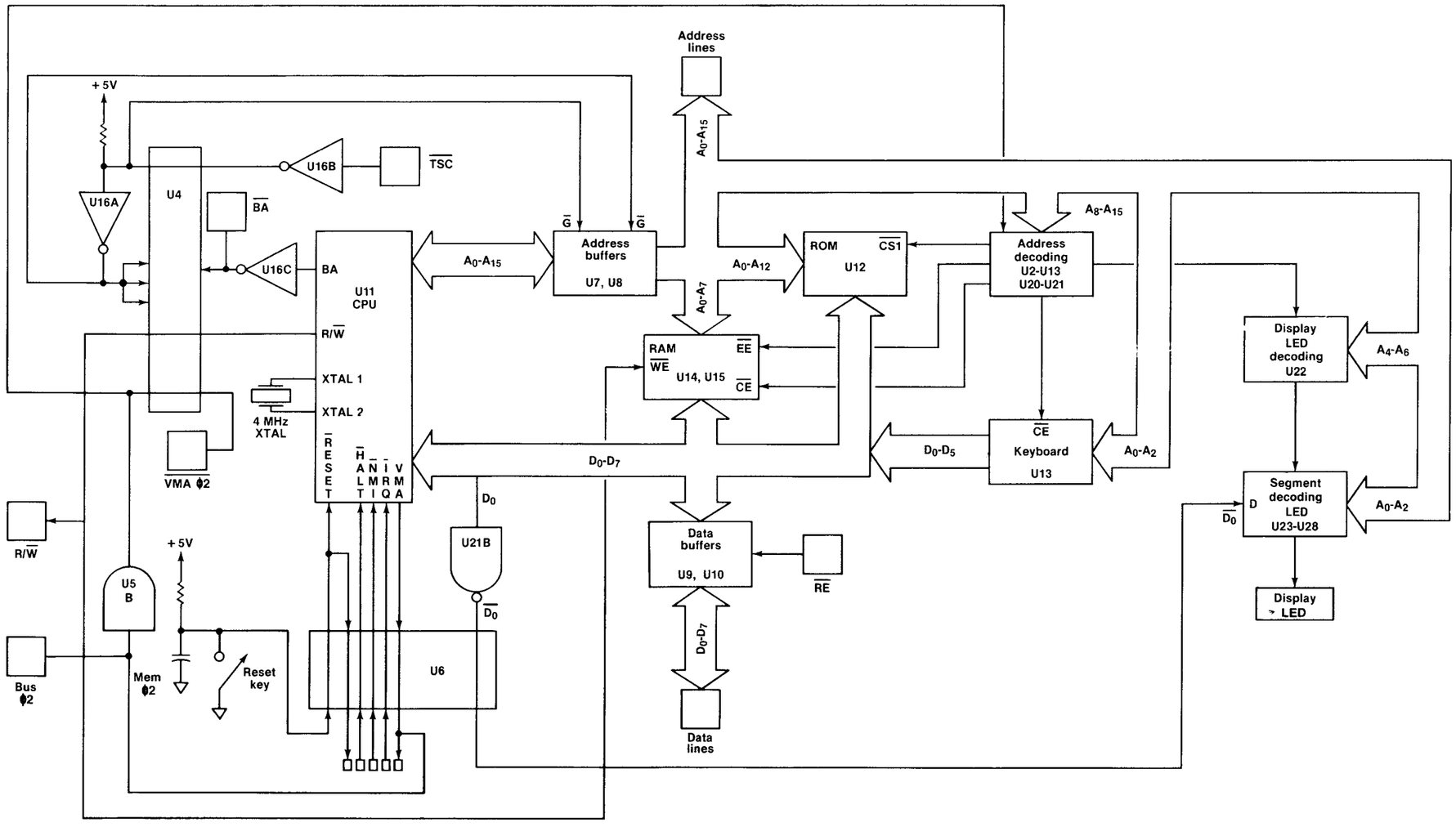
	X				U22 Input Pin				U22 Logic 0 at output pin	Logic 0 on enable pin 14 of IC	Display LED addressed
	A_7	A_6	A_5	A_4	12	13	14	15			
LED C1	—	1	1	0	0	1	1	0	7	23	H
		1	0	1	0	1	0	1	6	24	I
	—	1	0	0	0	1	0	0	5	25	N
	—	0	1	1	0	0	1	1	4	26	Z
	—	0	1	0	0	0	1	0	3	27	V
	—	0	0	1	0	0	0	1	2	28	C

	X				U23 through U28, input pins			U23 through U28 Output pin	LED pin	Segment*
	A_3	A_2	A_1	A_0	3,	2,	1			
LED segment C1X	1	1	1	0	1	1	0	11	1	a
	1	1	0	1	1	0	1	10	13	b
	1	1	0	0	1	0	0	9	10	c
	1	0	1	1	0	1	1	7	8	d
	1	0	1	0	0	1	0	6	7	e
	1	0	0	1	0	0	1	5	2	f
	1	0	0	0	0	0	0	4	11	g
	1	1	1	1	1	1	1	12	9	DP

*With a given output pin addressed, the logic level on that pin will follow the level on the D input of the addressed IC.

