mits Altair Floppy Disk

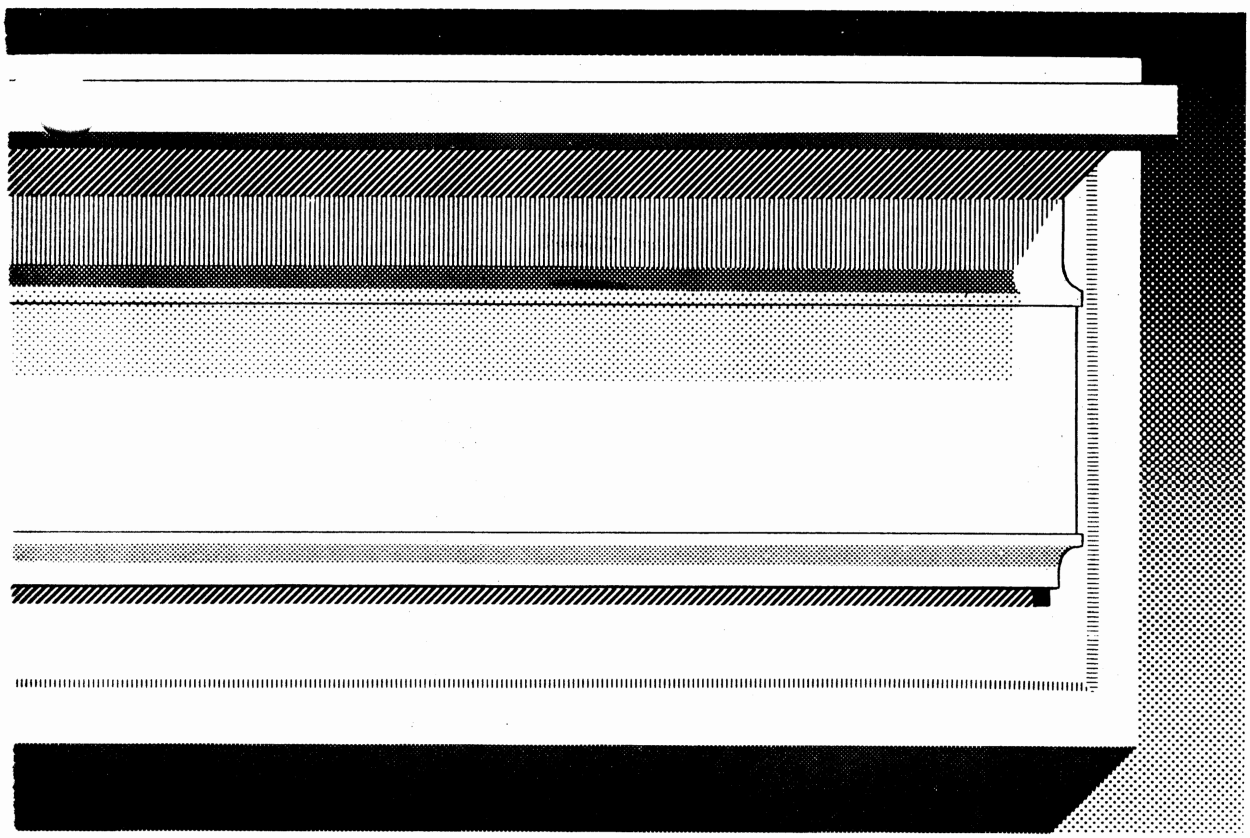
MITS, Inc. 1976

PRELIMINARY DOCUMENTATION RELEASE

This manual is incomplete in its present form. This page and an additional section will be sent to you within a short period for insertion.

This documentation contains the entire assembly and check-out information for both the disk controller and drive units. The Theory of Operation and some additional information will be in the insertation.

mits Altair Floppy Disk



drive & controller - hardware documentation

*********** ERRATA SHEET ***********

***** ALTAIR FLOPPY DISK *****

DRIVE & CONTROLLER - HARDWARE DOCUMENTATION

March 1976

THE FOLLOWING CHANGES HAVE BEEN MADE TO THE ABOVE TITLED MANUAL ON THE PAGES INDICATED.

PAGE 32: No heat-sink should be used for either X1 or X3. Mount both of these regulators directly to the board.

PAGE 95: Use a length of wire and connect the two pads labeled +8V together along the bottom edge of the board.

IT IS GENERALLY A GOOD IDEA TO GO THROUGH YOUR MANUAL AND MARK THESE CHANGES ON THE PAGES INDICATED BEFORE BEGINNING THE ACTUAL PROCESS OF ASSEMBLING YOUR UNIT.

MITS, Inc.
3/16/76

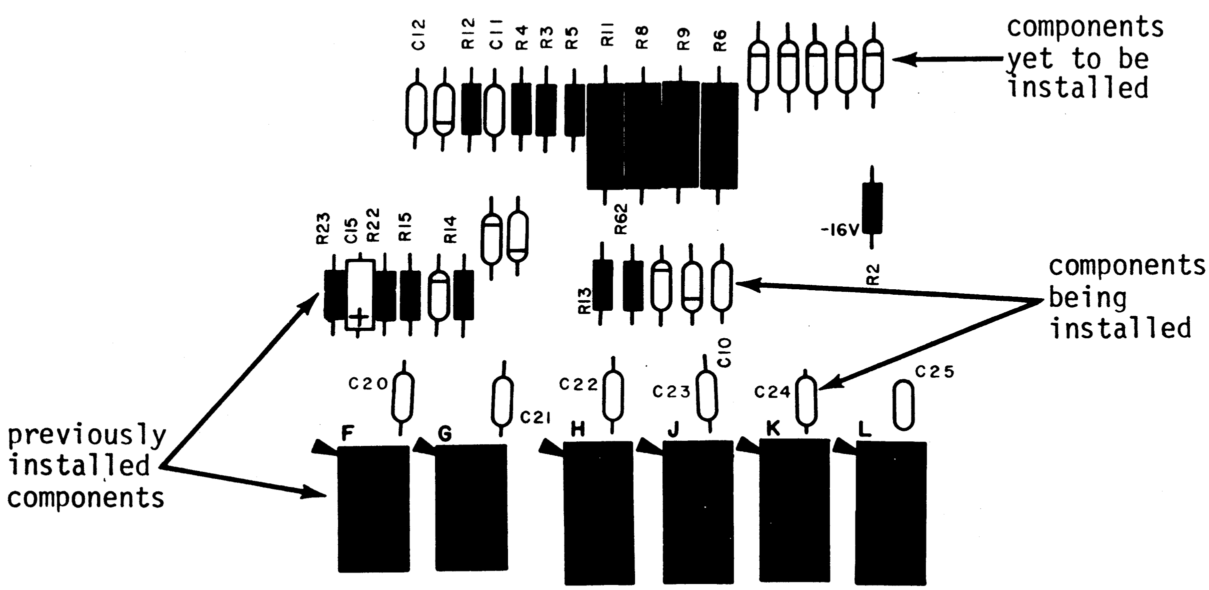
ASSEMBLY HINTS

Before beginning the construction of your unit, it is important that you read the "MITS Kits Assembly Hints" booklet included with your kit. Pay particular attention to the section on soldering, because most problems in the Altair occur as the result of poor soldering. It is essential that you use the correct type of soldering iron. A 25-30 watt iron with a chisel tip (such as an Ungar 776 with a 7155 tip) is recommended in the assembly hints booklet.

Some important warnings are also included in the hints booklet. Read them carefully before you begin work on your unit -- failure to heed these warnings could cause you to void your warranty.

Check the contents of your kit against the enclosed parts list to make sure you have all the required components, hardware and parts. The components are in plastic envelopes; do not open them until you need the components for an assembly step. You will need the tools called for in the "Kits Assembly Hints" booklet.

As you construct your kit, follow the instructions in the order they are presented in the assembly manual. Always complete each section before going on to the next. Two organizational aids are provided throughout the manual to assist you: 1) Boxed-off parts identification lists, with spaces provided to check off the components as they are installed; 2) Reproductions of the silk screens showing a) previously installed components, b) components being installed and c) components yet to be installed. (see below)



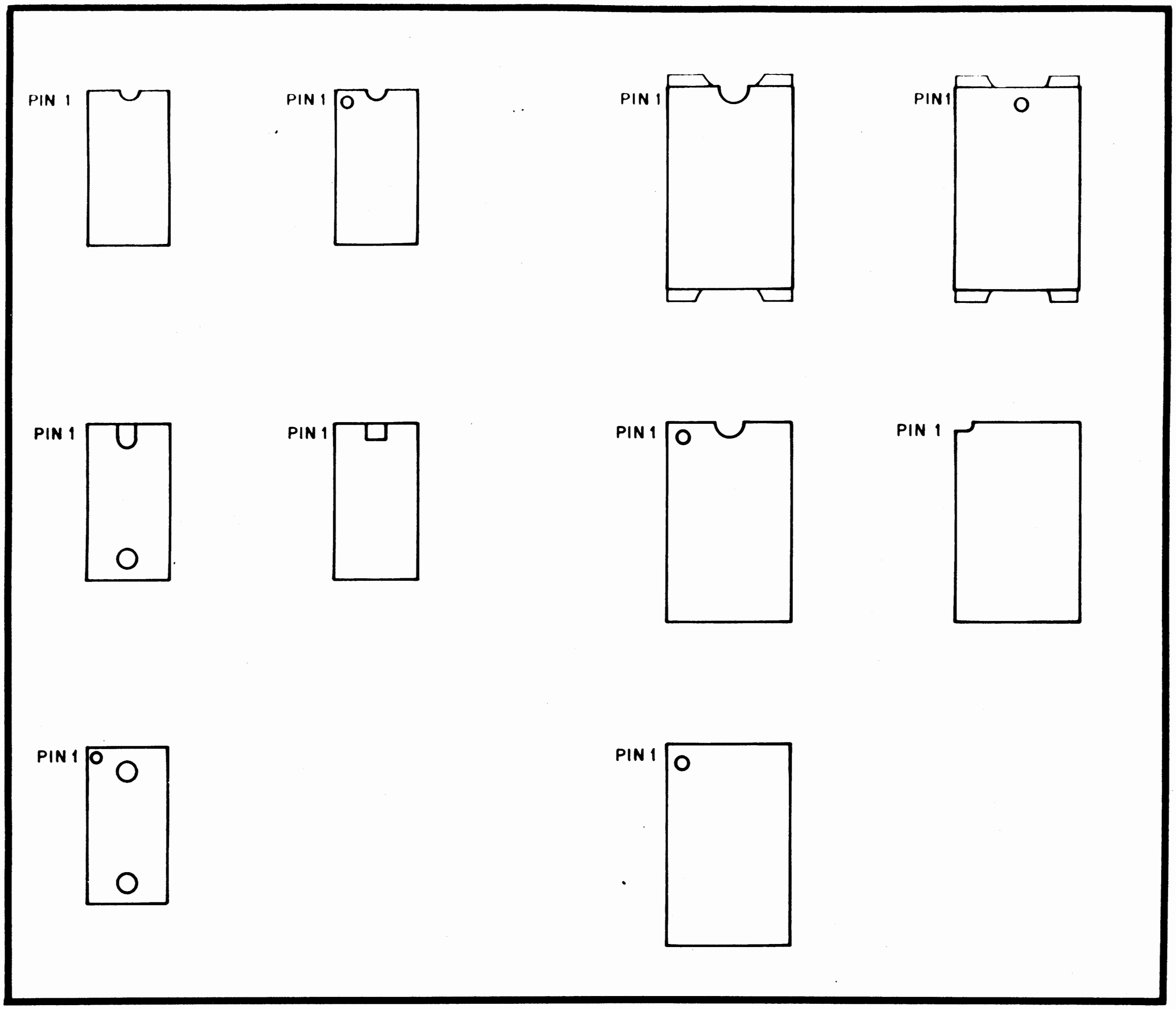
COMPONENT INSTALLATION METHODS

This section of the manual describes the proper procedures for installing various types of components in your kit.

Read these instructions over very carefully and refer back to them whenever necessary. Failure to properly install components may cause permanent damage to the component or the rest of the unit; it will definitely void your warranty.

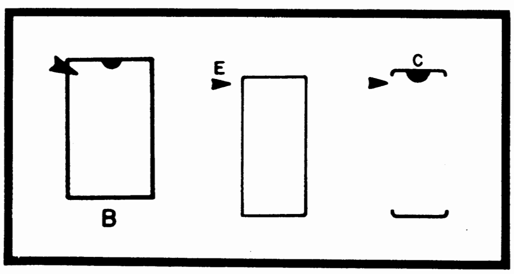
More specific instructions, or procedures of a less general nature, will be included within the assembly text itself.

Under no circumstances should you proceed with an assembly step without fully understanding the procedures involved. A little patience at this stage will save a great deal of time and potential "headaches" later.



INTEGRATED CIRCUITS (IC's) CAN COME WITH ANY ONE OF, OR A COMBINATION OF, SEVERAL DIFFERENT MARKINGS. THESE MARKINGS ARE VERY IMPORTANT IN DETERMINING THE CORRECT ORIENTATION FOR THE IC's WHEN THEY ARE PLACED ON THE PRINTED CIRCUIT BOARDS. REFER TO THE ABOVE DRAWING TO LOCATE PIN 1 OF THE IC's, THEN USE THIS INFORMATION IN CONJUNCTION WITH THE INFORMATION BELOW TO PROPERLY ORIENT EACH IC FOR INSTALLATION.

WARNING: INCORRECTLY ORIENTED IC's MAY CAUSE PERMANENT DAMAGE!



THE DRAWING ON THE LEFT INDICATES VARIOUS METHODS USED TO SHOW THE POSITION OF IC's ON THE PRINTED CIRCUIT BOARDS. THESE ARE SILK-SCREENED DIRECTLY ON THE BOARD. THE ARROWHEAD INDICATES THE POSITION FOR PIN 1 WHEN THE IC IS INSTALLED.

IC Installation

All ICs must be oriented so that the notched end is toward the end with the arrowhead printed on the PC board. Pin 1 of the IC should correspond with the pad marked with the arrowhead. If the IC does not have a notch on one end, refer to the chart on the preceeding page for the identification of Pin 1.

To prepare ICs for installation:
All ICs are damaged easily and should be handled carefully — especially static-sensitive MOS ICs. Always try to hold the IC by the ends, touching the pins as little as possible.

When you remove the IC from its holder, CAREFULLY straighten any bent pins using needle-nose pliers. All pins should be evenly spaced and should be aligned in a straight line, perpendicular to the body of the IC itself.

1. Orient the IC so that Pin 1 coincides with the arrowhead on the PC board.

2. Align the pins on one side of the IC so that just the tips are inserted into the proper holes on the board.

3. Lower the other side of the IC into place. If the pins don't go into their holes right away, rock the IC back, exerting a little inward pressure, and try again. Be patient. The tip of a small screwdriver may be used to help guide the pins into place. When the tips of all the pins have been started into their holes, push the IC into the board the rest of the way.

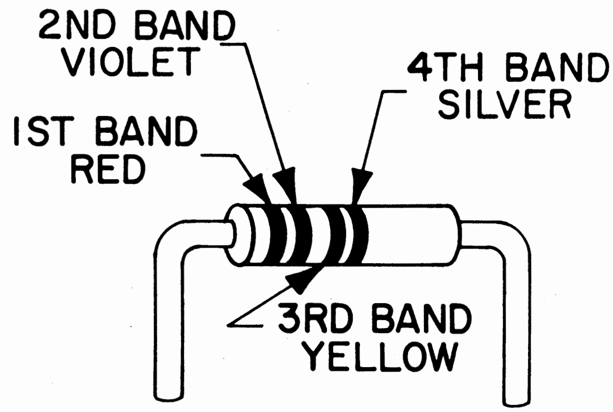
4. Tape the IC into place on the board with a piece of masking tape.

5. Turn the board over and solder each pin to the foil pattern on the back side of the board. Be sure to solder each pin and be careful not to leave any solder bridges.

6. Turn the board over again and remove the piece of masking tape.

Resistors have four (or possibly five) color-coded bands as represented in the chart below. The fourth band is gold or silver and indicates the tolerance. NOTE: In assembling a MITS kit, you need only be concerned with the three bands of color to the one side of the gold or silver (tolerance) band. These three bands denote the resistor's value in ohms. The first two bands correspond to the first two digits of the resistor's value and the third band represents a multiplier.

For example: a resistor with red, violet, yellow and silver bands has a value of 270,000 ohms and a tolerance of 10%. By looking at the chart below, you see that red is 2 and violet 7. By multiplying 27 by the yellow multiplier band (10,000), you find you have a 270,000 ohm (270K) resistor. The silver band denotes the 10% tolerance. Use this process to chose the correct resistor called for in the manual.



RESISTOR COLOR CODES

COLOR	BANDS 1&2	3rd BAND (Multiplier)
Black	0	1
Brown	1	10
Red	2	10^2
Orange	3	10^3
Yellow	4	10^4
Green	5	10^5
Blue	6	10^6
Violet	7	10^7
Gray	8	10^8
White	9	10^9

Use the following procedure to install the resistors onto the boards. Make sure the colored bands on each resistor match the colors called for in the list of Resistor Values and Color Codes given for each board.

1. Using needle-nose pliers, bend the leads of the resistor at right angles to match their respective holes on the PC board.

2. Install the resistor into the correct holes on the silk-screened side of the PC board.

3. Holding the resistor in place with one hand, turn the board over and bend the two leads slightly outward.

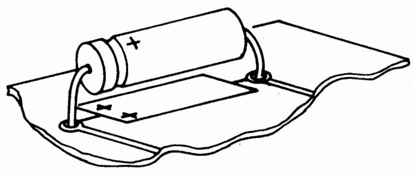
4. Solder the leads to the foil pattern on the back side of the board; then clip off any excess lead lengths.

Capacitor Installation

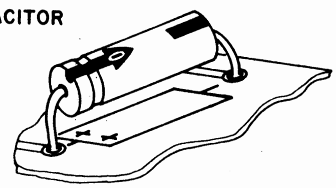
A. Electrolytic and Tantalum Capacitors

Polarity requirements must be noted on the electrolytic capacitors and the tantalum capacitor before they are installed.

The electrolytic capacitors contained in your kit may have one or possibly two of three types of polarity markings. To determine the correct orientation, look for the following.



ELECTROLYTIC CAPACITOR



One type will have plus (+) signs on the positive end; another will have a band or a groove around the positive side in addition to the plus signs. The third type will have an arrow on it; in the tip of the arrow there is a negative (-) sign and the capacitor must be oriented so the arrow points to the negative polarity side.

The tantalum capacitor is metallic in appearance and smaller than the electrolytic capacitors. Its positive end has a plus sign on it or a red dot.

Refer to the chart included for each board for correct Capacitor Values and install the electrolytic capacitors and tantalum capacitors using the following procedure.

1. Bend the two leads of the capacitor at right angles to match their respective holes on the board. Insert the capacitor into the holes on the silk-screened side of the board. Be sure to align the positive polarity side with the "+" signs printed on the board.

2. Holding the capacitor in place, turn the board over and bend the two leads slightly outward. Solder the leads to the foil pattern and clip off any excess lead lengths.

B. Ceramic Disk Capacitors

Refer to the chart included for each board for correct Capacitor Values, and install the ceramic disk capacitors using the following procedure.

1. Choose the correct value capacitor and straighten the two leads as necessary to fit their respective holes on the PC board.

2. Insert the capacitor into the correct holes from the silk-screened side of the board. Push the capacitor down until the ceramic insulation almost touches the foil pattern.

3. Holding the capacitor in place, turn the board over and bend the two leads slightly outward.

4. Solder the two leads to the foil pattern on the back side of the board; then clip off any excess lead lengths.

Transistor Installation

To install transistors, use the following instructions.

NOTE: Always check the part number of each transistor before you install it. (See listing of Transistor Part Numbers for each board.) Some transistors look identical but differ in electrical characteristics, according to part number. If you have received substitute part numbers for the transistors in you kit, check the Transistor Identification Chart which follows these instructions to be sure you make the correct substitutions.

NOTE: Always make sure the transistor is oriented so that the emitter lead is installed in the hole on the PC board labeled with an "E." To determine which lead is the emitter lead, refer to the Transistor Identification Chart.

1. After the correct transistor has been selected and the leads have been properly oriented, insert the transistor into the holes on the silk-screened side of the board.

2. Holding the transistor in place, turn the board over and bend the three leads slightly outward.

3. Solder the leads to the foil pattern on the back side of the board; then clip off any excess lead lengths.

Diode Installation

NOTE: Diodes are marked with a band on one end indicating the cathode end. Each diode must be installed so that the end with the band is oriented towards the band printed on the PC board. Failure to orient the diodes correctly may result in permanent damage to your unit.

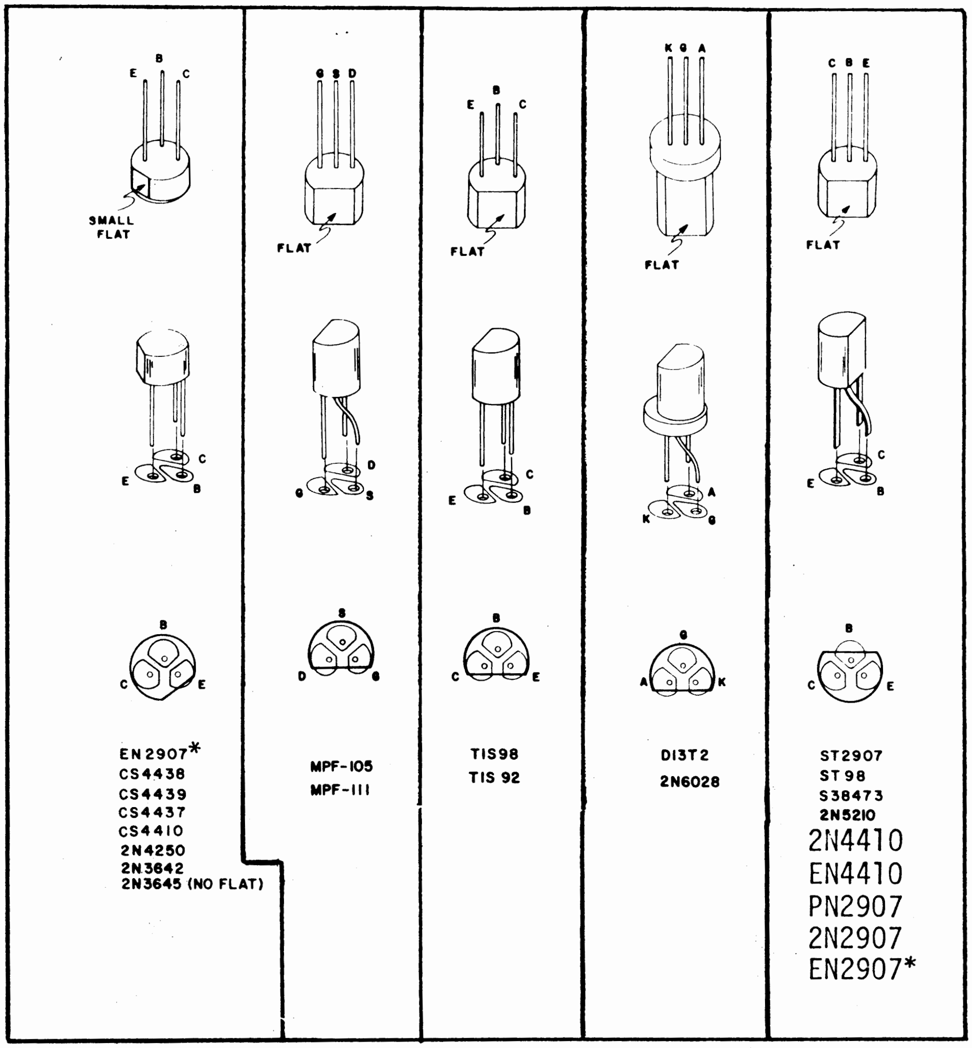
Use the following procedure to install diodes onto the board. Refer to the list of Diode Part Numbers included for each board to make sure you install the correct diode each time.

1. Bend the leads of the diode at right angles to match their respective holes on the board.

2. Insert the diode into the correct holes on the silk screen, making sure the cathode end is properly oriented. Turn the board over and bend the leads slightly outward.

3. Solder the two leads to the foil pattern on the back side of the board; then clip off any excess lead lengths.

TRANSISTOR IDENTIFICATION CHART



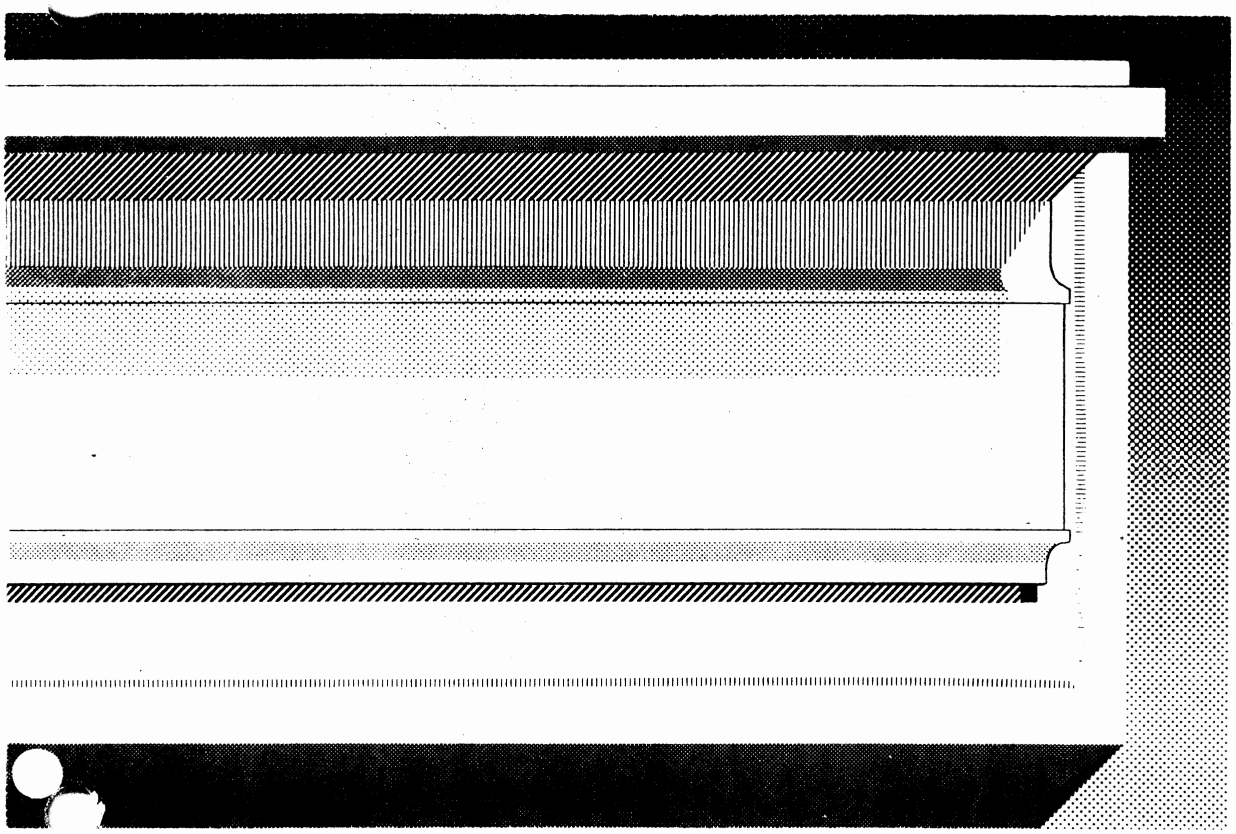
IN THE ILLUSTRATION ABOVE THE OUTLINE OF EACH TYPE OF TRANSISTOR IS SHOWN OVER THE PADS ON THE CIRCUIT BOARD WITH THE CORRECT DESIGNATION FOR EACH OF THE THREE LEADS. USE THIS INFORMATION TOGETHER WITH THE INFORMATION IN THE ASSEMBLY MANUAL FOR THE CORRECT ORIENTATION OF THE TRANSISTORS AS YOU INSTALL THEM.

THE FOLLOWING IS A LIST OF POSSIBLE SUBSTITUTIONS: IF ANY OTHERS ARE USED YOU WILL RISK DAMAGING YOUR UNIT:

2N4410 = EN4410 = CS4410 = CS4437, CS4438, TIS98, ST98, S38473 (NPN)
EN2907 = 2N2907 = PN2907 = ST2907, CS4439 (PNP)

WHEN MAKING SUBSTITUTIONS, REFER TO THE ILLUSTRATION TO DETERMINE THE CORRECT ORIENTATION FOR THE THREE LEADS.

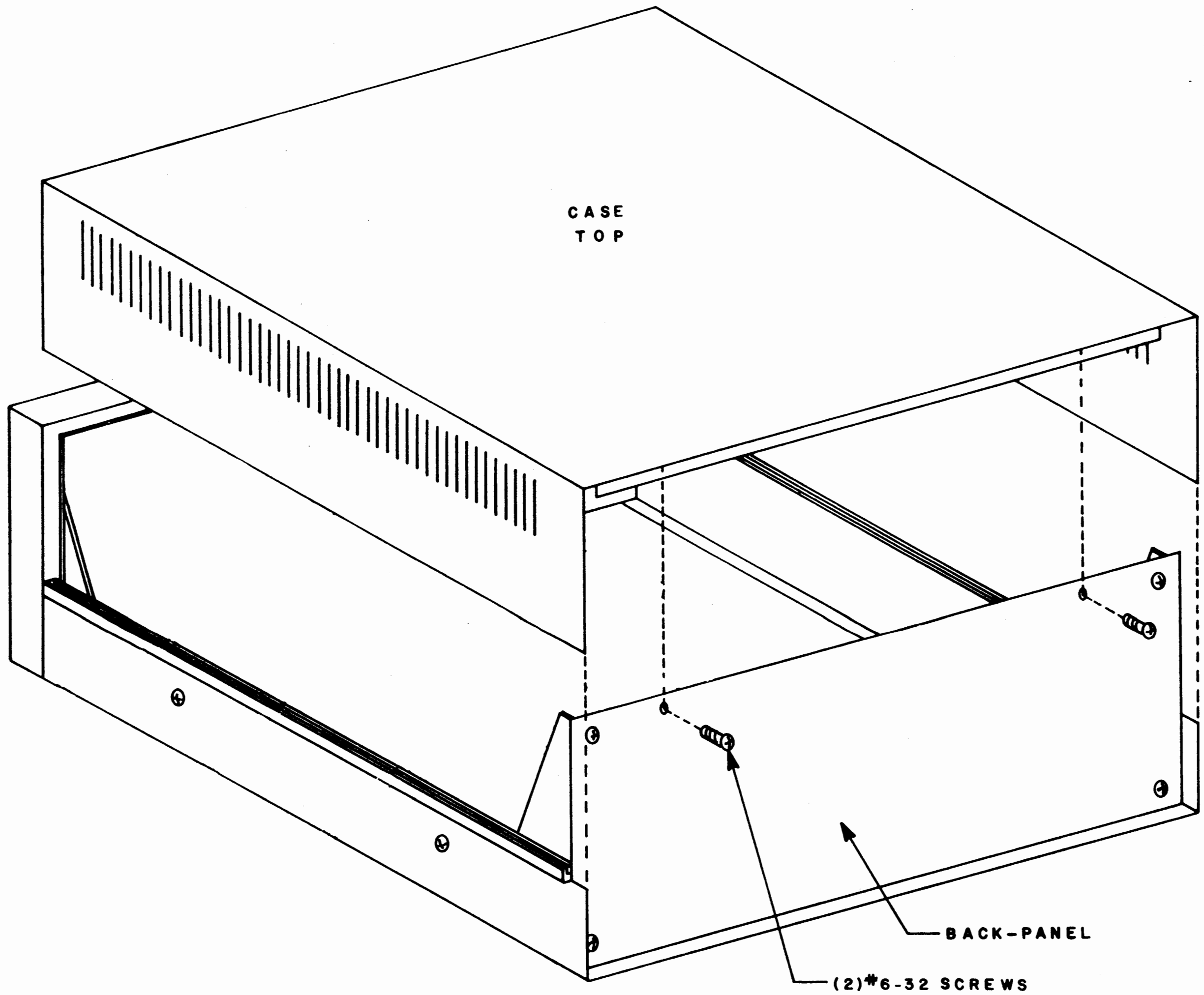
*Configuration of the leads on EN2907 may vary.



disk drive assembly procedure

Remove the top from the Disk Drive case by withdrawing the two screws indicated in the drawing below. Slide the case top backwards, lifting the back slightly, to remove it entirely from the chassis.

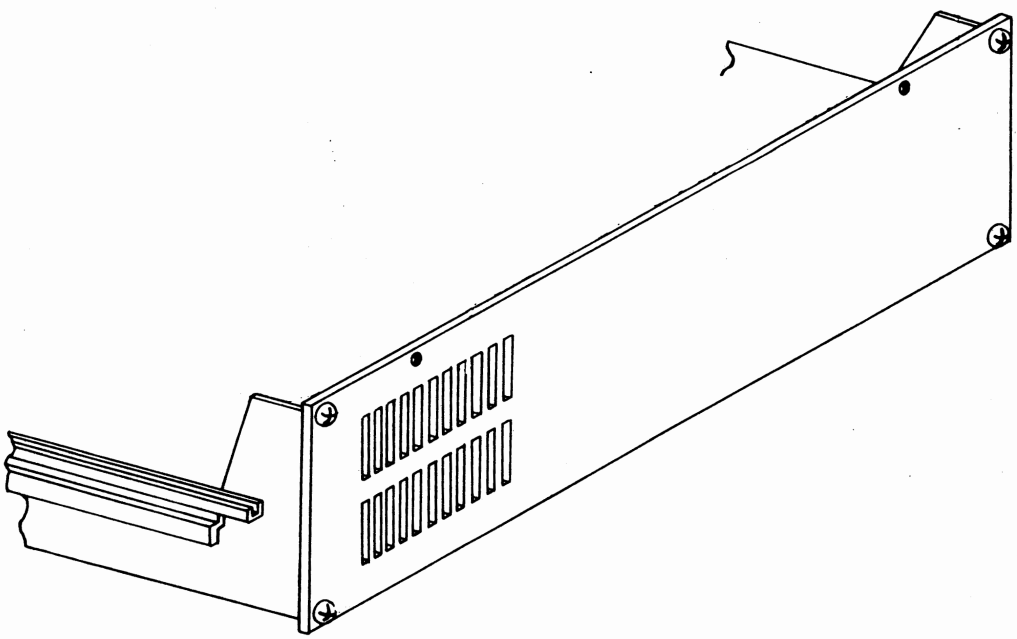
Also remove the 4 screws in the side of the case bottom, and remove the entire chassis assembly.



DISK DRIVE BACK PANEL ASSEMBLY

Remove the back panel from the case by withdrawing each of the four screws in the corners of the panel. These four screws are shown inserted in the drawing below.

Save these four screws for remounting the back panel later in the assembly procedure.

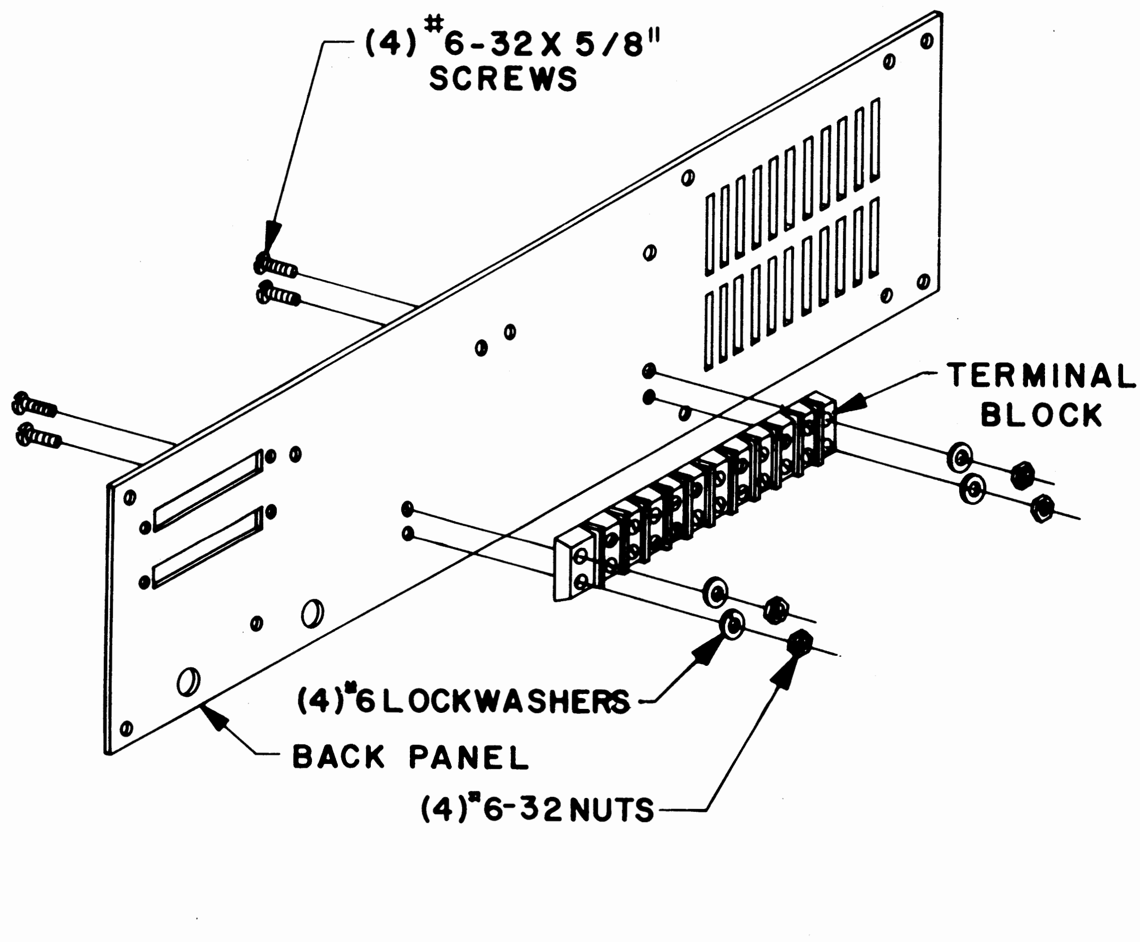


Terminal Block Installation

Mount the terminal block to the back panel as shown in the drawing below. Use the screw sizes and other hardware indicated in the drawing.

NOTE: Be sure that the back panel is oriented as shown; be careful not to mount the terminal block on the wrong side of the panel.

Tighten all four screws firmly into place.



Transformer Installation

There are two transformers included in this kit. The larger of the two will be refered to as T1, the smaller as T2.

Wire Preparation

Before mounting these transformers, the wires must be cut to the proper length and screw-mount crimp terminals attached to each of them. There are also three wires which will not be used at all, and will be cut off at the transformer coil.

Refering to the drawing on the opposite page, cut the wires on transformers T1 and T2 to the lengths indicated. The three unused wires should be cut off at the point where they enter the transformer coil itself.

Next, as indicated in the bottom of the drawing, strip exactly 1/2" of insulation from each of the eleven wires and bend the exposed portion in half to 1/4".

There are several screw-mount crimp terminals included with this kit. These have a slot in one end and an insulated portion on the other end (usually red) for attaching wires. One of these crimp terminals must be attached to each of the eleven transformer wires.

Insert one of the wires into one of the terminals as shown in the drawing. Push the wire in as far as it will go without distorting it or pushing it all the way through.

The wire should then be permanently connected to the terminal by either soldering it in place or crimping. To crimp the terminal use a crimping tool, if available, or else flatten the insulated portion of the terminal as tightly as possible using pliers.

Prepare each of the eleven transformer wires in the above manner.

Mounting

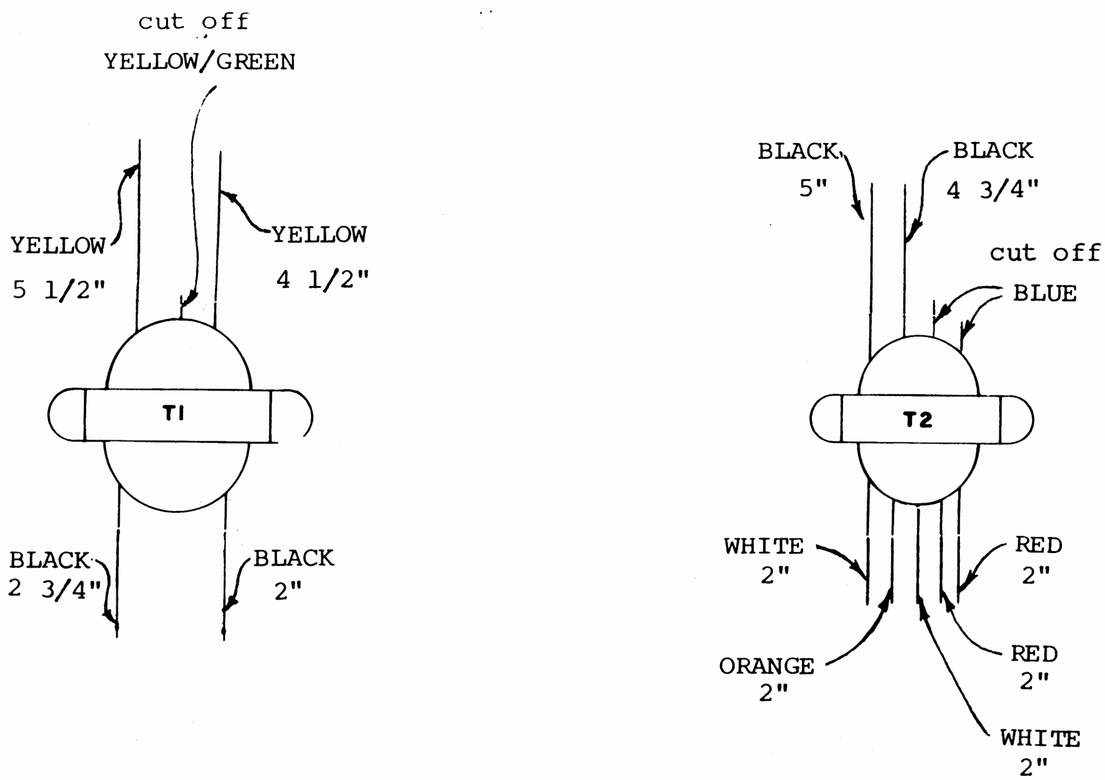
Refering to the drawings following the "Transformer Wire Preparation" drawing, mount transformers T1 & T2 to the back panel.

NOTE: For proper orientation, transformer T1 should have the two yellow wires towards the top of the panel (with reference to the drawings), and T2 should have the two black wires towards the top of the panel.

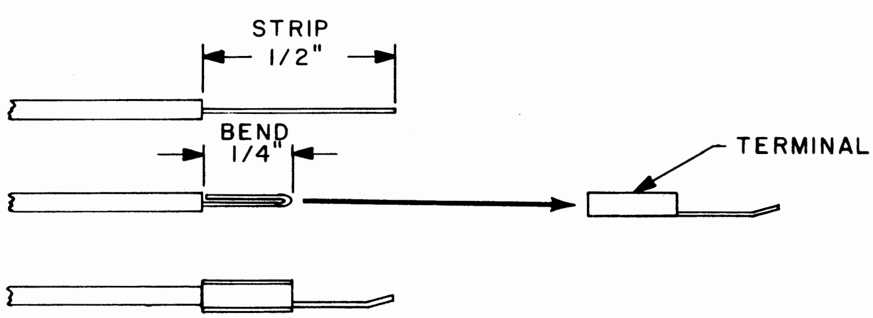
Be sure to install a terminal lug on transformer T1 as shown in the drawing. This is a solder type lug, and not the screw-mount type used for the transformer wires.

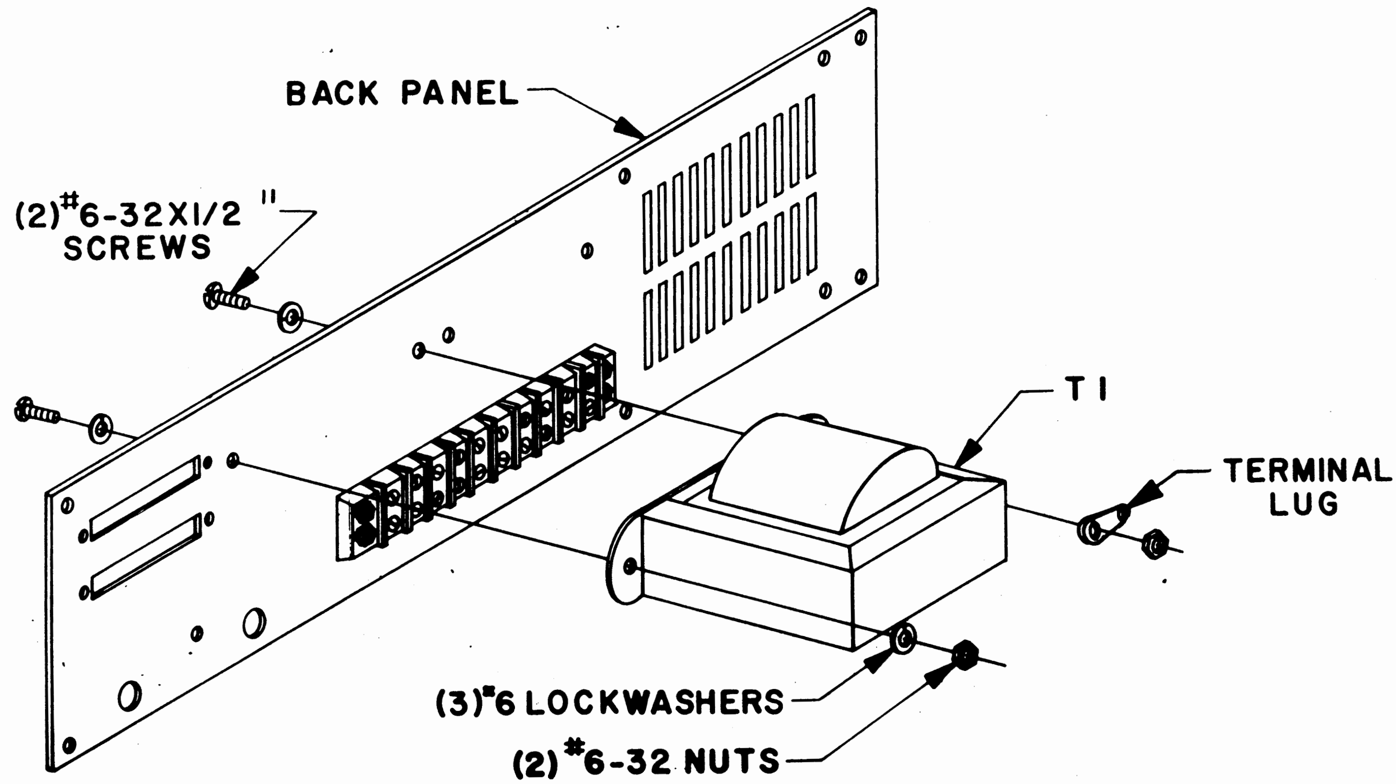
Use the hardware indicated in the drawings to mount the transformers and tighten the screws firmly into place.