D_0 LOGIC LEVEL CHART

		U21 Input Pins	U21 Output	U6 Input	U6 Output	U23 through U28
D_0 logic levels	D_0	4 and 5	Pin 6	Pin 2	Pin 18	Pin 13
	1	1	0	0	0	0



BLOCK DIAGRAM

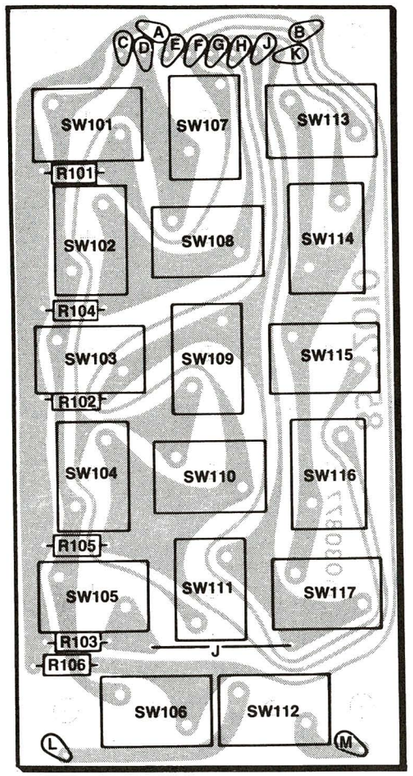
CIRCUIT BOARD X-RAY VIEW

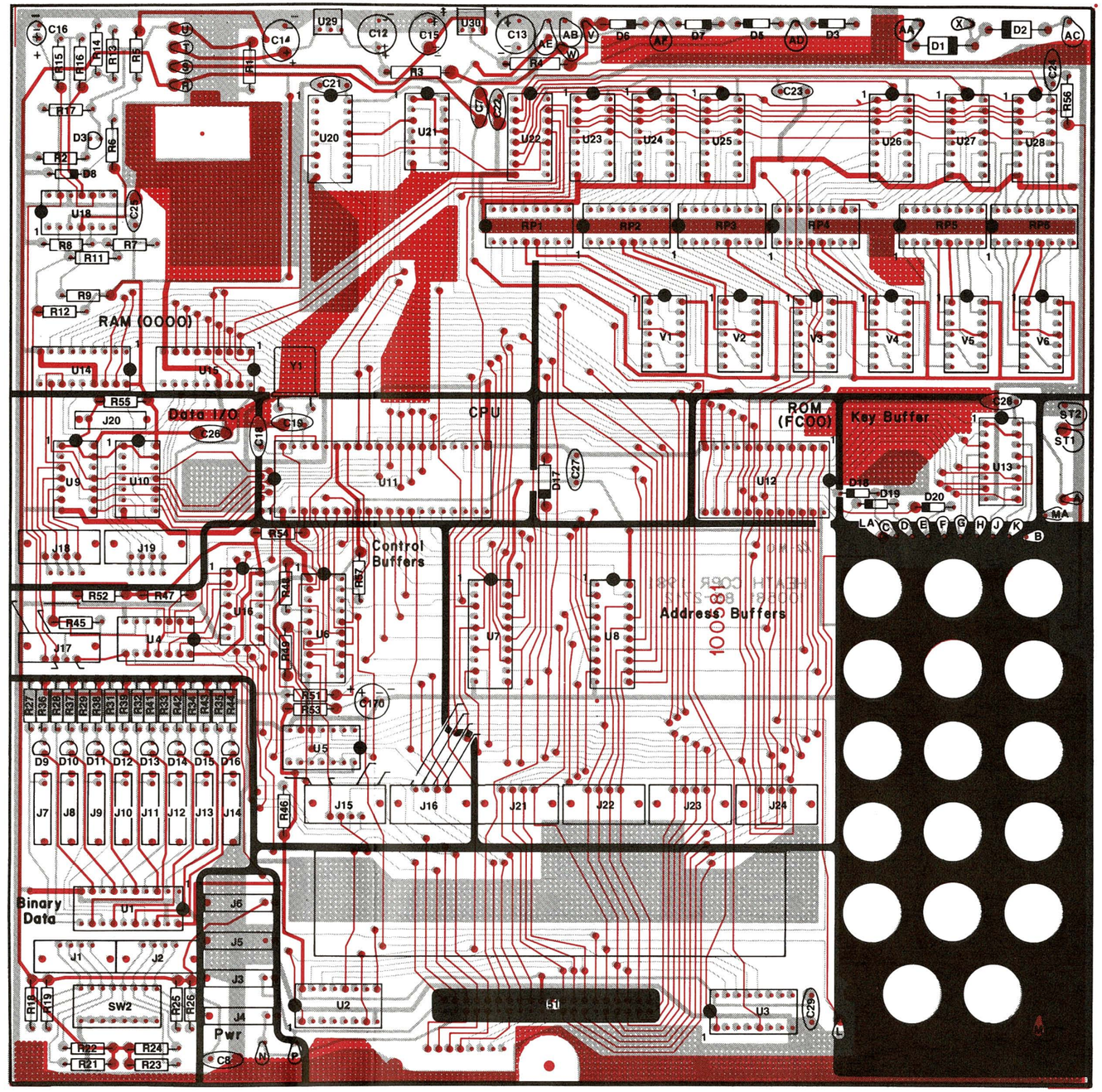
NOTE: To find the PART NUMBER of a component for the purpose of ordering a replacement part:

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)