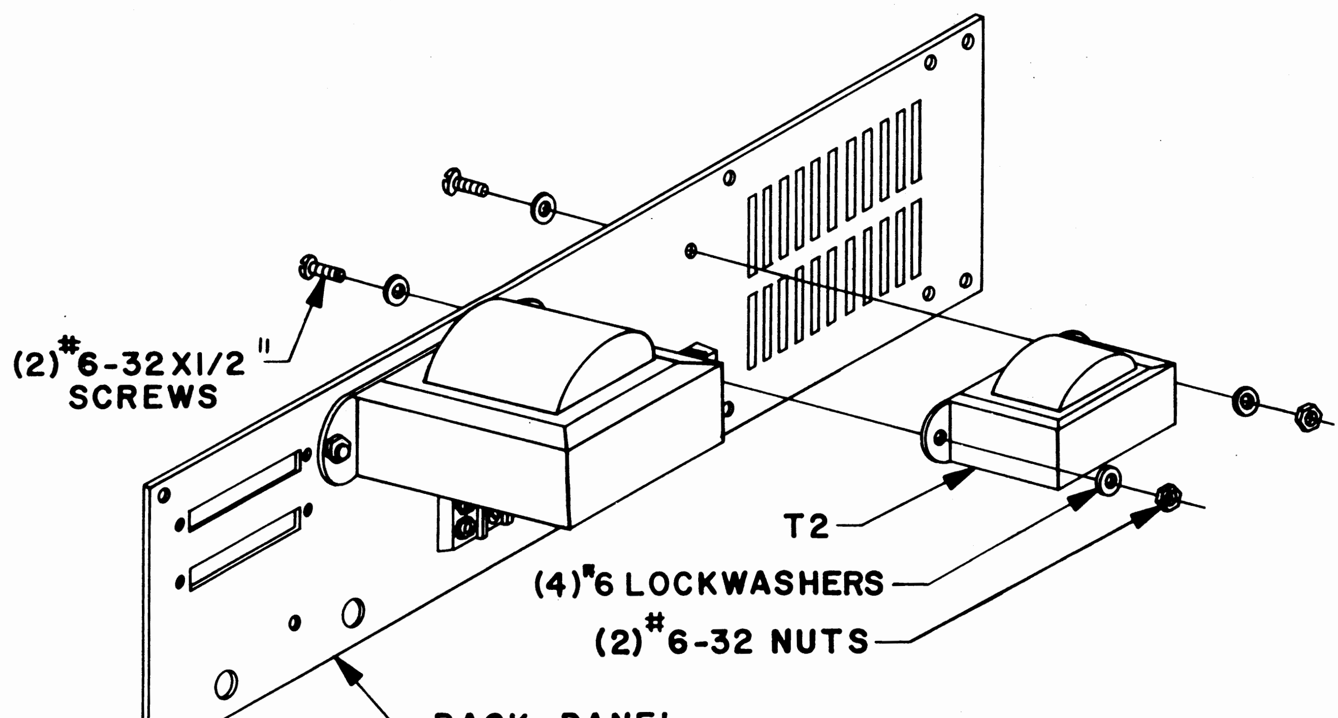
NOTE: Save all wires that you cut off for later use.



TRANSFORMER WIRE PREPARATION



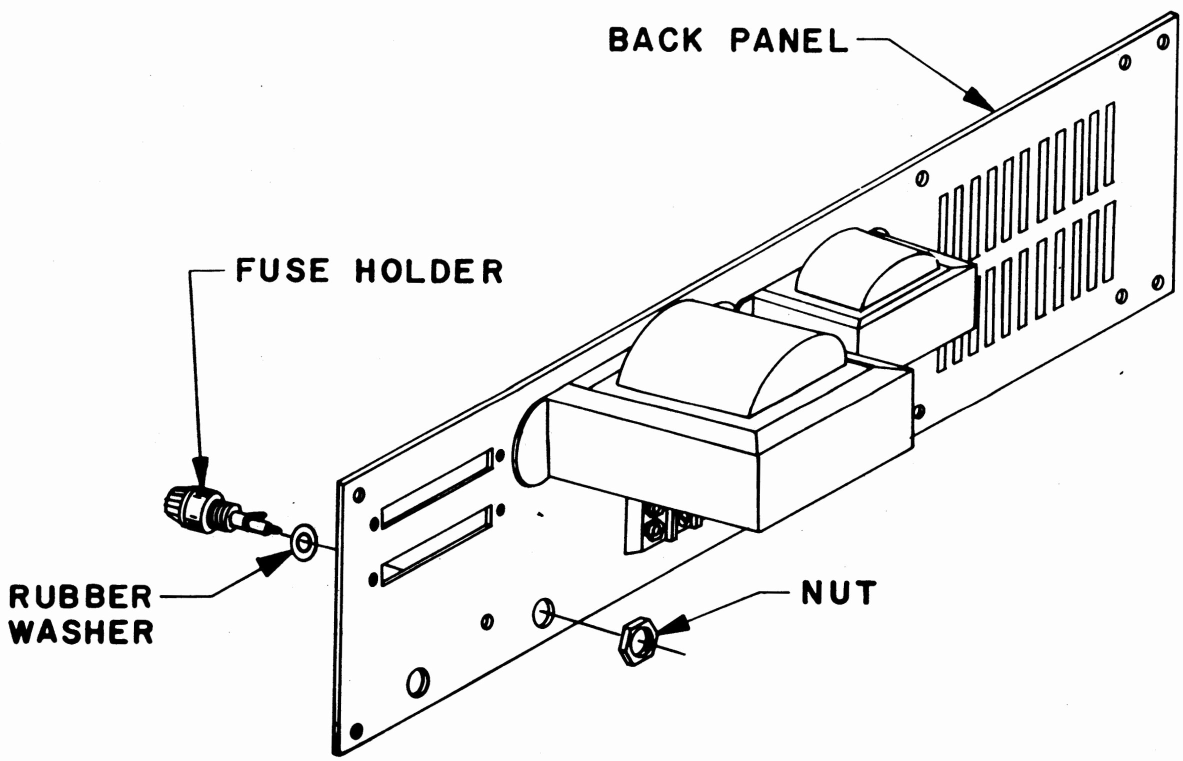

BACK PANEL
(2)#6-32X1/2"
SCREWS
T1
TERMINAL
LUG
(3)#6 LOCKWASHERS
(2)#6-32 NUTS


(2)#6-32X1/2"
SCREWS
T2
(4)#6 LOCKWASHERS
(2)#6-32 NUTS
BACK PANEL

Fuse Holder Installation

Refering to the drawing below, mount the fuse holder to the back panel using the rubber washer and nut provided. Tighten it firmly into place.

Remove the cap and place the fuse provided with your kit into the holder, then replace the cap.

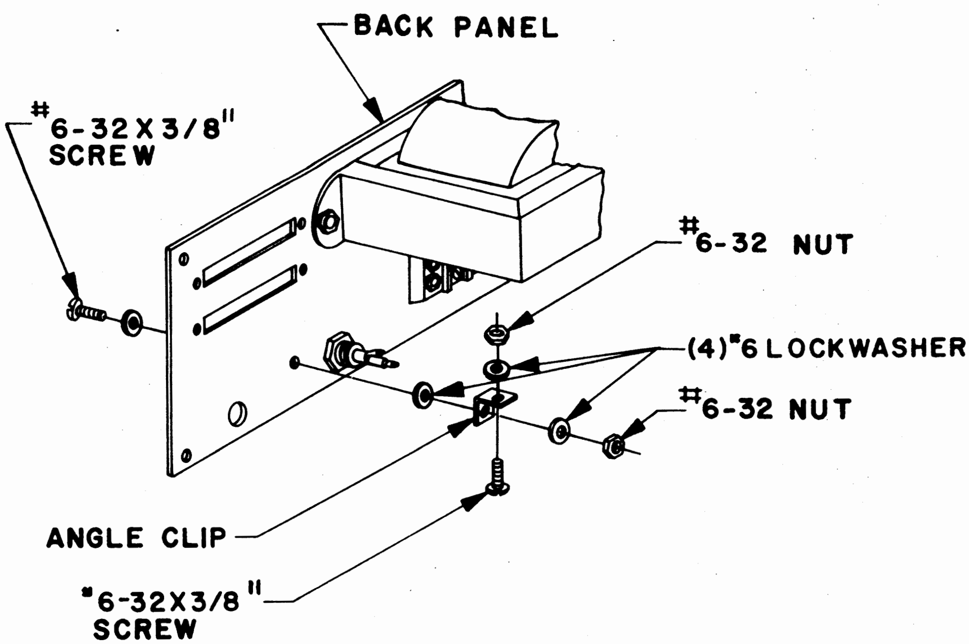


90° ANGLE CLIP INSTALLATION

The drawing below illustrates the hardware and orientation for mounting the 90° angle clip included with this kit.

NOTE: One side of the clip is slightly shorter than the other. The shorter side should be mounted against the back panel with the longer side extending at 90°.

Install the clip as shown below and tighten the screws firmly into place. Be sure that clip remains "square" with the panel when tightening the screws.



Fan Installation

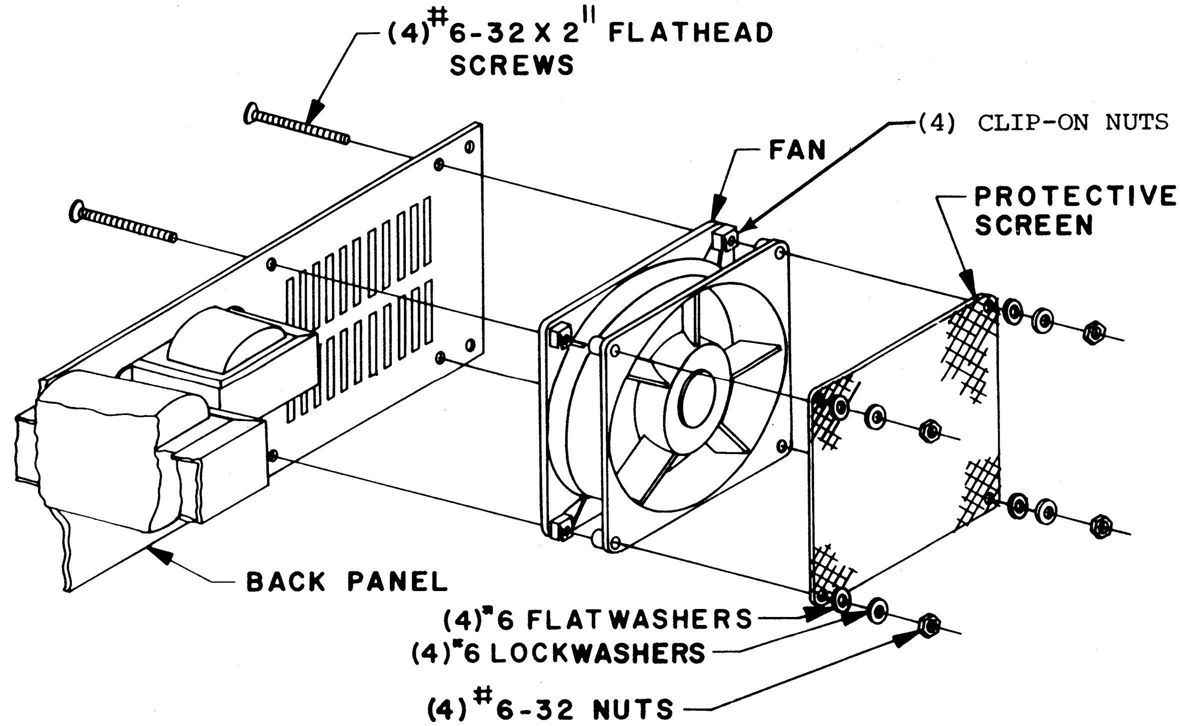
Before the cooling fan is installed onto the back panel, two lengths of wire must be prepared and connected to it.

There is some black wire included with the kit; cut two 6 1/2 inch lengths of this wire. Strip 1/2 inch of insulation from one end of each of the wires, and 1/4 inch of insulation from the other.

In the same manner as described on page , attach a screw-mount crimp terminal to the 1/2" stripped end of each of the two wires. Tin the 1/4" stripped ends of the wires by applying a thin coat of solder.

There are two terminals on the fan in one of the corners. Solder the ends of the two wires opposite the crimp terminals to the terminals on the fan.

Refering to the drawing below, mount the fan and screen to the back panel using the hardware indicated. For proper orientation, the terminals with the two wires attached should be towards the bottom on the side nearest the terminal block. The arrow printed on the fan to indicate airflow should be facing towards the screen. The screen itself has a bump on one side in each of the four corners. The side with the bumps should be towards the fan.



Power Cord Installation

There is a 3-wire power cord included with this kit which must be prepared as follows before installation.

1) Strip 4" of the cord casing from the wires by cutting a circle 4" from the end and pulling off the black insulation. Be careful not to cut into the insulation on any of the wires inside.

2) The green wire inside should already be at the correct length of 4 inches. Cut the white wire to 3 1/2 inches, and the black wire to 1 1/4 inches. Strip 1/4 inch of insulation from the ends of each of the three wires.

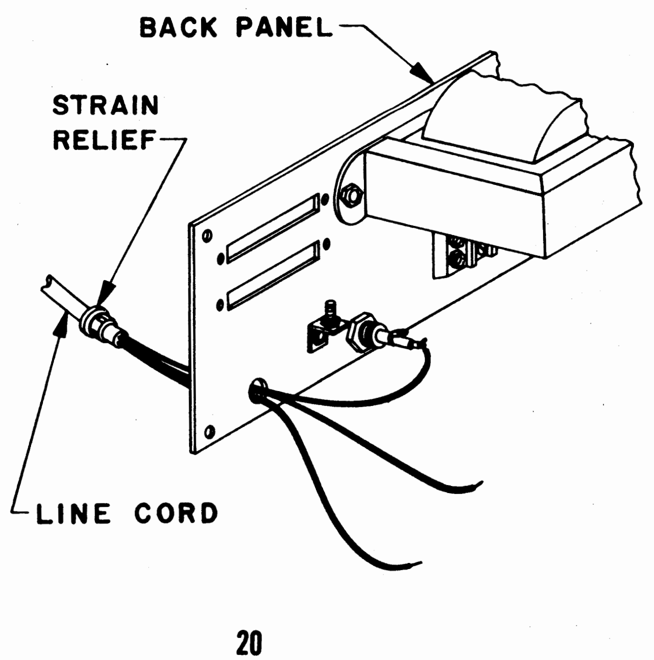
3) Tin the exposed 1/4" of the black wire by applying a thin coat of solder.

4) Solder or crimp screw-mount crimp terminals to the white and green wires.

Place the strain relief, included with the kit, over the power cord. Be sure that the larger diameter end of the relief is towards the male plug end of the cord.

Be sure that there is approximately three inches of the cord's black insulation case extending beyond the strain relief*, then snap it into place on the back panel as shown below.

* The black wire should reach to the center of the fuse holder when the cord & strain relief are in place.



Wire Preparation

Using the wire supplied with this kit, and the length of yellow/green wire cut from transformer T1, prepare the power supply interconnect wires according to the following instructions.

To avoid confusion, it would be best to prepare these wires one at a time.

The list on the right indicates the color of each wire, the length to which it should be cut, and a reference "tag".

Use the following steps to prepare each wire:

1) Cut the specified color wire to the length indicated.

2) Strip 1/2 inch of insulation from one end and 1/4 inch from the other.

3) Tin the wire exposed 1/4 inch by applying a thin coat of solder.

4) According to the instructions on page , connect a screw-mount crimp terminal to the 1/2 inch stripped end.

5) Approximately 5 inches from the 1/4 inch tinned end of the wire label it, using masking tape, with the reference tag indicated.

An additional length of BLACK wire should be cut to 22 1/2 inches and 1/4 inch of insulation stripped from each end. Tin both ends by applying a thin coat of solder. Label this wire "FUSE".

Interconnect Wires

COLOR	LENGTH		TAG
Yellow/ Green*	2	inches	3
Black	22 3/4	"	3
Black	17 3/4	"	9
Black	17 1/2	"	10
Black	25	"	1
White	18	"	6
White	17 3/4	"	8
Orange	17 3/4	"	7
Orange	18 1/2	"	4
Orange	18 1/4	"	5

*From transformer T1, This wire need not be labeled.

Back Panel Wiring

The disk back panel assembly may now be completed by connecting all of the wires to their appropriate locations.

(See drawing page 23)

Three solder connections are necessary and should be made first. These include the black power cord wire, the yellow/green wire and the black 22 1/2 inch wire labeled "FUSE".

1) Solder the 1/4 inch tinned end of the yellow/green wire to the solder lug on transformer T1.

2) Solder the black power cord wire to the center terminal on the fuse holder.

3) Solder one end of the black "FUSE" wire to the other fuse holder terminal.

The remaining connections will be made to the terminal block.

The drawing (P.23) shows the proper orientation and connections for all of the wires on the back panel. The "tags" on the wires you prepared earlier refer to the numbers shown on the terminal block.

WARNING: The power supply is a critical part of any electronic system. Check the wiring here several times to be sure you have it correct. Be sure that each of the wires is in the proper location and that all of the screws on the terminal block are tight.

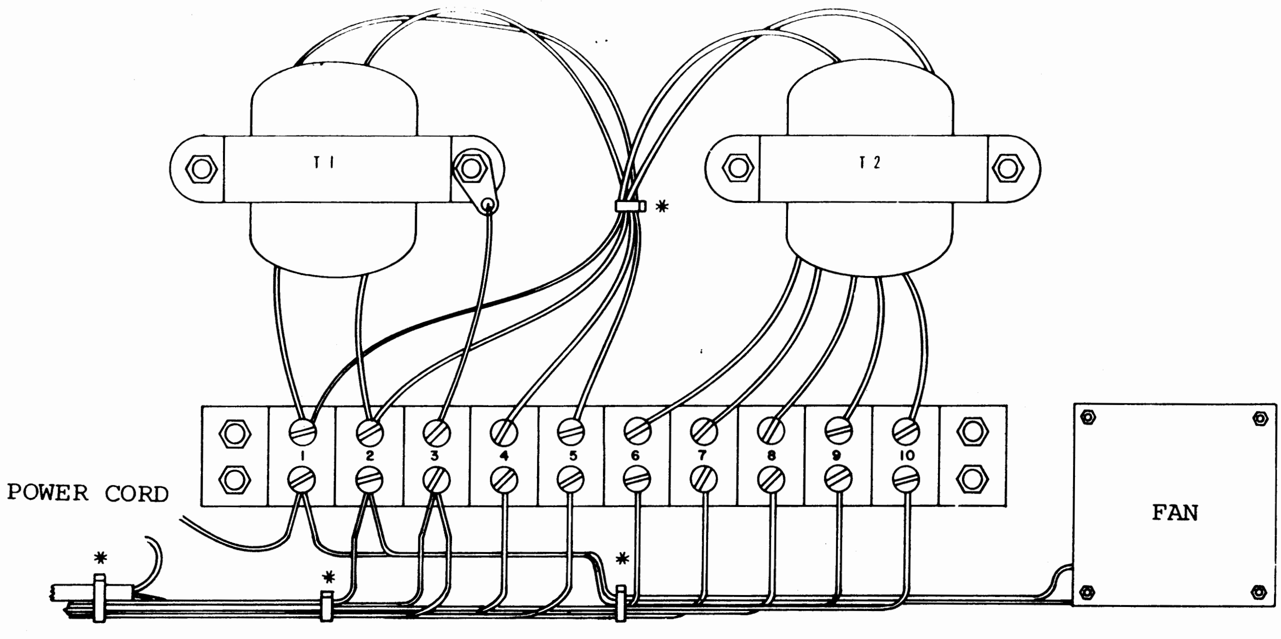
Use the drawing below for reference and connect all of the wires as indicated. Match the "tags" on the wires prepared earlier with the numbered positions on the terminal block. There should be a total of 25 crimp terminal connections made to the block.

NOTE: Where two terminals are to be connected to the same screw, place them "back to back". In this position they will fit flat together, and make a much more solid connection.

The ON-OFF SWitch may also be soldered in at this time. Use the free end of the black "FUSE" wire and the free end of the wire labeled "1" to connect to the switch terminals. There are three terminals on the switch. Use the center terminal and one to either side of it. (The switch position towards the side where the connections are made will be its OFF position

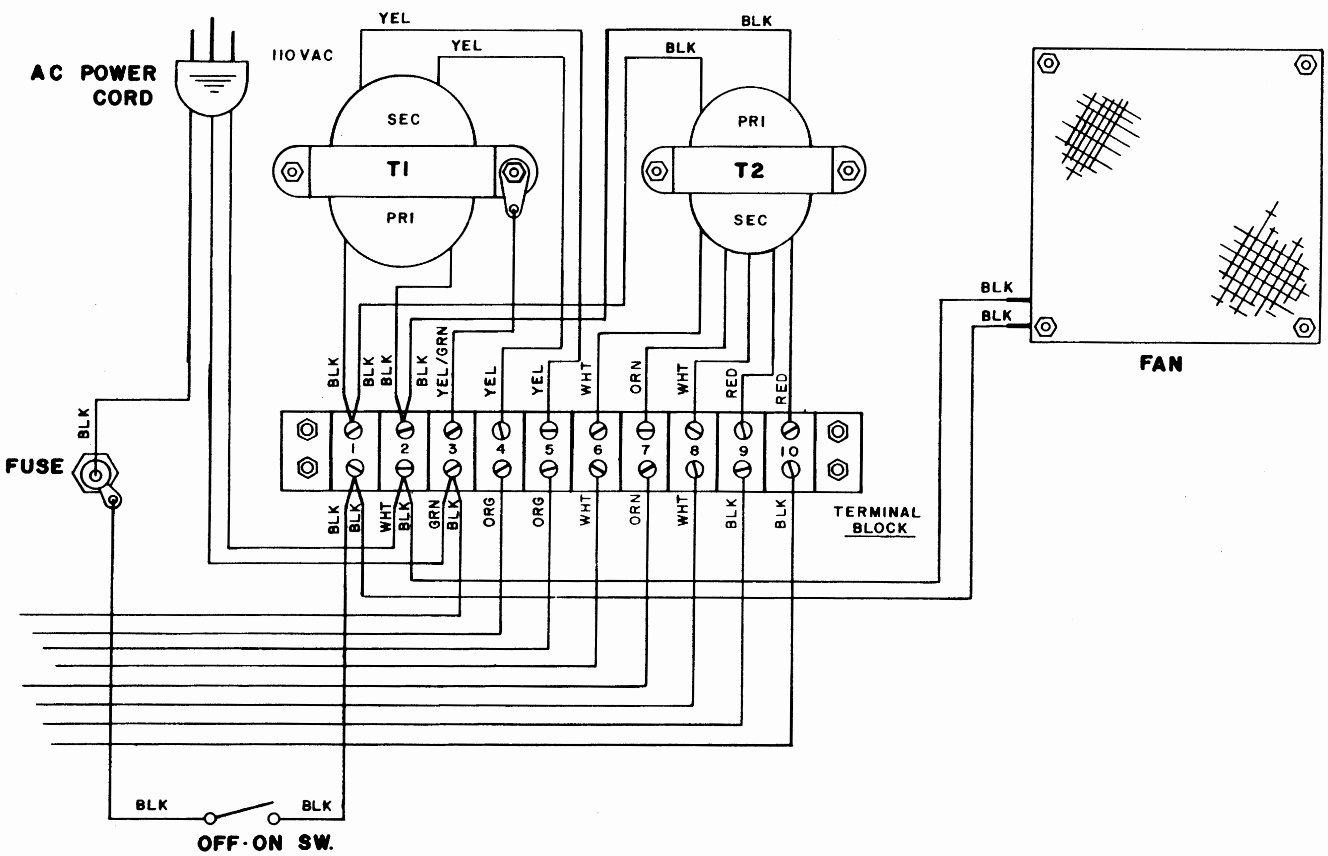
Install the 4 tie wraps in the positions shown in the top drawing on page 23.

WIRE ROUTING & TIE WRAPS



*TIE WRAPS (4)

BACK PANEL WIRING



DISK POWER SUPPLY BOARD ASSEMBLY

NOTE: Save all component leads clipped off during assembly until the entire unit is complete. Some of the leads will be used during the assembly process.

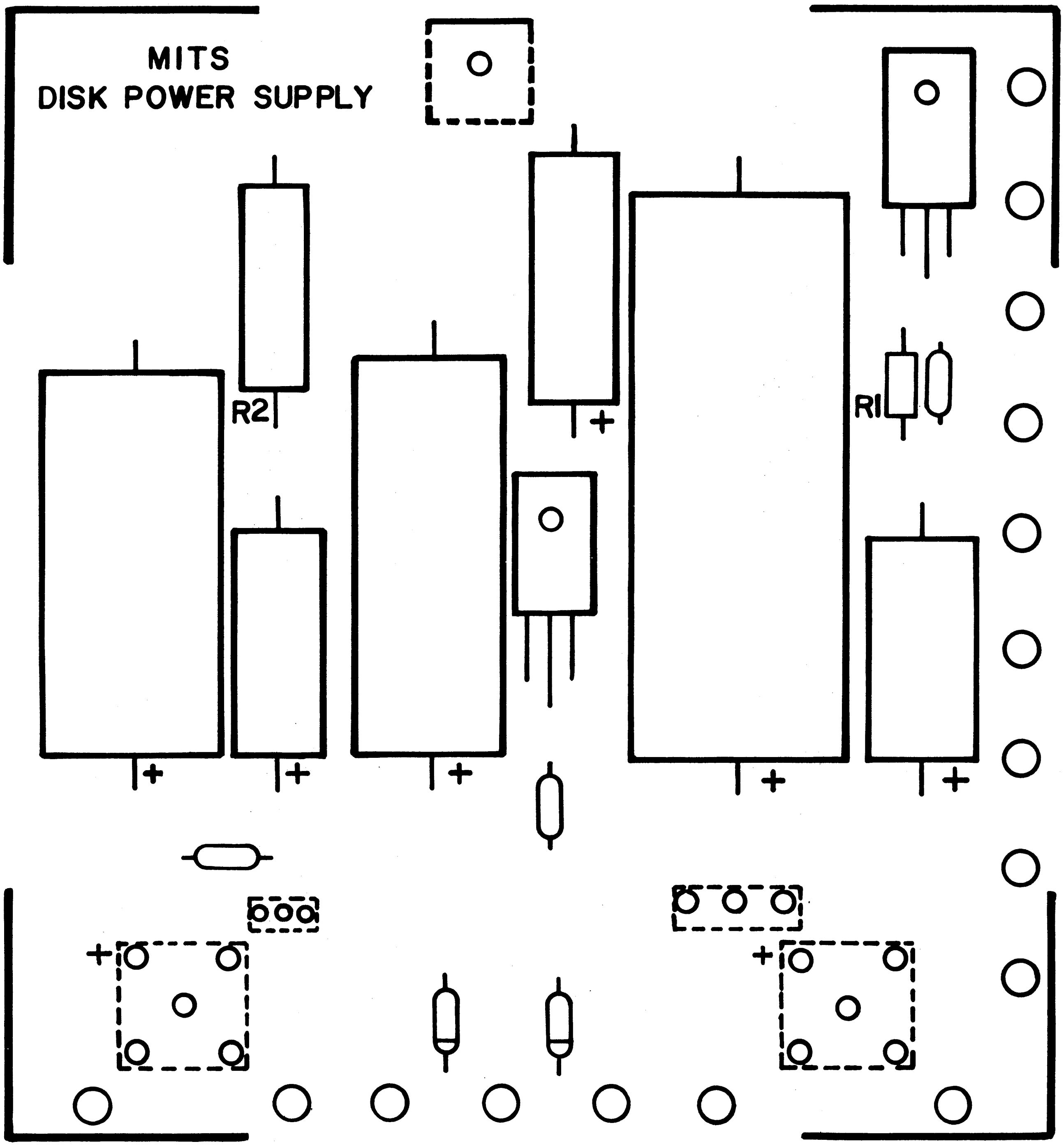
RESISTOR INSTALLATION

Install the following 2 resistors according to the instructions listed on page **5** .

RESISTOR VALUES AND COLOR CODES

( ) R1 is 33 ohm (orange-orange-black) 1/2 W

( ) R2 is 7.5 ohm, 5 W (this may be color coded, violet-green-3rd band white or gold; or it may be a solid body color, with the value printed directly on the resistor itself.


MITS
DISK POWER SUPPLY
R2
R1
+

CAPACITOR INSTALLATION

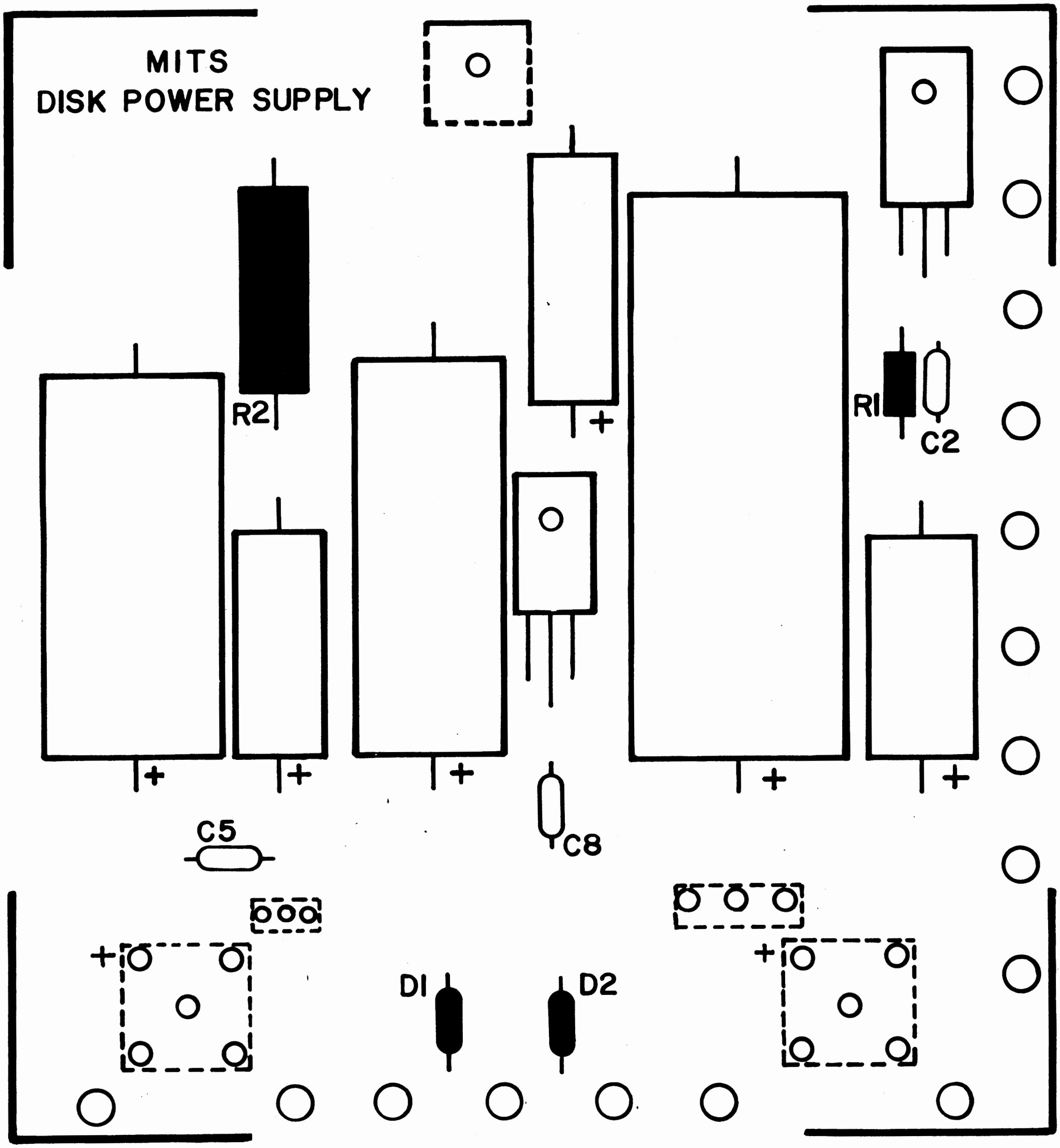
Install the following 3 ceramic disk capacitors according to the instructions on page 6 .

CAPACITOR VALUES

( ) C2 = .luf, 50V

( ) C5 = .luf, 50V

( ) C8 = .luf, 50V



CAPACITOR INSTALLATION

Install the following 6 electrolytic capacitors according to the instructions listed on page 6 .

CAPACITOR VALUES

( ) C1 = 2200uf, 50V

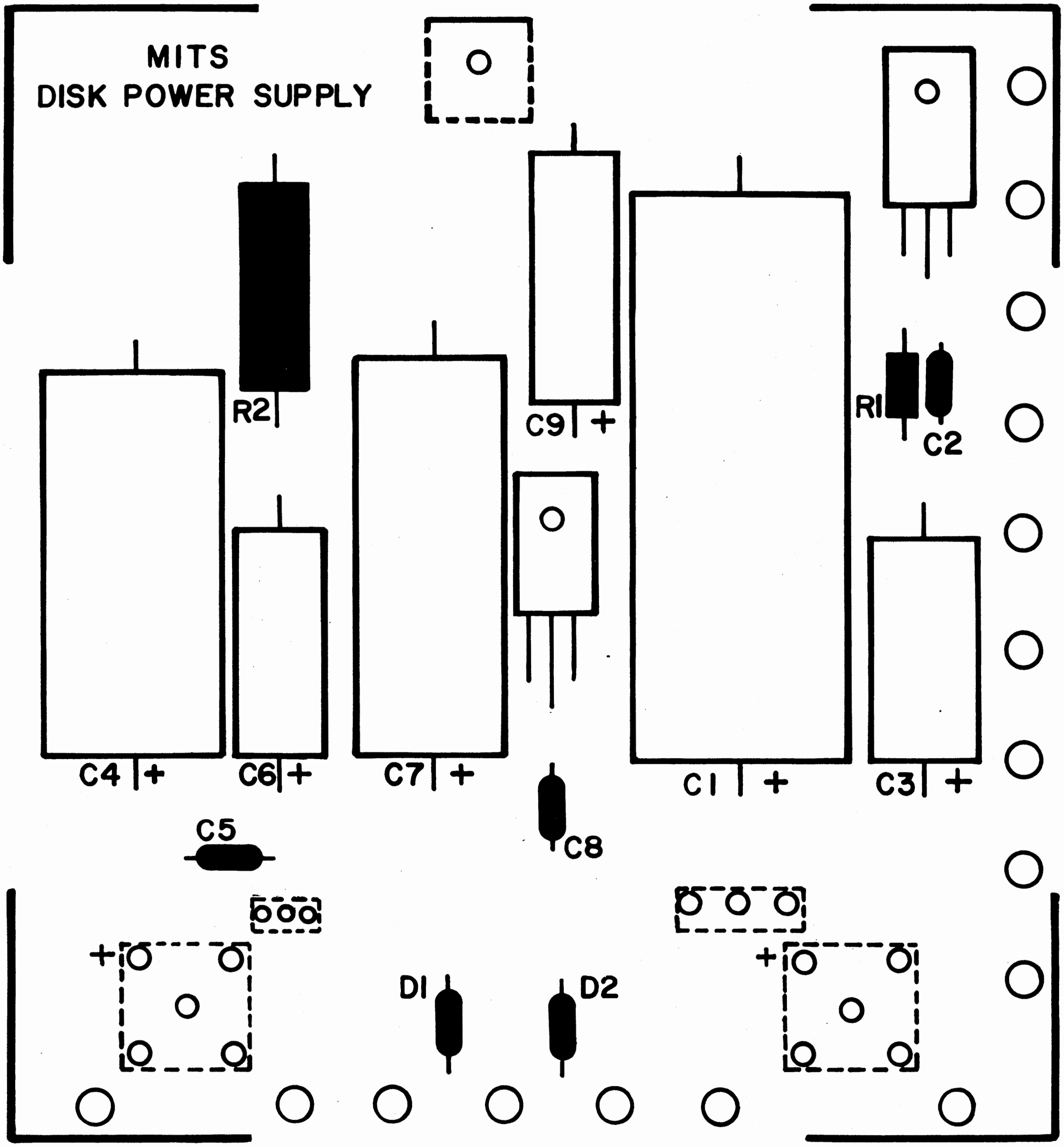
( ) C3 = 33uf, 50V

( ) C4 = 3300uf, 16V

( ) C6 = 33uf, 50V

( ) C7 = 1000uf, 25V

( ) C9 = 33uf, 50V

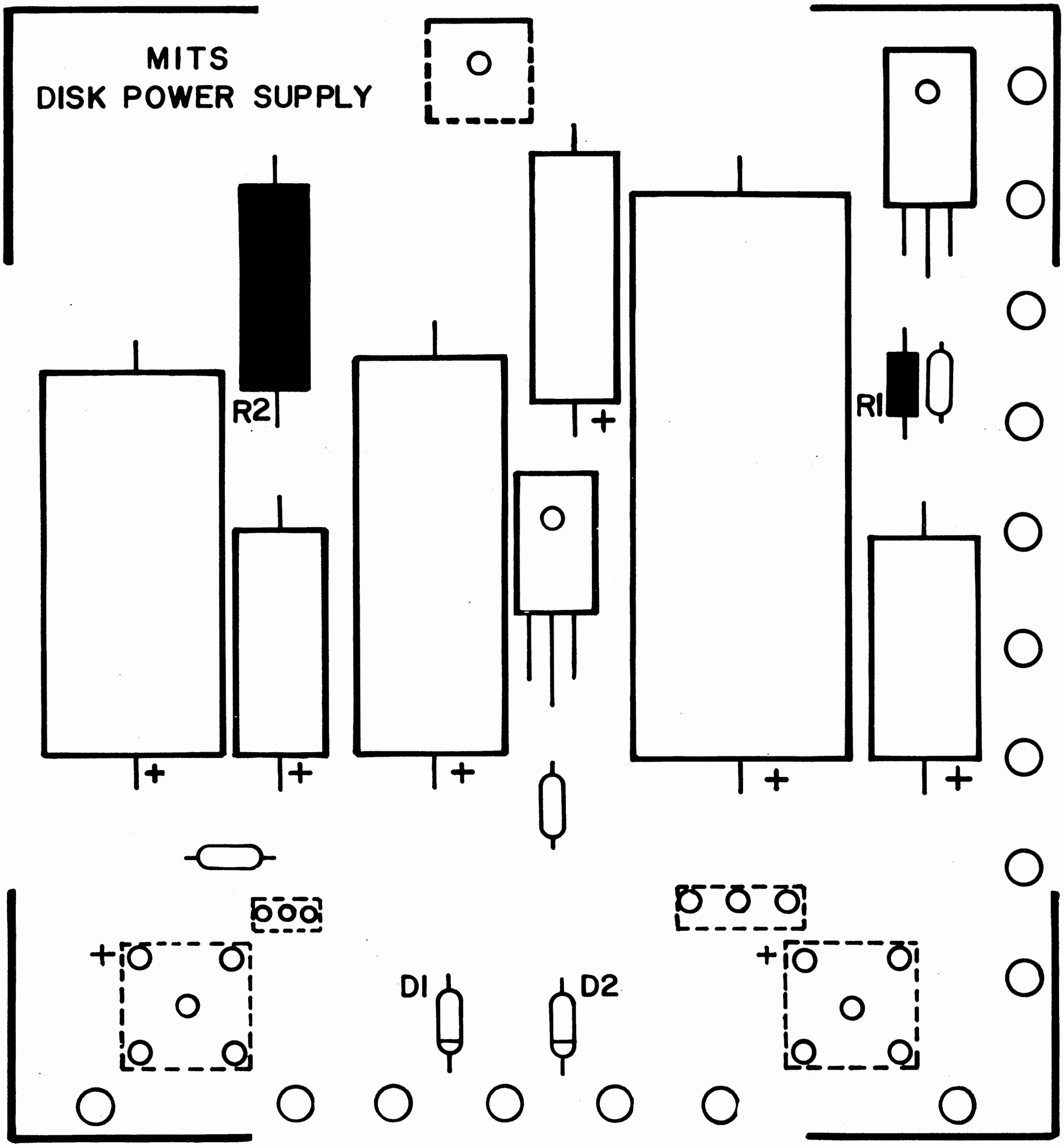

MITS
DISK POWER SUPPLY
R2
C9
+
R1
C2
C4
+
C6
+
C7
+
C1
+
C3
+
C5
C8
+
+
D1
D2

DIODE INSTALLATION

Install the following 2 diodes according to the instructions on page 7.

( ) D1 = 1N4004

( ) D2 = 1N4004



VOLTAGE REGULATOR INSTALLATION

There are 2 voltage regulators to be installed on the silk-screened side of the power supply board, X1 & X3. ← No Heat Sinks

These are to be installed according to the following procedure. (see drawing-right)

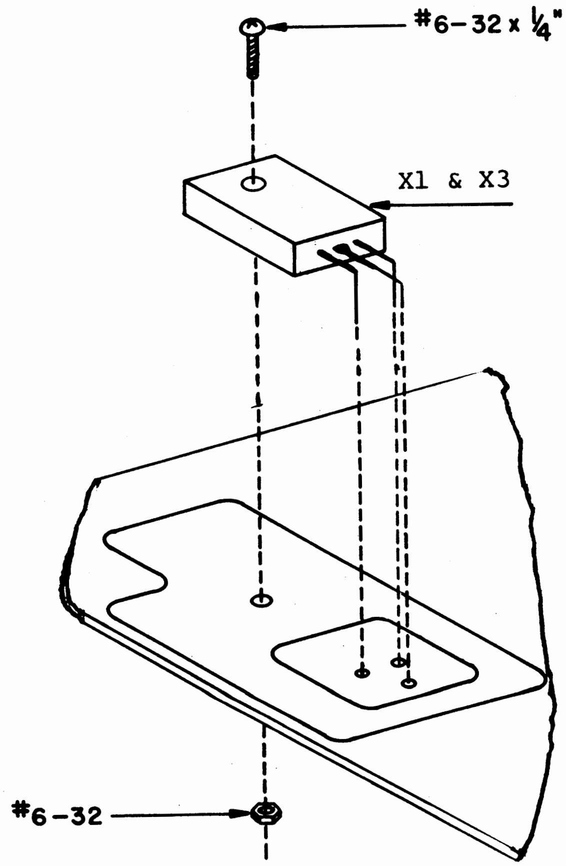
(1) Set the regulator in place over the board so that the mounting hole in the regulator and the board align.

(2) Use a pencil to mark the point on each of the regulator's three leads directly over its corresponding hole in the board.

(3) Bend the three leads, using needle-nose pliers, at right angles from the printed side of the component.

NOTE: Use heat-sink grease when installing this component. Apply it to the surface where the regulator & board come in contact.

(4) Referring to the drawing, set the regulator in place on the silk-screened side of the board. Secure it to the board using a #6-32 nut and screw. Hold the regulator in place as you tighten the nut to keep from twisting the leads.



(5) Turn the board over and solder the three leads to the foil pattern on the back side of the board. Be sure not to leave any solder bridges.

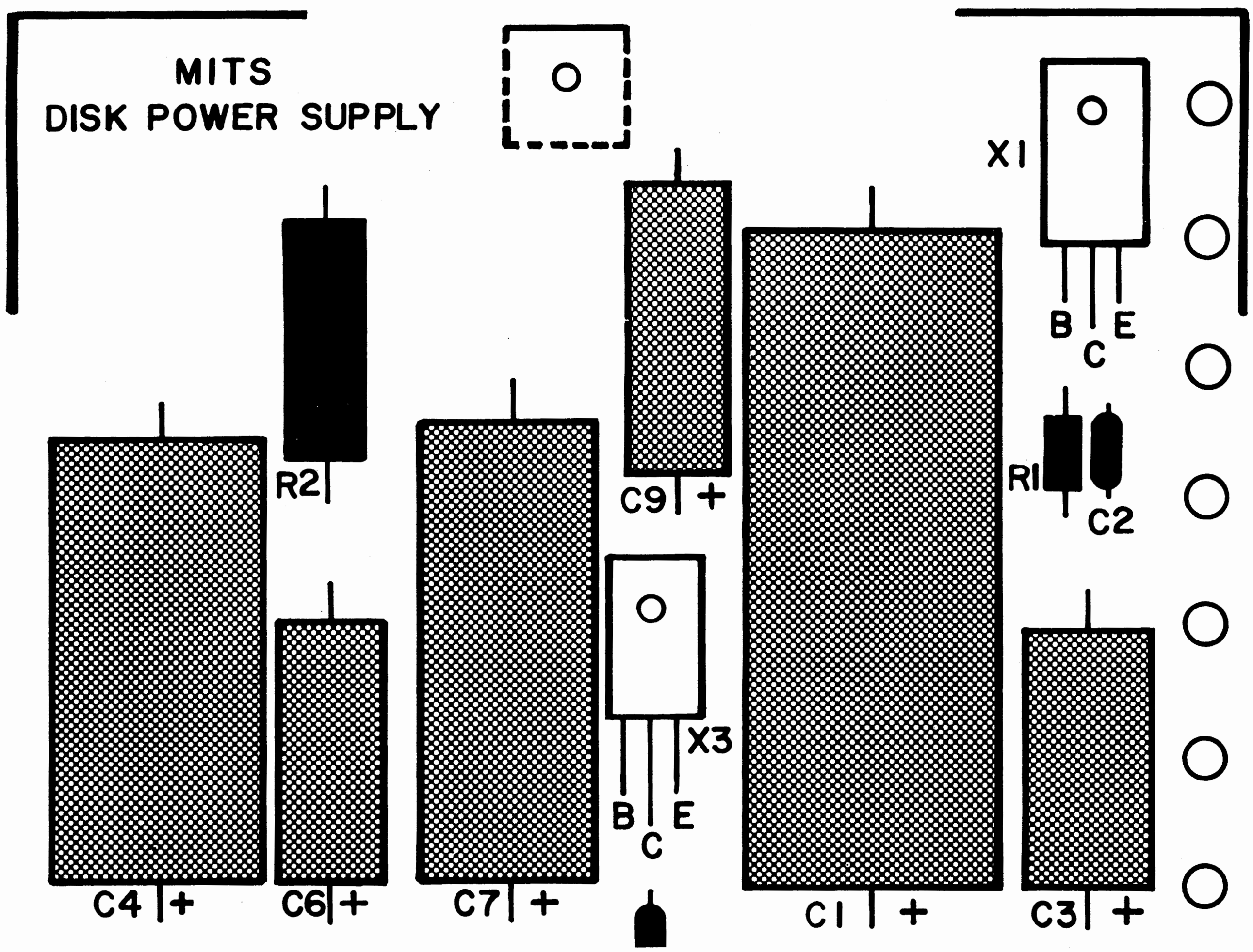
(6) Clip off any excess lead lengths.

Use the above procedure to install both of the regulators, X1 & X3.

VOLTAGE REGULATOR INSTALLATION

(✓) X1 = 7824

(✓) X3 = 7805



BRIDGE RECTIFIER INSTALLATION

There are two bridge rectifiers, BR1 & BR2, to be installed on the power supply board.

WARNING: Read the following instructions closely. Proper orientation of these two components is absolutely critical.

These two components are indicated on the silk-screen by broken lines. This is to indicate that they are to be mounted on the bottom (non-silk-screened) side of the board.

You will observe a "+" sign printed near one corner of the rectifier. The lead nearest this "+" sign is the positive lead of the rectifier. This lead must be inserted into the hole marked on the silk-screen with a "+" sign.

NOTE: There is also a "-" sign printed on the regulator. The lead nearest this sign is the negative lead of the rectifier, and should be diagonally opposite the "+" lead on the board.

BE ABSOLUTELY SURE THAT THE PROPER ORIENTATION IS USED WHEN INSTALLING THESE TWO COMPONENTS.

Install the rectifiers according to the following procedure:

(1) Insert the four leads of the BR1 rectifier into their respective holes from the non-silk-screened side of the board. Be sure the "+" lead of the rectifier is inserted in the hole labeled "+" on the silk-screened side of the board.

(2) Insert the BR2 rectifier in the same manner. Be sure both rectifiers are pushed all the way against the board.

(3) There is a 90° angle bracket included with your parts. Each of the two sides has two holes in it.

Using the side with the two holes the furthest apart, set the angle bracket over the two rectifiers. The holes in the bracket, the rectifiers, and the board should align.

Temporarily attach the bracket & rectifiers to the board through these holes using #6-32 & 5/8" screws and nuts.

(4) Check the orientation once more, then solder all four leads of each rectifier to the board on the silk-screened side.

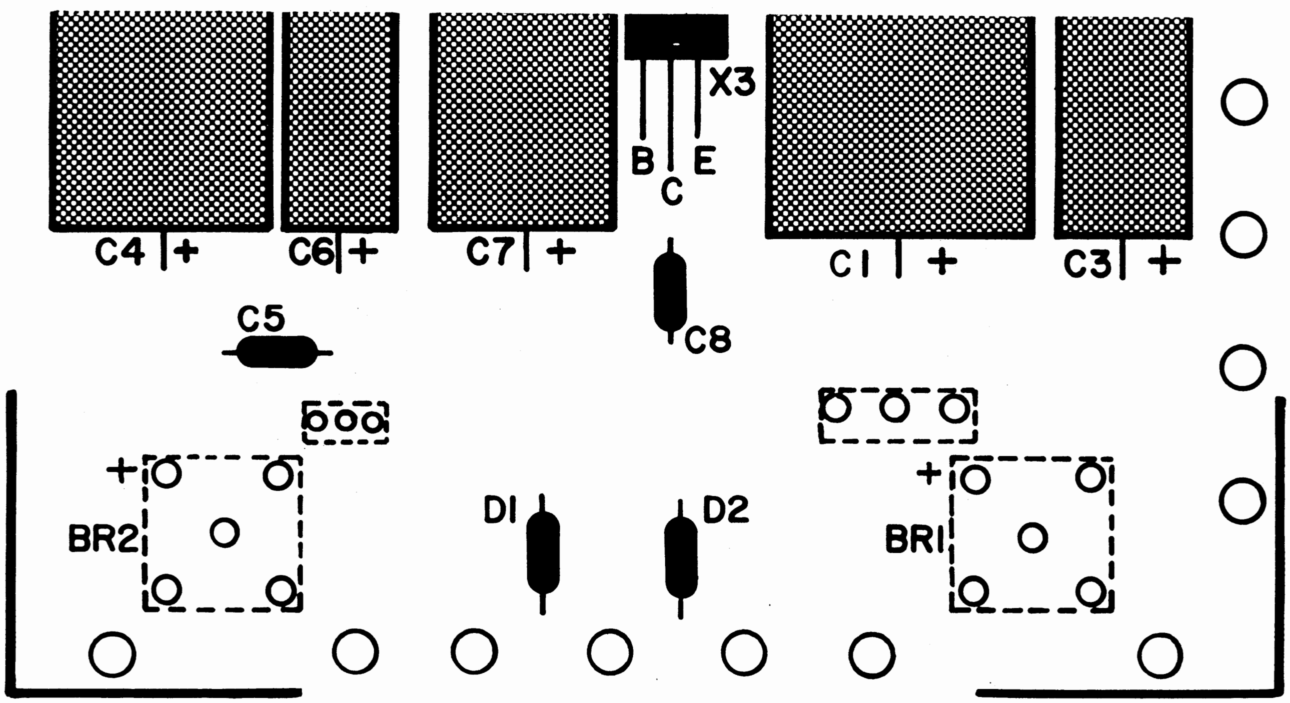
(5) Clip off any excess lead lengths. Leave the angle bracket in place for the next procedure.

NOTE: Apply heat-sink compound to all mating surfaces.

BRIDGE RECTIFIER INSTALLATION

(X) BR1 = VJ048

(X) BR2 = VJ048



DISK BUFFER BOARD
ASSEMBLY

IC INSTALLATION

Install the following 7 ICs onto the Disk Buffer Board according to the method described on page 4.

IC SILK-SCREEN DESIGNATIONS AND PART NUMBERS

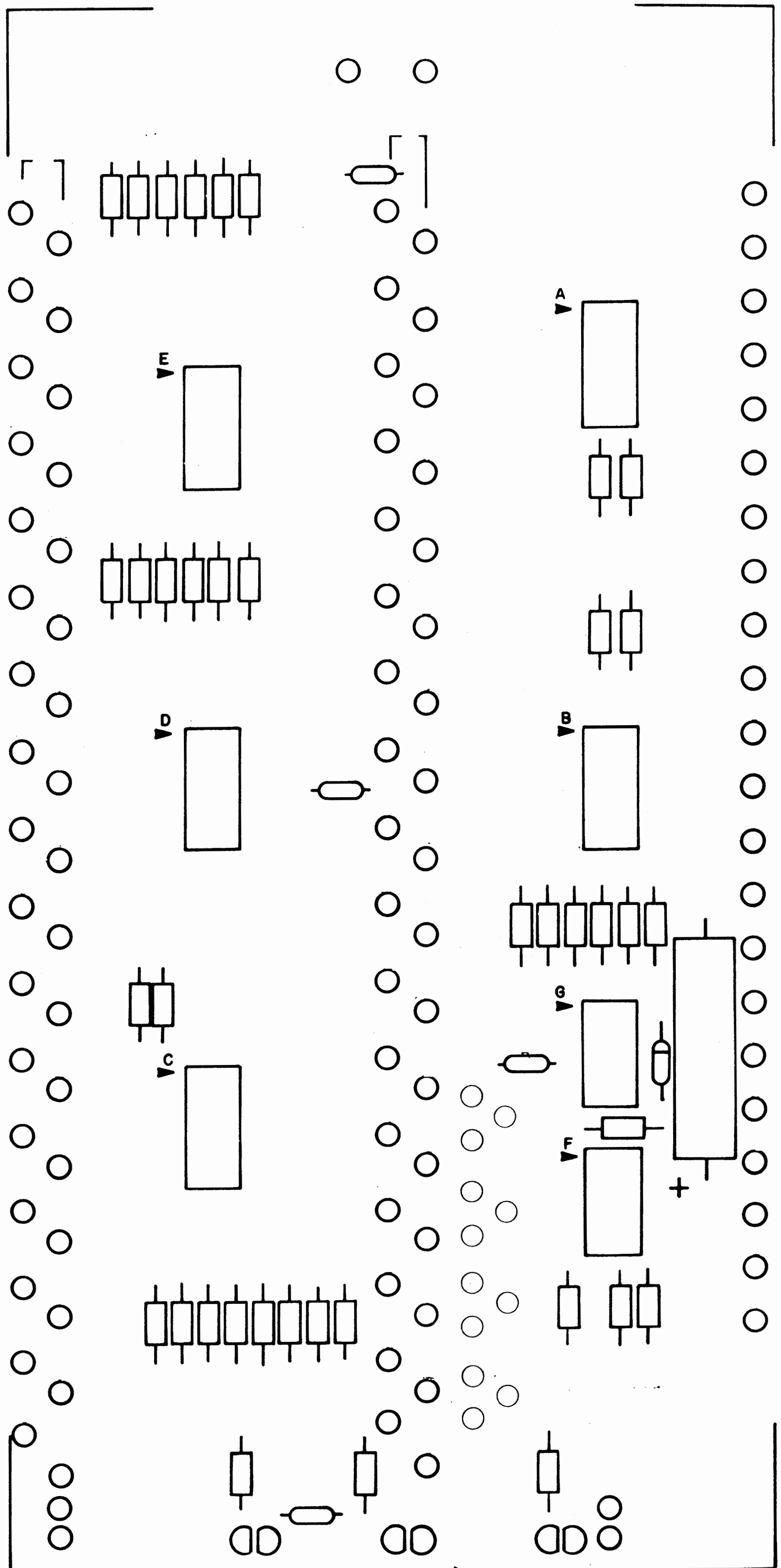
36361

( ) A, B, D, & E = 8T97

( ) C = 8T98

( ) F = 74L30

( ) G = 9601



RESISTOR INSTALLATION

Install the following 39 resistors according to the instructions listed on page **5** .

RESISTOR VALUES AND COLOR CODES

( ) R9, R7, R5 are 220 ohm (red-red-brown) 1/2 W

( ) R10, R8, R6 are 330 ohm (orange-orange-brown) 1/2 W

( ) R12, R14, R16 are 330 ohm (orange-orange-brown) 1/2 W

( ) R11, R13, R15 are 220 ohm (red-red-brown) 1/2 W

( ) R33 is 220 ohm (red-red-brown) 1/2 W

( ) R34 is 330 ohm (orange-orange-brown) 1/2 W

( ) R31, R29, R27, R25 are 220 ohm (red-red-brown) 1/2 W

( ) R32, R30, R28, R26 are 330 ohm (orange-orange-brown) 1/2 W

( ) R36, R35, R37 are 150 ohm (brown-green-brown) 1/4 W

( ) R40 is 330 ohm (orange-orange-brown) 1/2 W

( ) R39 is 220 ohm (red-red-brown) 1/2 W

( ) R38 is 1K ohm (brown-black-red) 1/2 W

( ) R41 is 39K ohm (orange-white-orange) 1/2 W

( ) R20, R22, R24 are 330 ohm (orange-orange-brown) 1/2 W

( ) R19, R21, R23 are 220 ohm (red-red-brown) 1/2 W

( ) R4 & R18 are 330 ohm (orange-orange-brown) 1/2 W

( ) R3 & R17 are 220 ohm (red-red-brown) 1/2 W

Insert Page

ALTAIR FLOPPY DISK

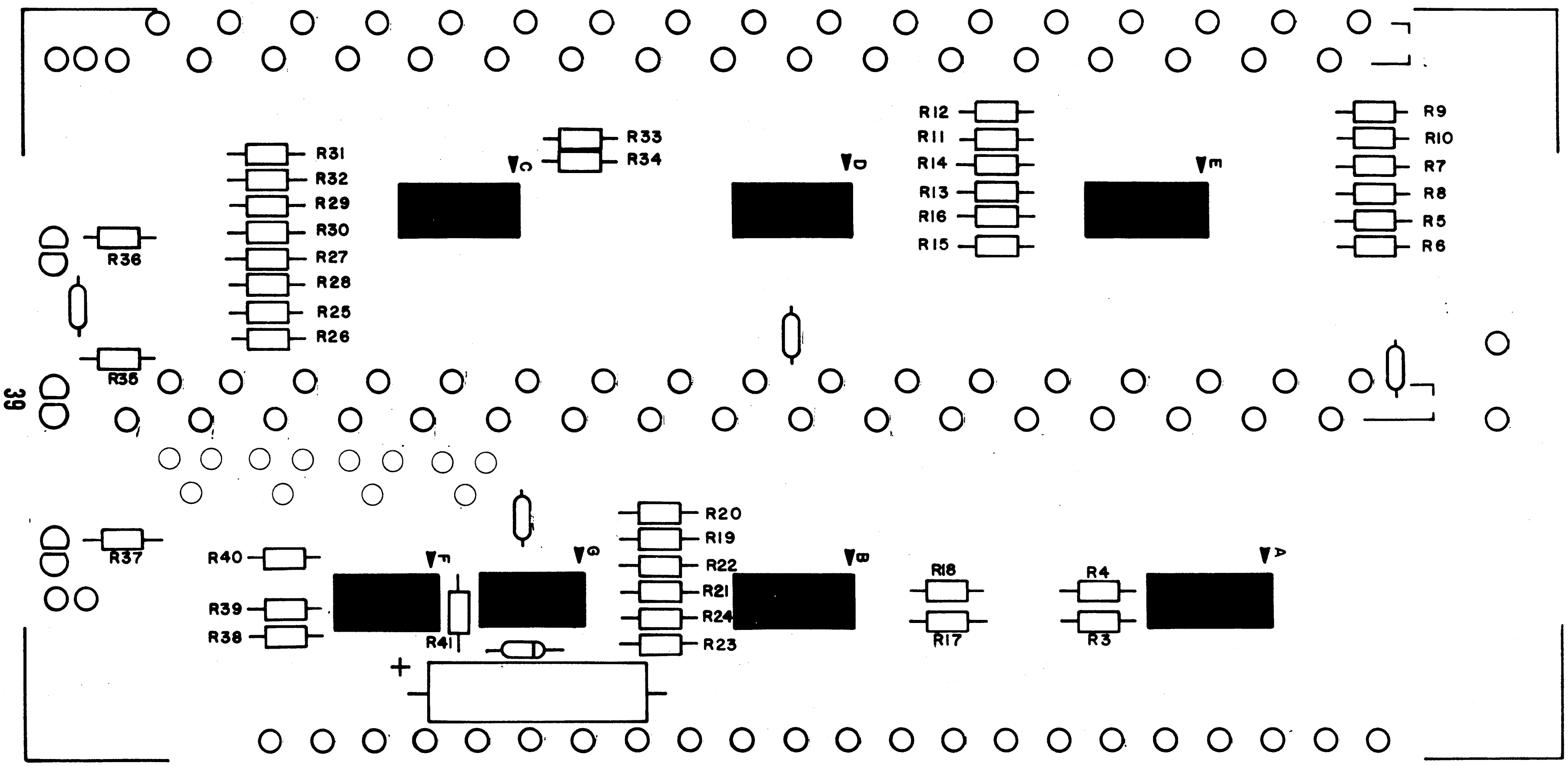
Disk Drive Assembly Procedure

Resistor Value Changes, page 38

R39 should be 330 ohms

R40 should be 220 ohms

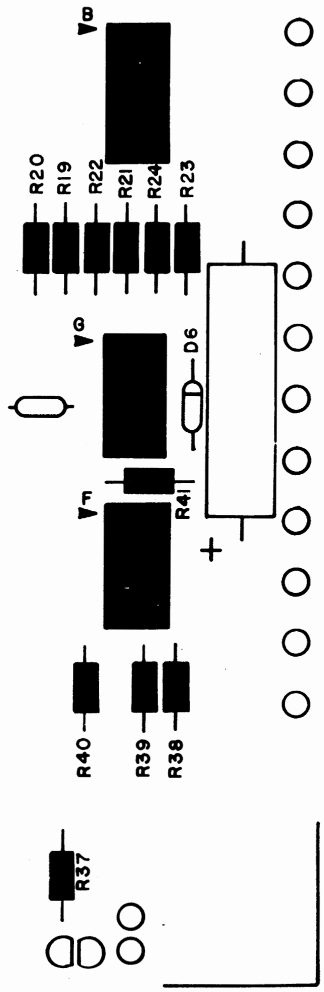
MITS, Inc.
August, 1976


R31
R32
R29
R30
R27
R28
R25
R26
R33
R34
C
D
E
R12
R11
R14
R13
R16
R15
R9
R10
R7
R8
R5
R6
R36
R35
R37
R40
R39
R38
R41
F
G
R20
R19
R22
R21
R24
R23
B
R18
R17
R4
R3
A
+

DIODE INSTALLATION

Install diode D6 according to the instructions on page 7 .

( ) D6 = 1N914



CAPACITOR INSTALLATION

Capacitor C14 is an electrolytic capacitor. Capacitors C10, C11, C12, and C13 are ceramic disk capacitors.

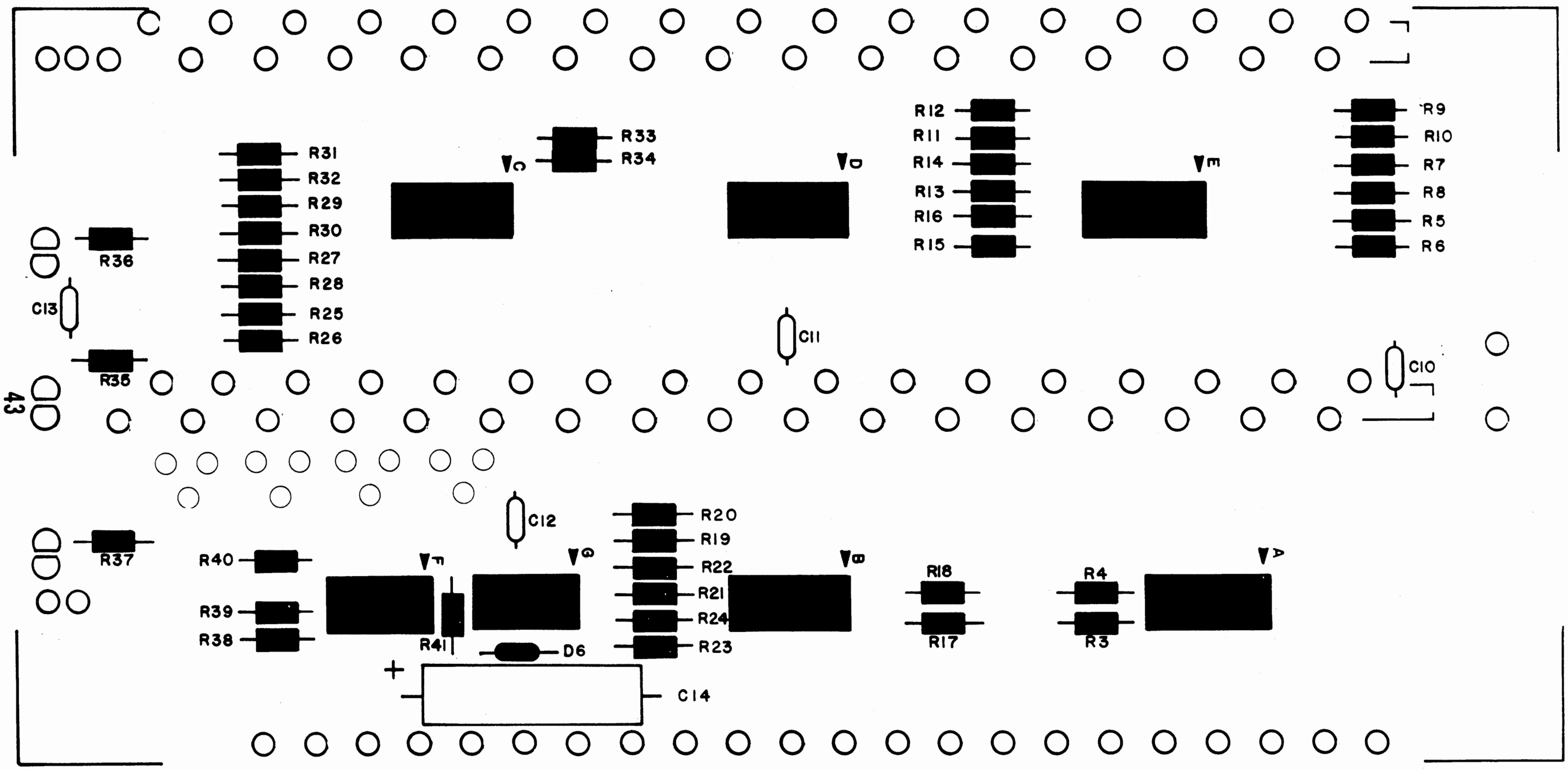
Install these components according to the instructions listed on page 6 .

CAPACITOR VALUES

(Different voltages may be substituted in some cases.)

(X) C14 = 500 uf, 25V electrolytic

(✓) C10, C11, C12 & C13 are .1 uf, 12V ceramic disks.


R12
R11
R14
R13
R16
R15
R9
R10
R7
R8
R5
R6
R33
R34
C
D
E
R31
R32
R29
R30
R27
R28
R25
R26
R36
C13
R35
C11
C10
R37
R40
R39
R38
F
G
C12
R20
R19
R22
R21
R24
R23
B
A
R18
R17
R4
R3
R41
D6
+
C14

Ribbon Cable Preparation

There are three ribbon cable assemblies to be prepared for installation in the disk drive unit. A 12' length of 18-twisted pairs cable has been provided for this purpose.

First, cut the 12' length of cable into two 18-inch lengths and one 25-inch length. The remainder of the cable should be saved for later use.

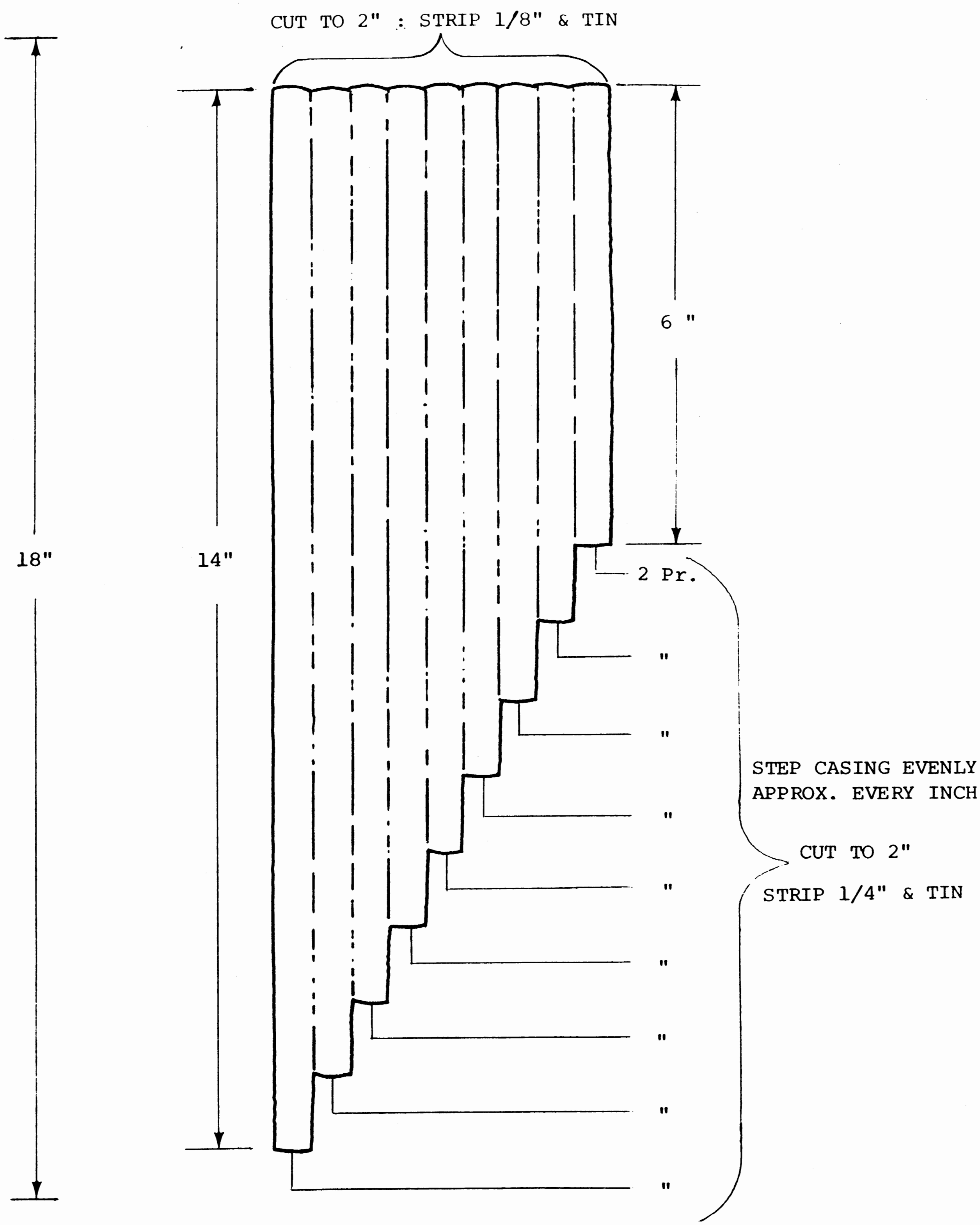
The following two pages contain diagrams for the proper lengths and arrangement for the three cable pieces you have just cut. The two 18" lengths will be prepared identically.

The cable sheath itself may be cut using scissors, and can be stripped by simply pulling it apart. You will note that the plastic sheath has "welds" approximately every inch between the twisted pairs. Try not to make any cuts on the welds themselves.

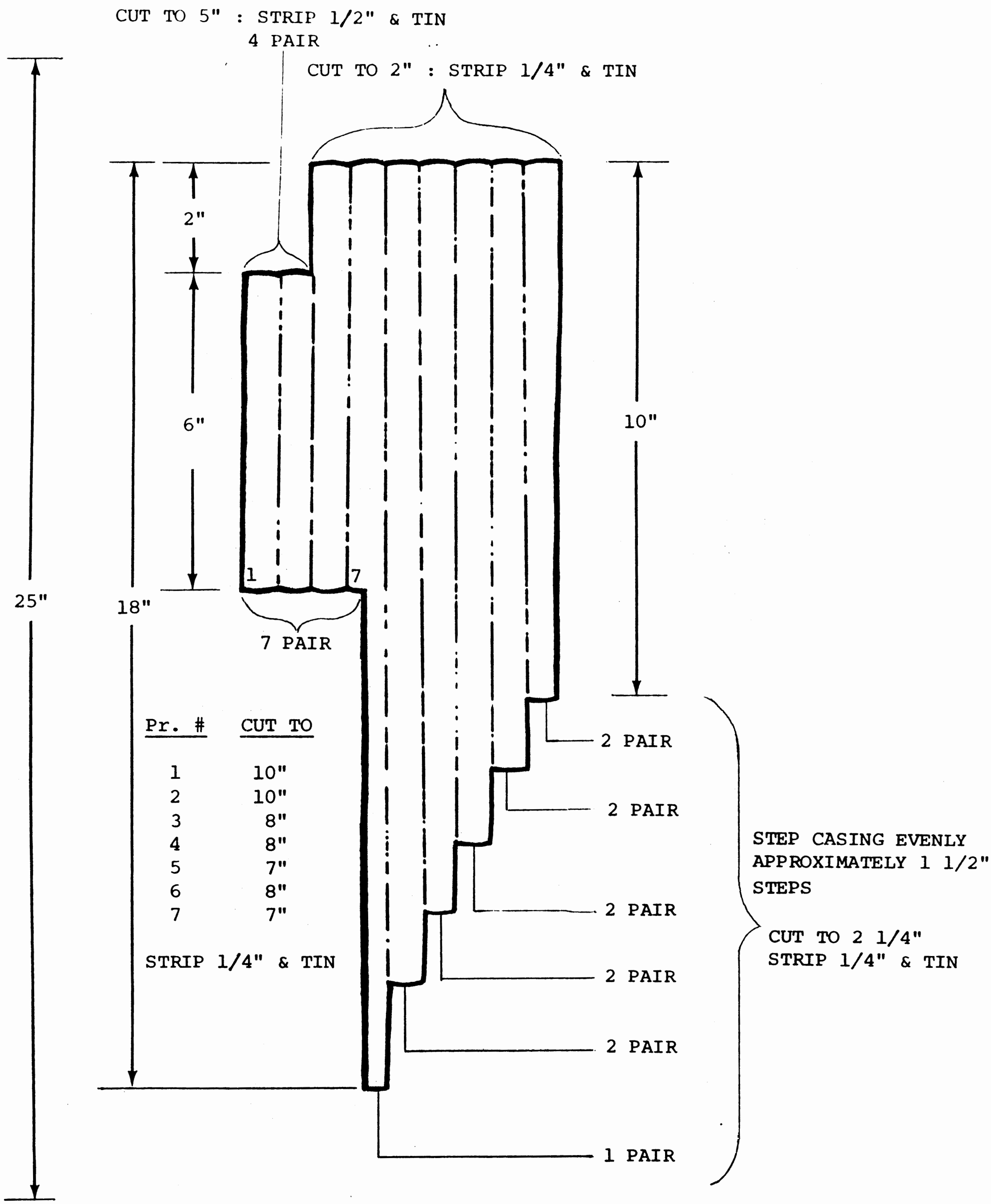
Each time a 1/4" of insulation is stripped from the wires themselves, the bare ends should be tinned by applying a thin coat of solder.

Study the diagrams on the next two pages and prepare the three cable assemblies as shown. Be careful to cut the wires precisely as indicated, and do not damage the wire insulation when cutting the cable sheath.

18 INCH RIBBON CABLES



25 INCH RIBBON CABLE



Pr. #	CUT TO
1	10"
2	10"
3	8"
4	8"
5	7"
6	8"
7	7"

STRIP 1/4" & TIN

There are several 37-pin connectors in this kit. One male connector and one female connector will be used now to connect onto one end of each of the two 18 inch lengths of ribbon cable that you have just prepared. The other end of the two cables will connect directly to the Disk Buffer board.

Connector Preparation

The two 37-pin connectors must first be prepared for attaching to the cables. It may be helpful to solidly mount the connectors to some steady object during this and the following procedures.

1) Place the connector in front of you with the hollow solder pins facing upwards.

2) Using your soldering iron, very carefully heat each pin one at a time and fill the hollow space with solder. The solder should not quite fill the pin and should have a slightly concave surface.

Prepare all 37 pins on one male and one female connector in this manner. Be sure not to leave any solder bridges between the pins, and be careful not to melt any of the nylon insulation around them.

WARNING

During the following procedure, and later steps involving ribbon cable, be sure that you fully understand all of the instructions before you begin. These points are the most likely areas for assembly errors to occur.

Cable Assembly

The following procedure should be used for assembling both of the 18 inch cables. In order to minimize the possibility of error, the cables will be attached to the 37-pin connectors and the Disk Buffer board during the same procedure. Read this entire procedure over carefully before beginning.

You will note that the pins on the 37-pin connectors are all numbered. Note also that the numbers on the male connector are the reverse of the female. The male connector will be wired to the rows of pads on the buffer board labeled "TO". The female connector will be wired to the rows of pads labeled "FROM". The numbers on the connector pins correspond directly with the numbers that label the pads on the buffer board.

The following pages contain drawings of both the 37-pin connectors, and the Disk Buffer board silk-screen. There is a space provided to "check-off" each of the twisted-pair wires as they are connected. Double arrows are also shown to indicate the connection points for each of the twisted-pairs.

Orient one of the 18 inch cables so that the "stepped" edge of the cable casing is along the rows of pads on the buffer board labeled "TO". The longest wires should be near the pads labeled "19 & 37" and the shortest wires near the pads labeled "1 & 20". Place the MALE 37-pin connector near the other end of the cable.

Begin with the shortest twisted-pair of wires, nearest the outside edge of the cable casing, on the buffer board end.

Separate the two wires slightly, then solder them into the two pads labeled "1 & 20" on the buffer board. Do this by inserting the wires from the silk-screened side of the board and soldering them on the back. Be careful not to push any of the wire insulation into the holes. Clip off any excess wire from the connections and then check-off the appropriate space on the silk-screen drawing.

The same twisted-pair of wires should now be connected to the pins numbered "1 & 20" on the 37-pin connector.

Observe the color of the wire now connected to the pad on the buffer board labeled "1". Be sure to connect this same wire to the pin numbered "1" on the connector. Do the same with pad "20" and pin "20".

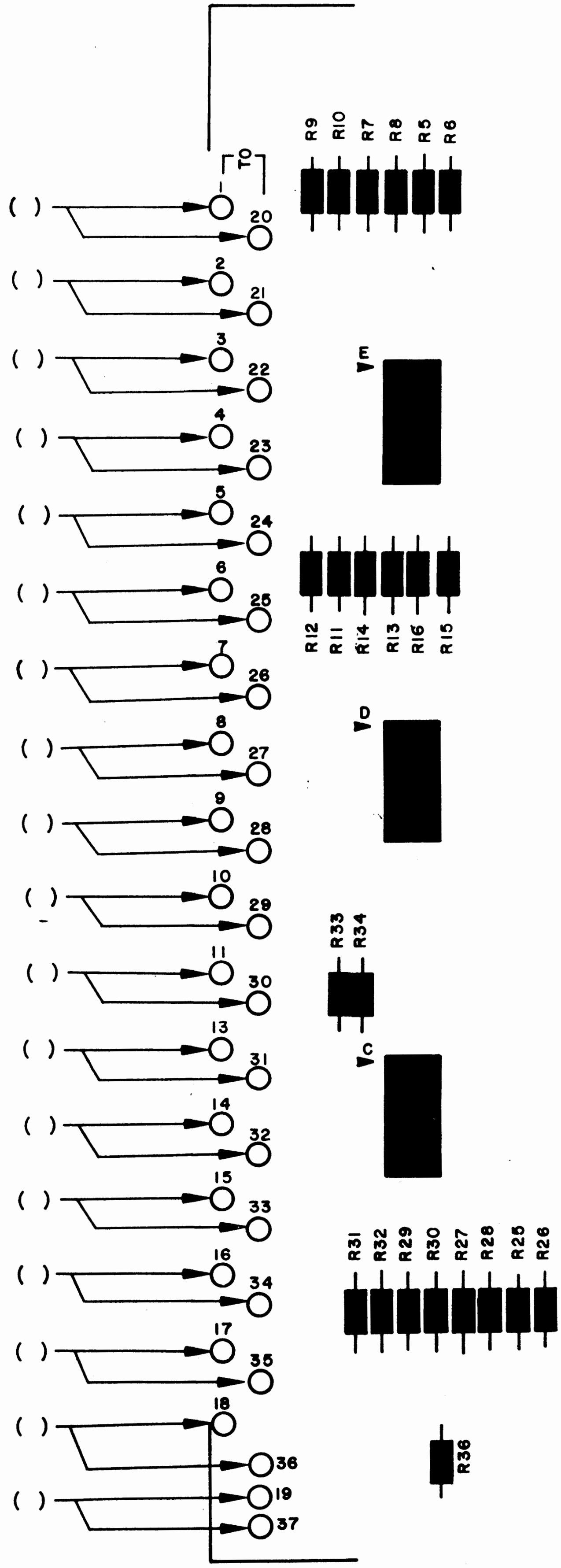
Make the connections by re-melting the solder in the pins and inserting the wires up to their insulation. Remove the heat from the pins while still holding the wires in place until the solder cools. Check-off the appropriate space on the connector drawing.

Move to the next twisted-pair of wires in the ribbon cable and use the same procedure to connect pads "2 & 21" with pins "2 & 21". Continue in this manner, moving across the ribbon cable one pair at a time, until all 18 twisted-pairs are in place. Be sure that you do not connect any wires to pin "12" on the connector.

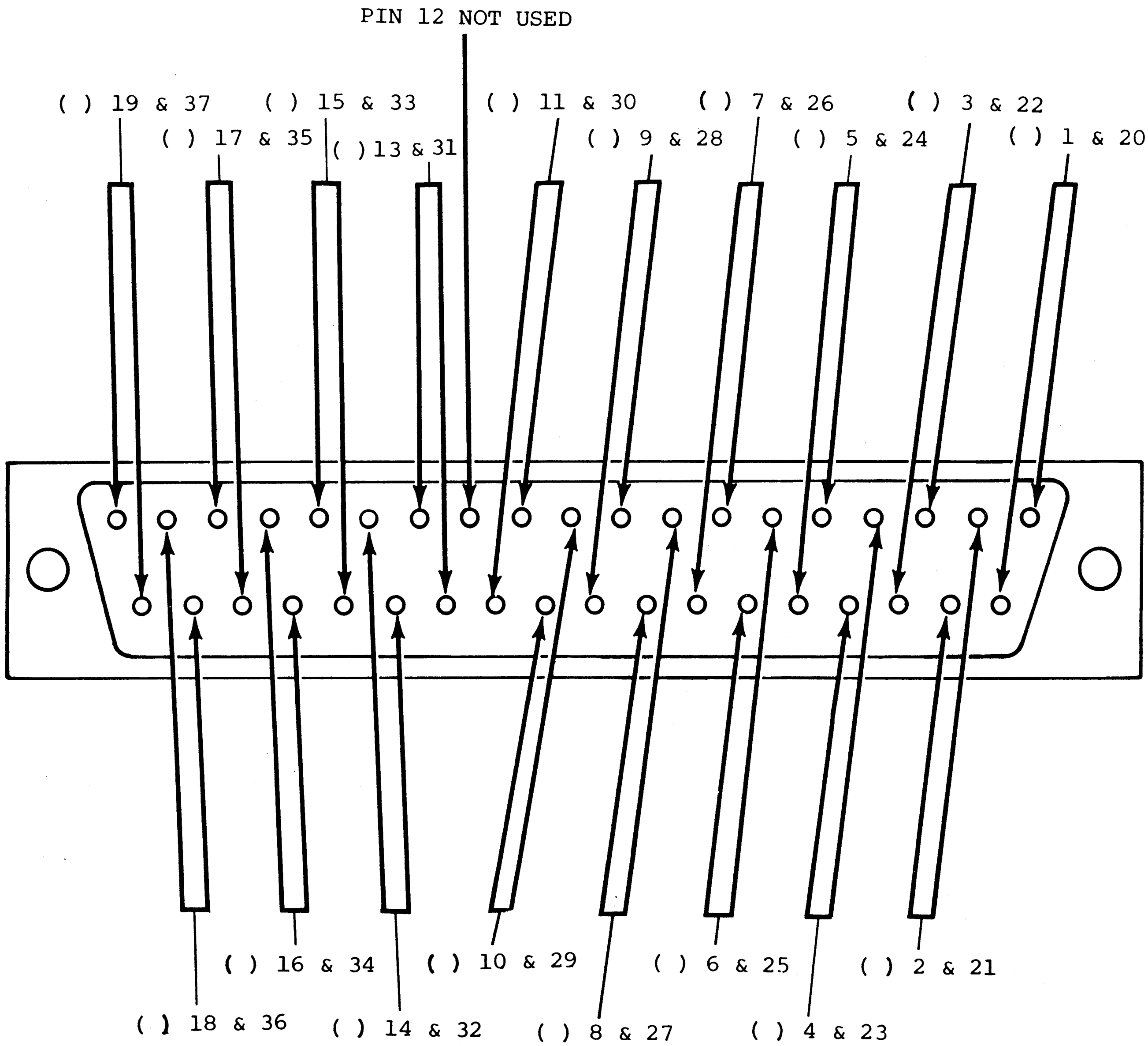
<u>NOTE</u>: Take your time and be careful while soldering the wires to the connectors. Do not melt any of the wire insulation or leave any solder bridges.

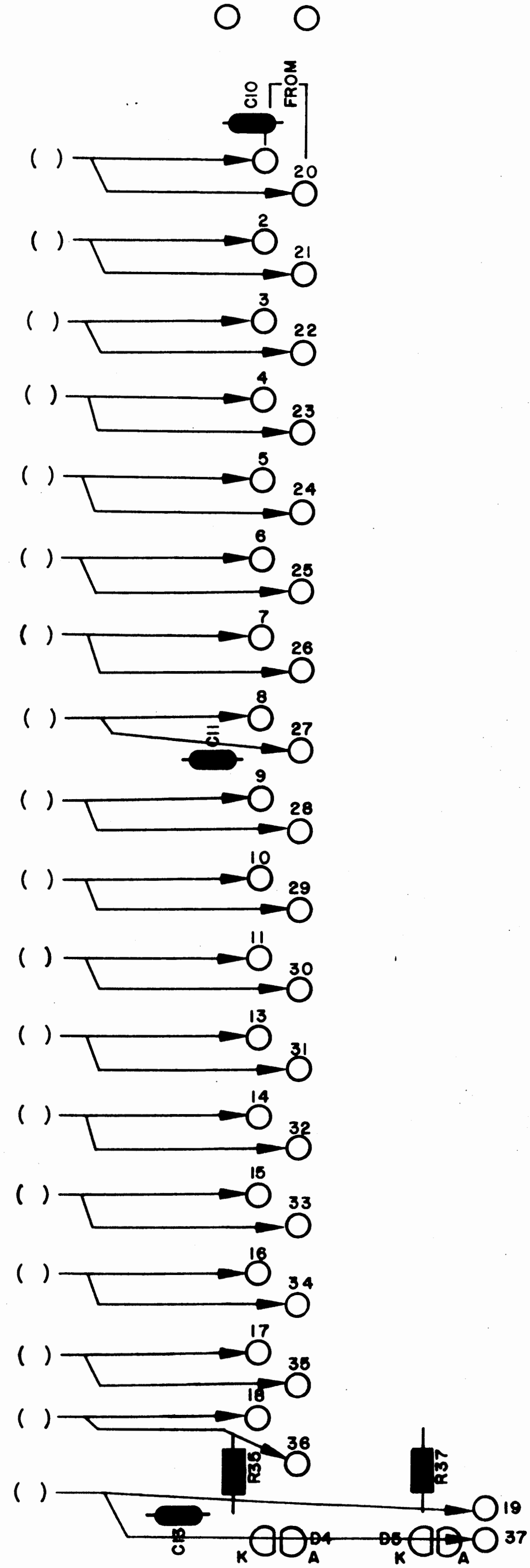
Check your work as you go along and be <u>sure</u> that 1 is connected to 1, 2 to 2, 3 to 3, etc., because corrections will be very difficult later.

Use this procedure to assemble both of the 18 inch cables. Be sure that the MALE 37-pin connector goes to the pads labeled "TO" and the FEMALE connector to the pads labeled "FROM". Refer to the drawing on page to get a rough idea of how these and the next cable will appear when connected to the board.

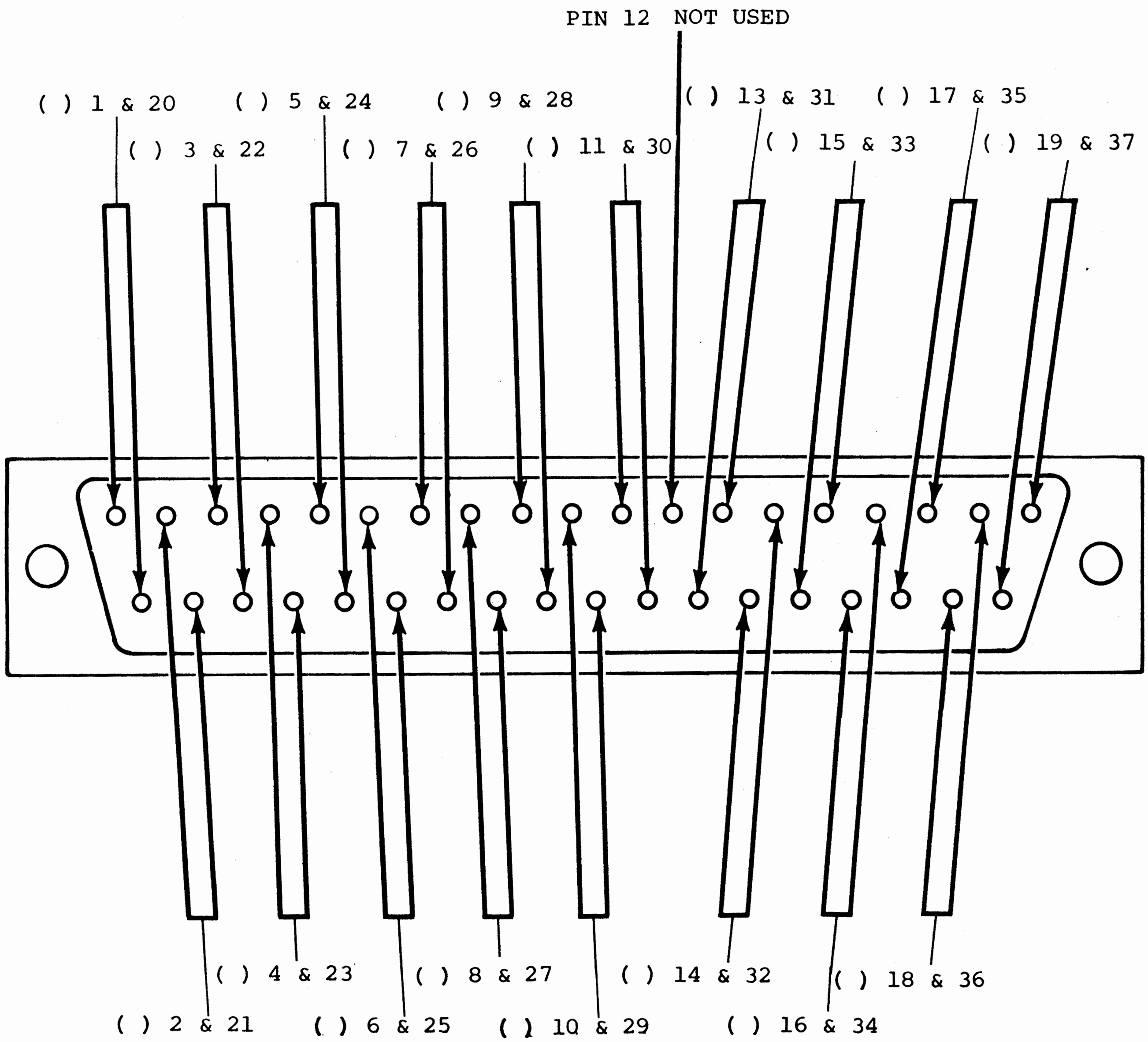


37-PIN MALE CONNECTOR




C10
FROM
20
2
21
3
22
4
23
5
24
6
25
7
26
8
27
C11
9
28
10
29
11
30
13
31
14
32
15
33
16
34
17
35
18
36
R36
R37
19
37
C13
D4
D5
K
A
K
A

37-PIN FEMALE CONNECTOR



Due to its complexity, the 25 inch length of ribbon cable will be assembled in a slightly different manner.

The following two pages contain drawings of one end of the ribbon cable and the 44-pin edge connector included with this kit. These connections, on one end of the ribbon cable only, will be made first.

> <u>NOTE</u>: Be sure to observe that the orientation of the edge connector is not the same in all of the drawings. Use the pin designations themselves for any reference when making connections.

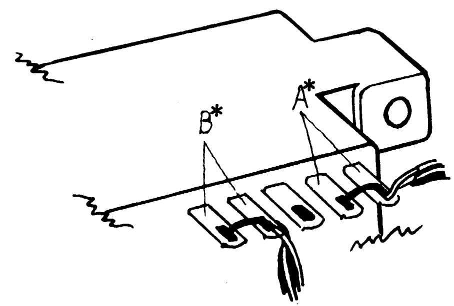
Orient the 25 inch ribbon cable as shown in the drawing on page **46**. The end that is shown at the top of this drawing will be attached to the 44-pin edge connector. The Connection Chart on the following page also refers to this drawing for the proper orientation. Twisted-pair #1 is the pair furthest to the right in the drawing, and pair #18 is furthest to the left. It is very important to begin numbering from the correct side when making the connections.

The Connection Chart on the following page indicates where on the edge connector each twisted-pair should be attached. The pin designations in the chart and in the drawings refer to those stamped into the plastic of the connector itself. Be sure that you connect the proper wires to the correct pins according to the designations stamped on the connector.

In most cases a single wire will connect to a single pin on the connector. Make these connections by first making a good mechanical connection, and then soldering the wire into place. Be careful not to leave any solder bridges, or to melt any insulation.

For twisted-pair #12, and pair #13, you will connect both wires of the pair to a single pin instead of each to a separate one.

For twisted-pairs #15 & #16, all four of the wires should first be twisted together and then all four attached to both of the pins A & B. Do the same for pairs #17 & #18 to connect them to pins D & E. Be sure that there is a solid electrical connection between both of the pins in each case. (see drawing below)



A*=pairs #15 & #16
B*=pairs #17 & #18

Be sure to check-off the appropriate space on the chart as you make each of the connections.

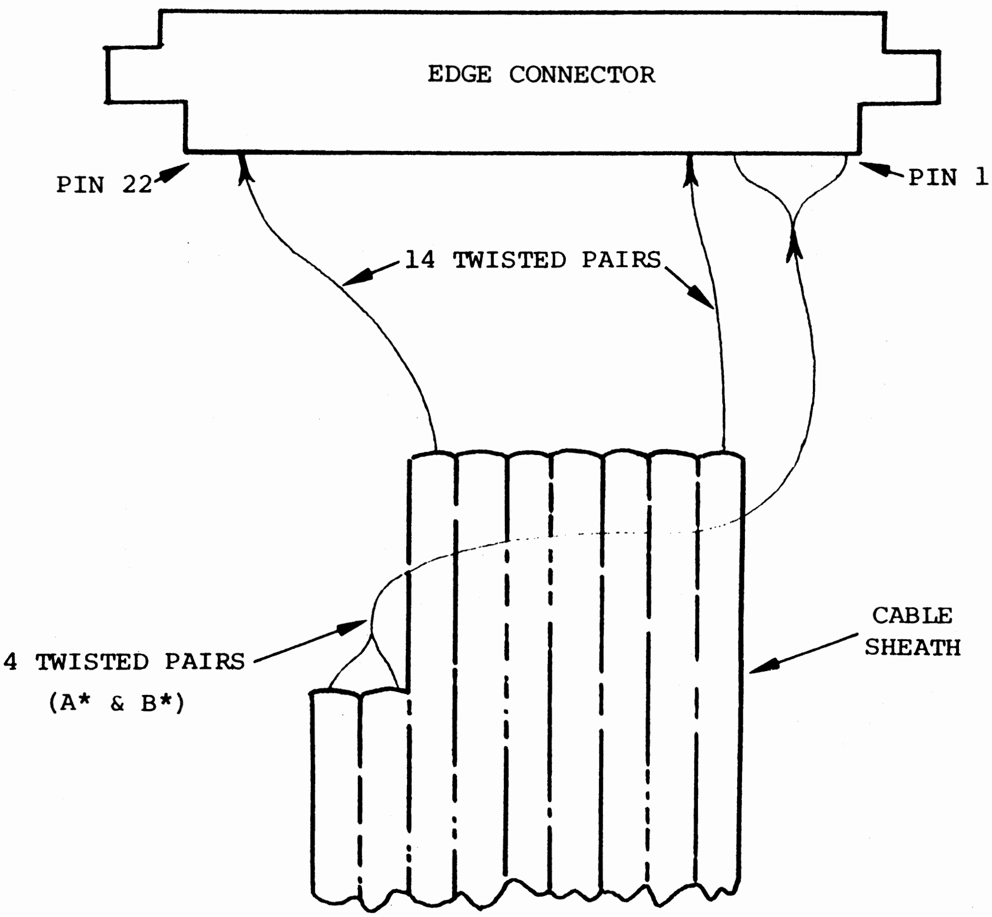
Use a small piece of ribbon cable wire to connect pin 18 to pin V on the edge connector.

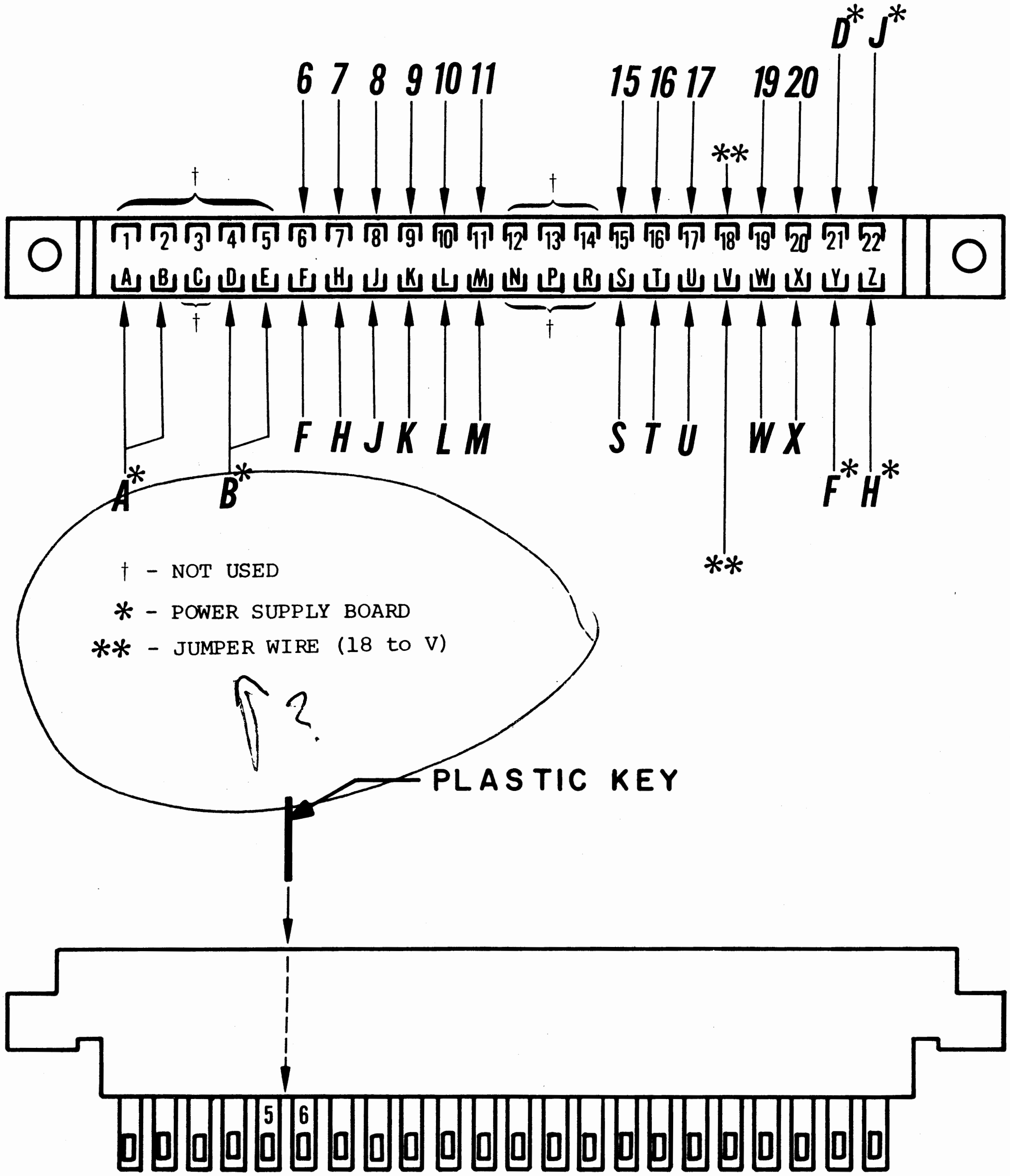
Insert the plastic key, packaged with the edge connector, into the slot between pins 5 & 6 as shown in the drawing on the bottom of page .

CONNECTION CHART

PAIR #	CONNECTOR PIN(S)	PAIR #	CONNECTOR PIN(S)
1	6 & F ( )	10	19 & W ( )
2	7 & H ( )	11	20 & X ( )
3	8 & J ( )	12	21 ( )
4	9 & K ( )	13	Y ( )
5	10 & L ( )	14	22 & Z ( )
6	11 & M ( )	15, 16	A & B ( )
7	15 & S ( )		
8	16 & T ( )	17, 18	D & E ( )
9	17 & U ( )		

( ) Jumper 18 to V




6 7 8 9 10 11
15 16 17
19 20
D* J*
**
†
F H J K L M
S T U
W X
A* B*
F* H*
**
† - NOT USED
* - POWER SUPPLY BOARD
** - JUMPER WIRE (18 to V)
PLASTIC KEY
5 6

The other end of the ribbon cable will connect to both the Disk Buffer board and the Power Supply board.

When making these connections, the same numbering system will be used for the twisted-pairs as previously. That is, the pair furthest to the right in the drawing on page will be refered to as pair #1.

Page contains silk-screen drawings of both PC boards, with arrows to indicate the twisted-pair connections and a space to check-off each as it is completed.

The first eleven twisted-pairs will connect to the remaining row of pads on the Disk Buffer board. Make these connections in the same manner as the previous ribbon cable connections to this board.

Begin with pair #1 and connect one of its wires to pad 6 and the other to pad 7 on the board. Observe the color of the wires connected to the equivalent pins on the edge connector. Be sure you connect pin 6 to pad 6, F to F, etc., as when making the previous connections. Continue the connections through the first eleven of the twisted-pairs in this manner, checking-off the appropriate space as each is completed.

The next seven twisted-pairs will connect to the Power Supply board in nearly the same manner, except that all but two of the connections involve more than one of the wires.

The two wires of pair #12 should be twisted together and both connected to pad D. Pair #13 should connect to pad F in the same manner.

Twisted-pairs #15 & #16 should have all four wires (2 each) twisted together and connected to pad A. Pairs #17 & #18 should be connected to pad B in the same manner.

Only twisted-pair #14 should be separated and connected to pads J & H in the same manner as the first eleven pair.

Make all of the Power Supply board connections as described, checking-off the appropriate space as you complete each of them.

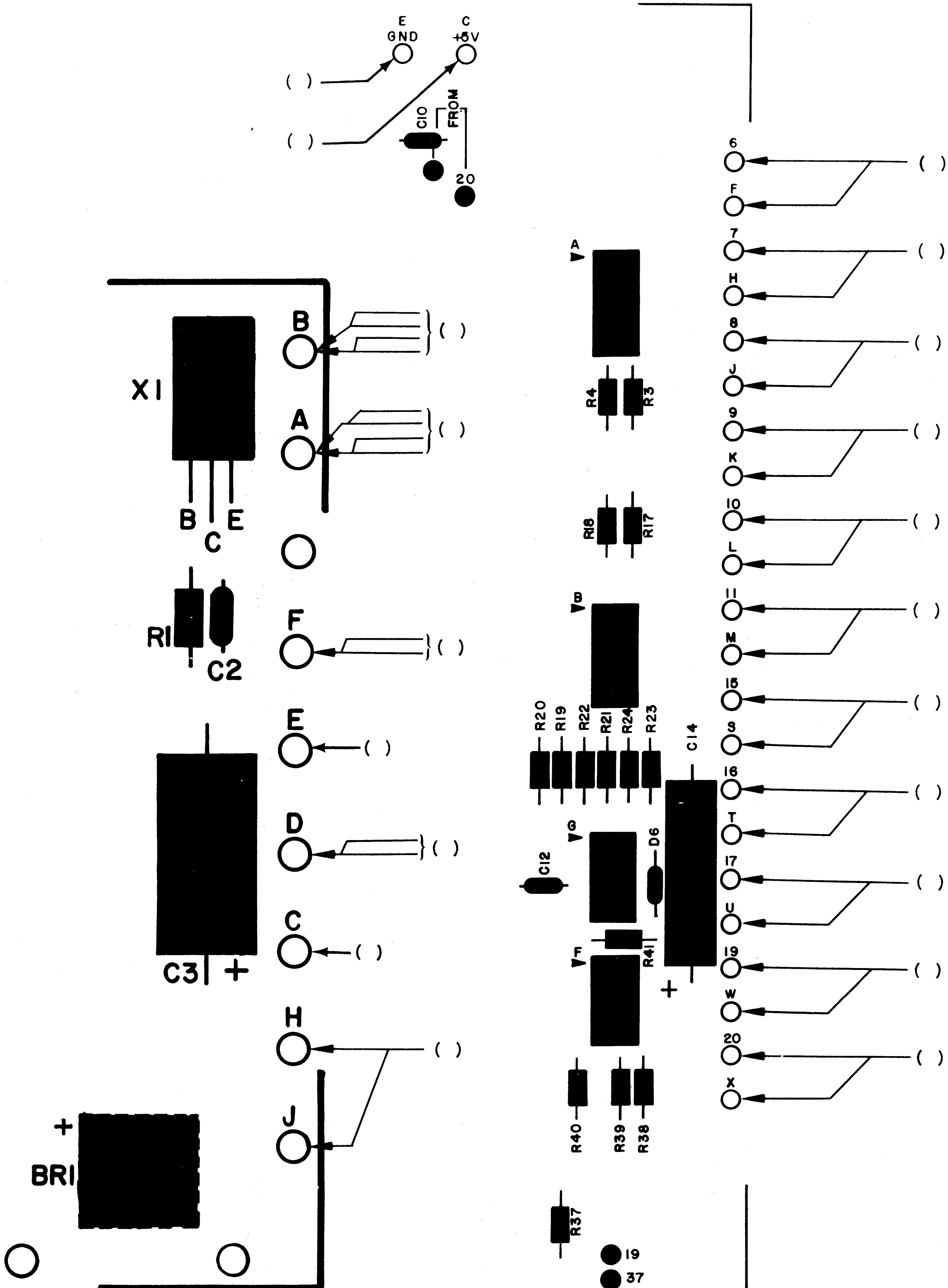
Starting approximately 1 inch from the cable casing, and moving along the Power Supply cable wires, attach a tie-wrap approximately every inch until 5 of them are used. Do these as necessary to make a neat, tight cable.

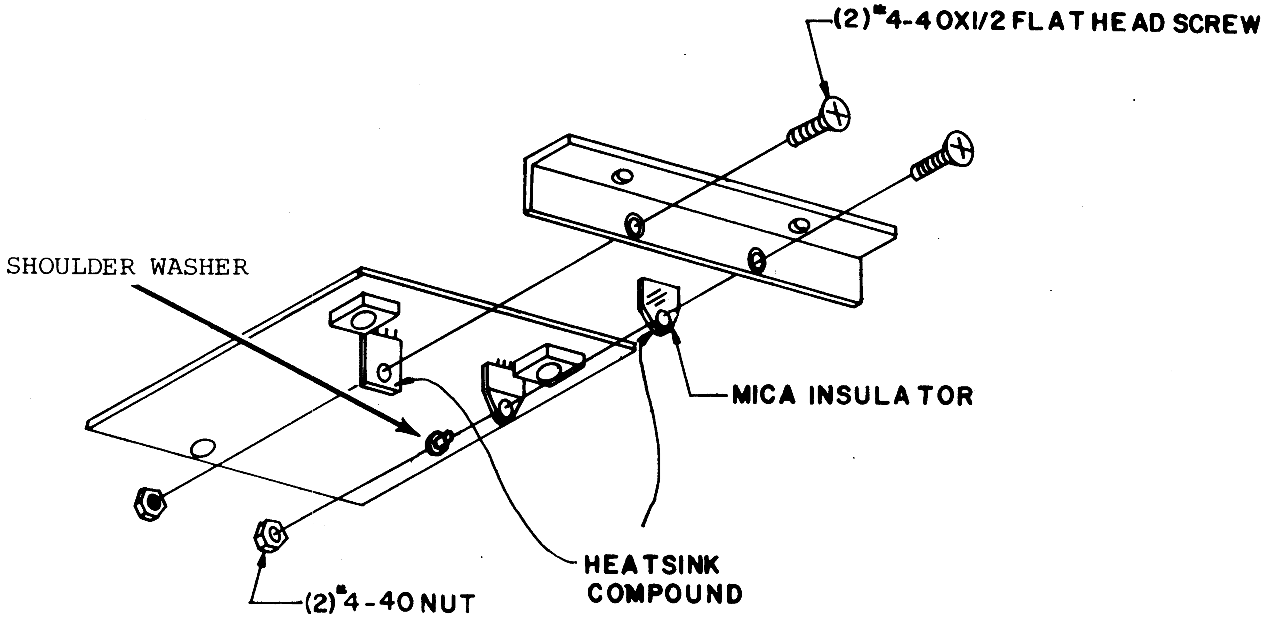
There are two other wires which should be installed at this time. Using the same wire that you used when making the connections to the terminal block, cut one 8 inch length of orange wire and one 8 inch length of black wire. Strip 1/4 inch of insulation from both ends on each of them and tin the exposed portion.

Connect the orange wire between pad C on the Power Supply board and pad C on the buffer board. +5V

Insert the wire from the silk-screened side of the board and solder it on the bottom.

Connect the black wire between E and E in the same manner. GND


E
GND
C
+5V
C10
FROM
20
X1
B
C
E
B
A
R1
C2
F
E
D
C
C3
+
H
J
+
BR1
A
R4
R3
R18
R17
B
R20
R19
R22
R21
R24
R23
C14
G
D6
C12
F
R41
+
R40
R39
R38
R37
19
37
6
F
7
H
8
J
9
K
10
L
11
M
15
S
16
T
17
U
19
W
20
X



VOLTAGE REGULATOR
INSTALLATION

The next two components will be mounted on the bottom side of the Power Supply board. These components will also be mounted to the 90° angle bracket, as with BR1 & BR2, in the two remaining holes.

When installing these components refer to the drawing above and orient them so that the markings on the components face away from the bracket.

Insert the two regulators from the bottom side of the board as shown. *Use heatsink compound between all mating surfaces. Be sure to place the mica insulating washer between Q1 and the bracket, and the shoulder washer between Q1 and the mounting nut.

Tighten the mounting screws firmly, being sure not to twist the component leads as you do so.

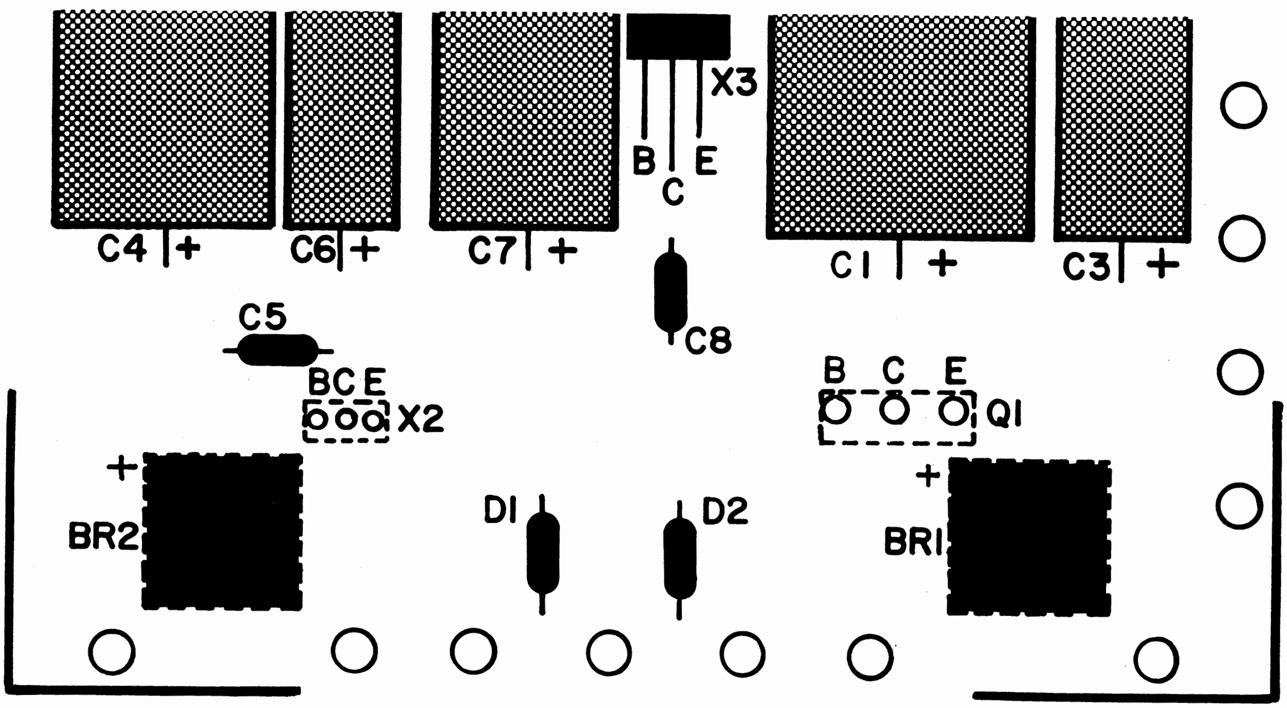
Solder all three leads of both components to the board on the silk-screened side.

Clip off the excess lead lengths; then remove the two screws used earlier to mount BR1 & BR2. The screws mounting X2 & Q1 should remain.

VOLTAGE REGULATOR INSTALLATION

( ) X2 = 7805

( ) Q1 - TIP 145 (w/Mica insulating washer and shoulder washer)



DISK CHASSIS ASSEMBLY

The next step in the assembly procedure is to prepare the chassis itself for mounting the boards and drive unit.

1) Refering to the drawing on the following page, mount the cross beam as shown using the existing screws now holding it in place. Note the number of holes for proper placement.

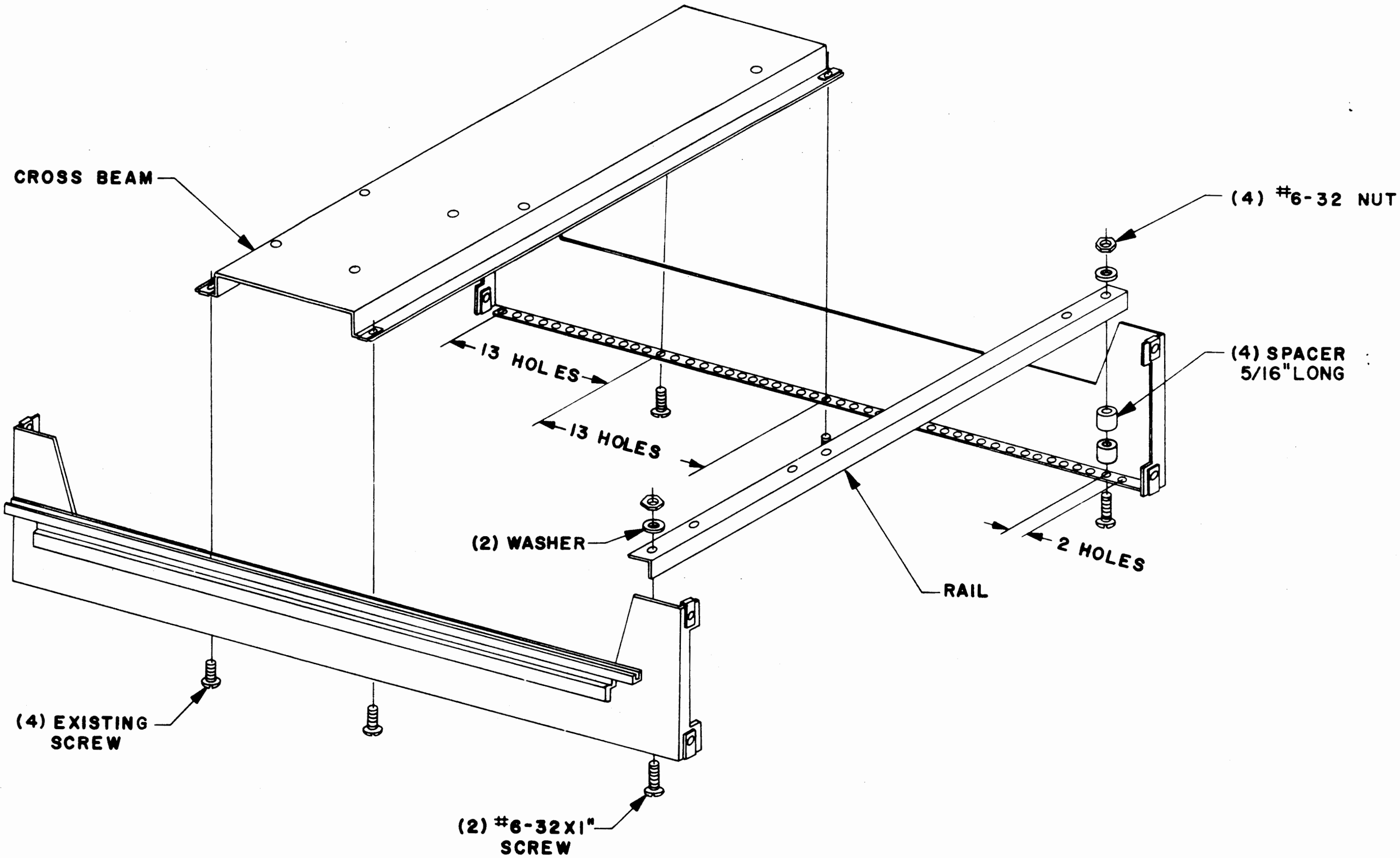
2) To make the following procedures as simple as possible, remove the front panels at this time. Save the screws used to mount the panel to the chassis.

3) Refering to the same drawing again, mount the rail as shown in the 2nd hole from the front. Be sure to include the 2 spacers as shown on each side.

There are 6 additional screws to be added to the chassis members, 4 on the beam and 2 on the rail.

4) Install two #6-32 x 3/4" screws onto the rail in the positions indicated on the same drawing. Insert them from the bottom and tighten them firmly using #6-32 lockwashers and nuts.

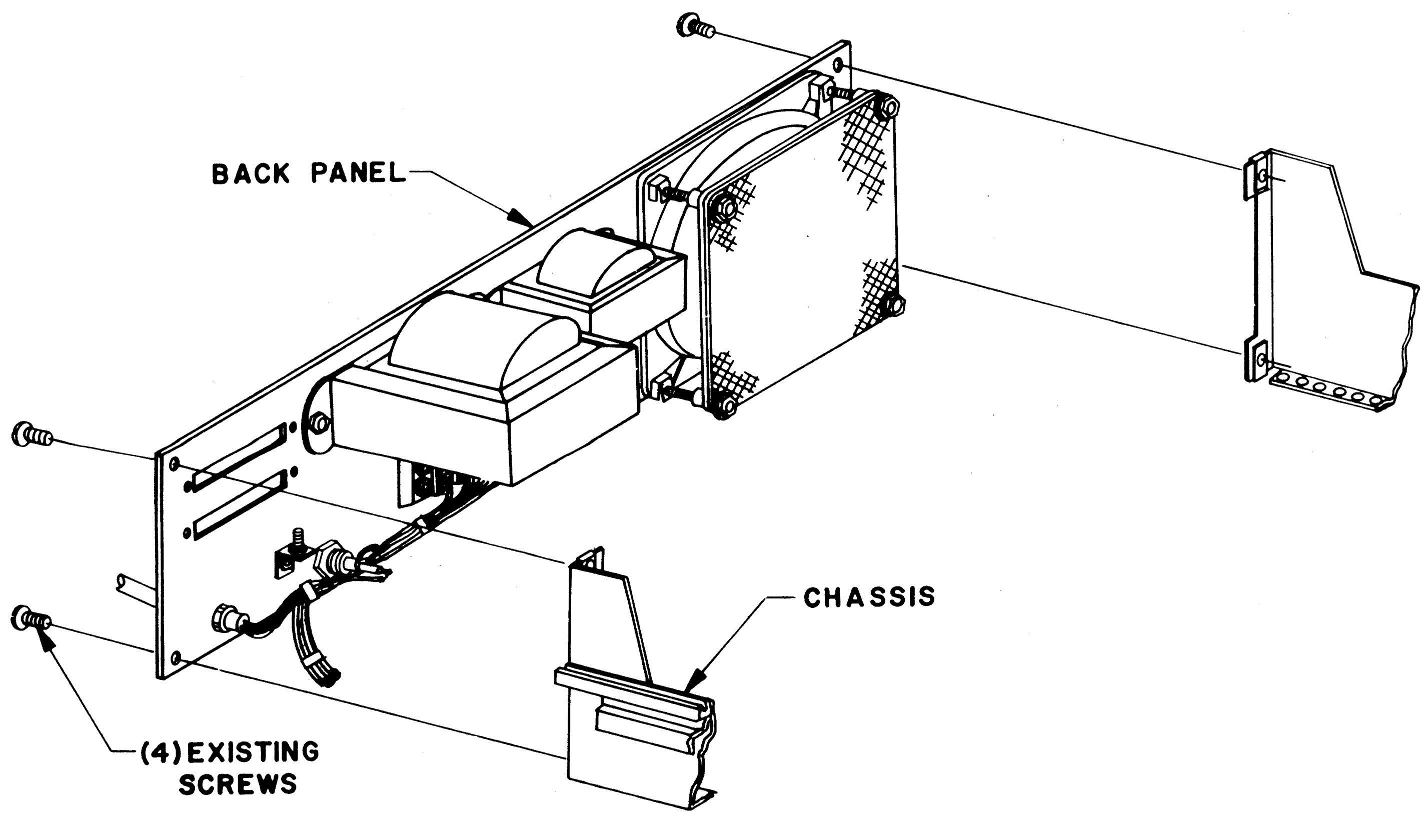
5) Install two 4-40 x 1" screws and two 6-32 x 1" screws on the cross beam as shown using the indicated hardware.


CROSS BEAM
(4) #6-32 NUT
13 HOLES
13 HOLES
(4) SPACER
5/16" LONG
(2) WASHER
2 HOLES
RAIL
(4) EXISTING
SCREW
(2) #6-32X1"
SCREW

BACK PANEL MOUNTING

Mount the back panel to the rear of the chassis as shown below using the same screws previously used to mount it.

Be careful not to catch any wires between the chassis and the panel.

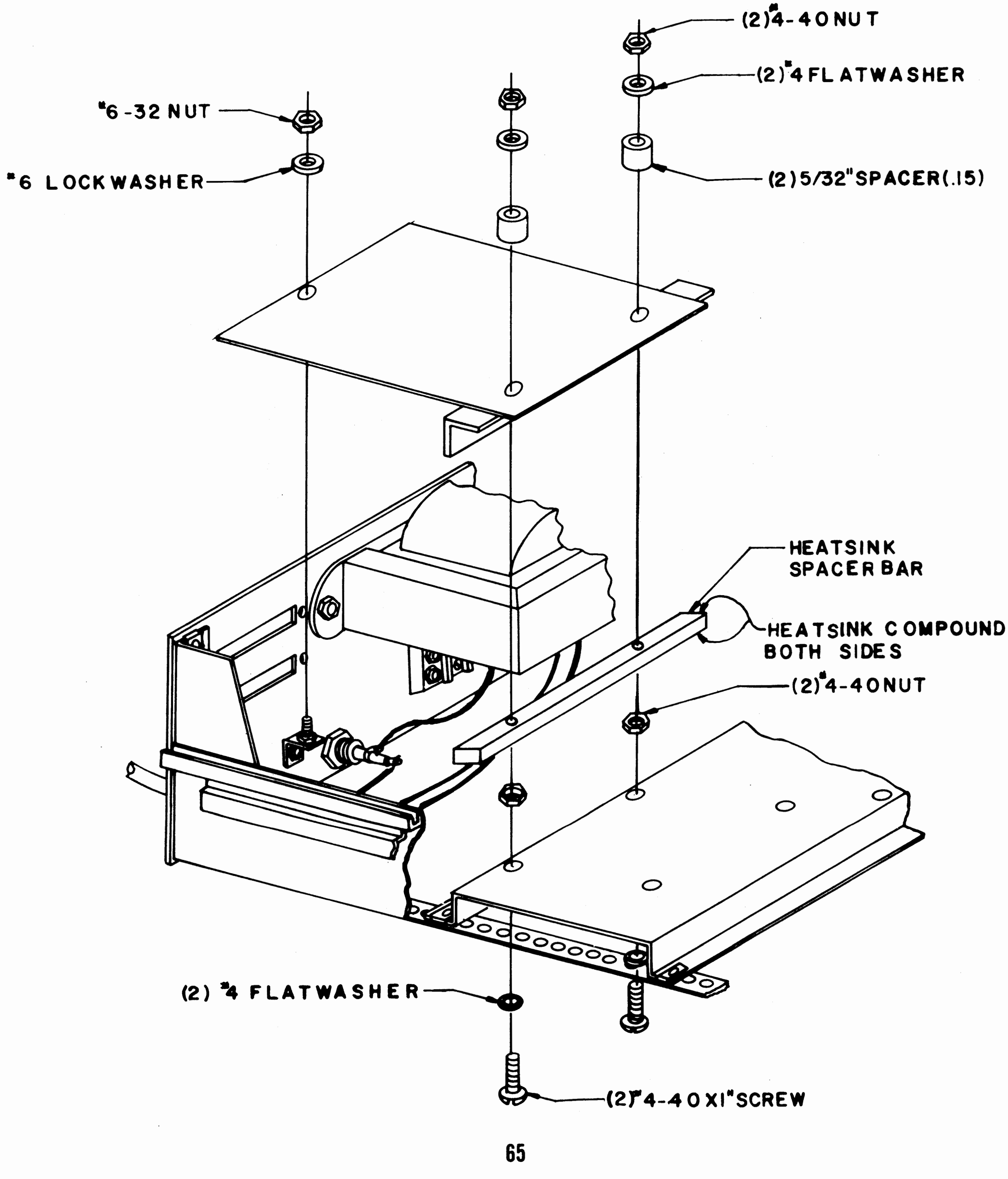


POWER SUPPLY BOARD MOUNTING

Refering to the drawing on the following page, mount the Power Supply board to the 90° angle clip and bracket as shown. Study the drawing carefully before beginning.

NOTE: The #4-40 screw shown are those installed earlier.

Be careful not to disturb the wire connects previously made between this board and the buffer board and cables.

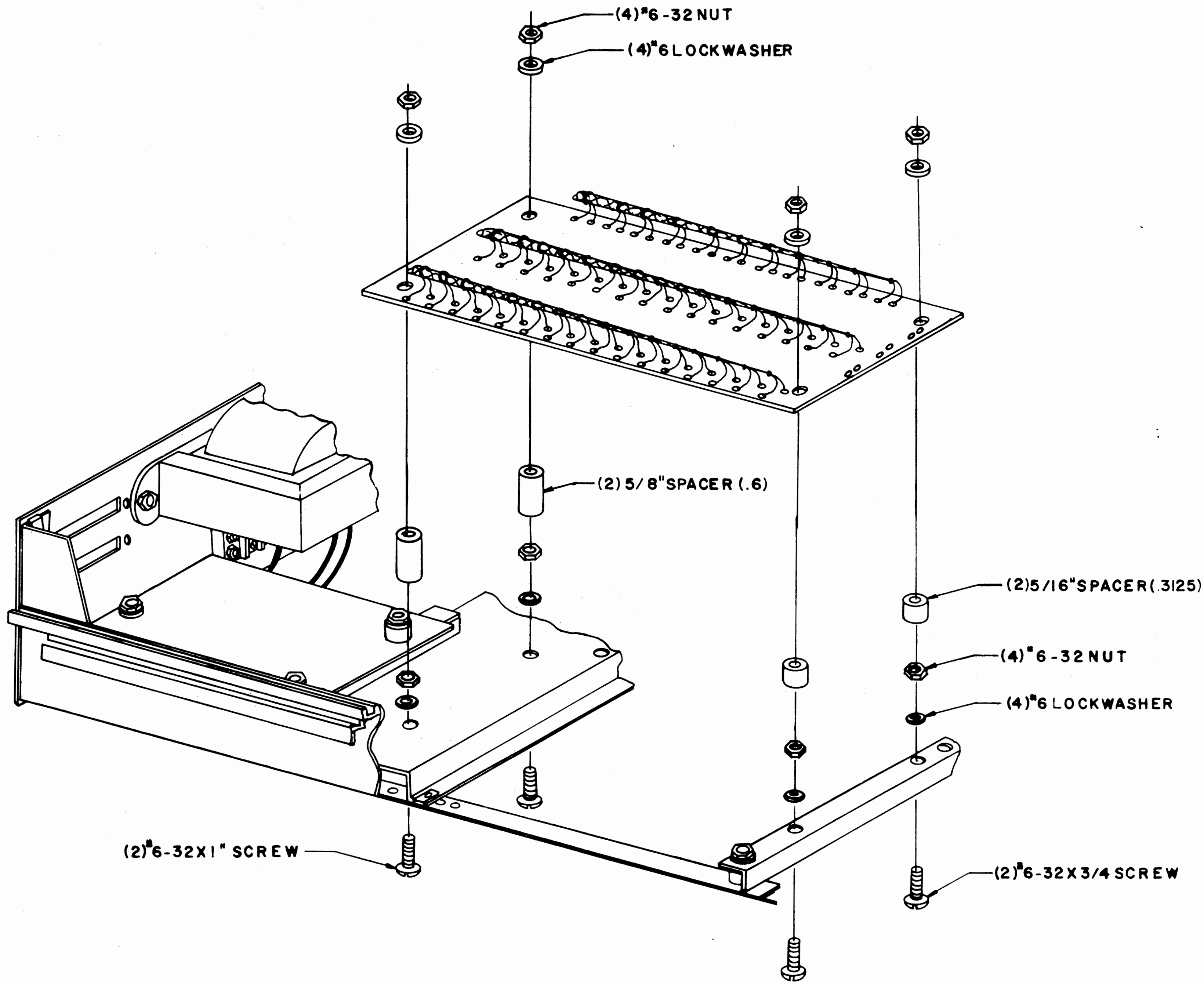

(2) #4-40 NUT
(2) #4 FLATWASHER
#6-32 NUT
#6 LOCKWASHER
(2) 5/32" SPACER (.15)
HEATSINK
SPACER BAR
HEATSINK COMPOUND
BOTH SIDES
(2) #4-40 NUT
(2) #4 FLATWASHER
(2) #4-40 X1" SCREW

DISK BUFFER BOARD MOUNTING

Refering to the drawing on the following page, mount the Disk Buffer board as shown.

Again, study the drawing carefully before beginning. The screws shown have already been installed.

The connectors on the three cables should face towards the back panel.



(4)#6-32 NUT
(4)#6 LOCKWASHER
(2) 5/8" SPACER (.6)
(2) 5/16" SPACER (.3125)
(4)#6-32 NUT
(4)#6 LOCKWASHER
(2)#6-32X1" SCREW
(2)#6-32X3/4 SCREW

<u>POWER SUPPLY WIRING</u>

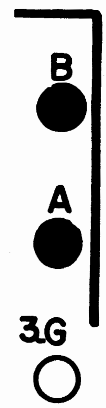
Refering to the silk-screen drawing below, and the wiring diagram on the following page, connect the wires from the terminal block to the pads on the Power Supply board.

Use the following procedure:

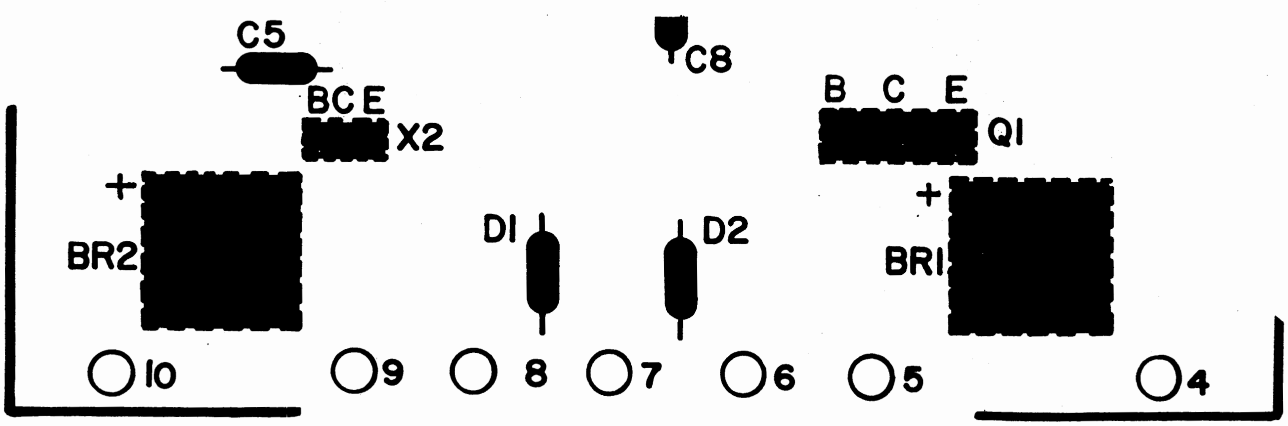
1) All of the wires should be connected to the pads on the board marked with the same designation as the tags placed on them earlier.

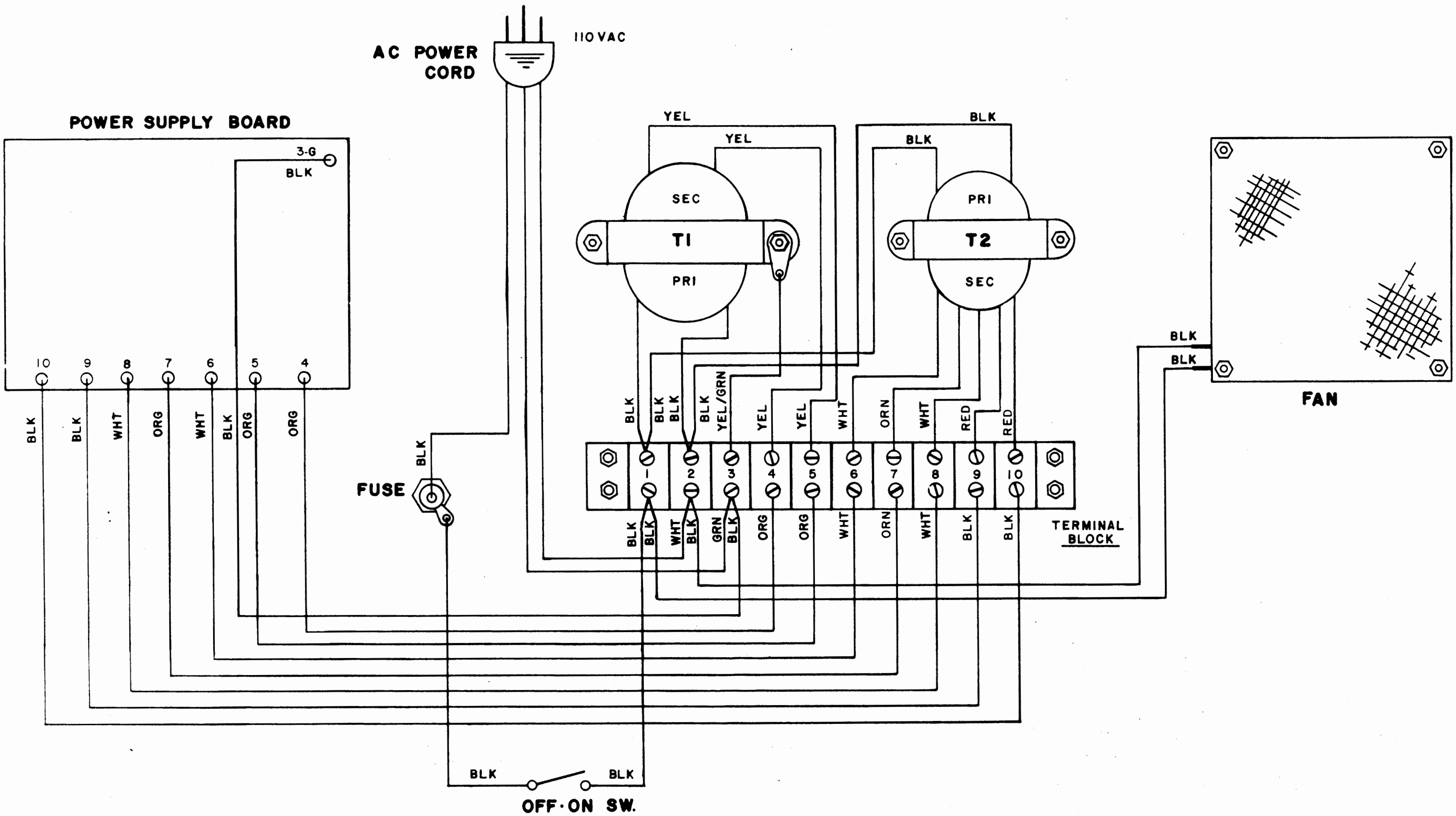
2) Insert all of the wires from the silk-screened side of the board, <u>almost</u> to the insulation. Add solder from the same side of the board except wire "3-G", and then continue applying heat while pushing the wires down as far as possible until the insulation just touches the solder. Be careful not to melt any insulation.

3) Turn the board over to solder wire "3-G" and then clip off all excess lead lengths.



Check this wiring over again carefully, and then remove the tags from the wires.





ALTAIR FLOPPY DISK DRIVE

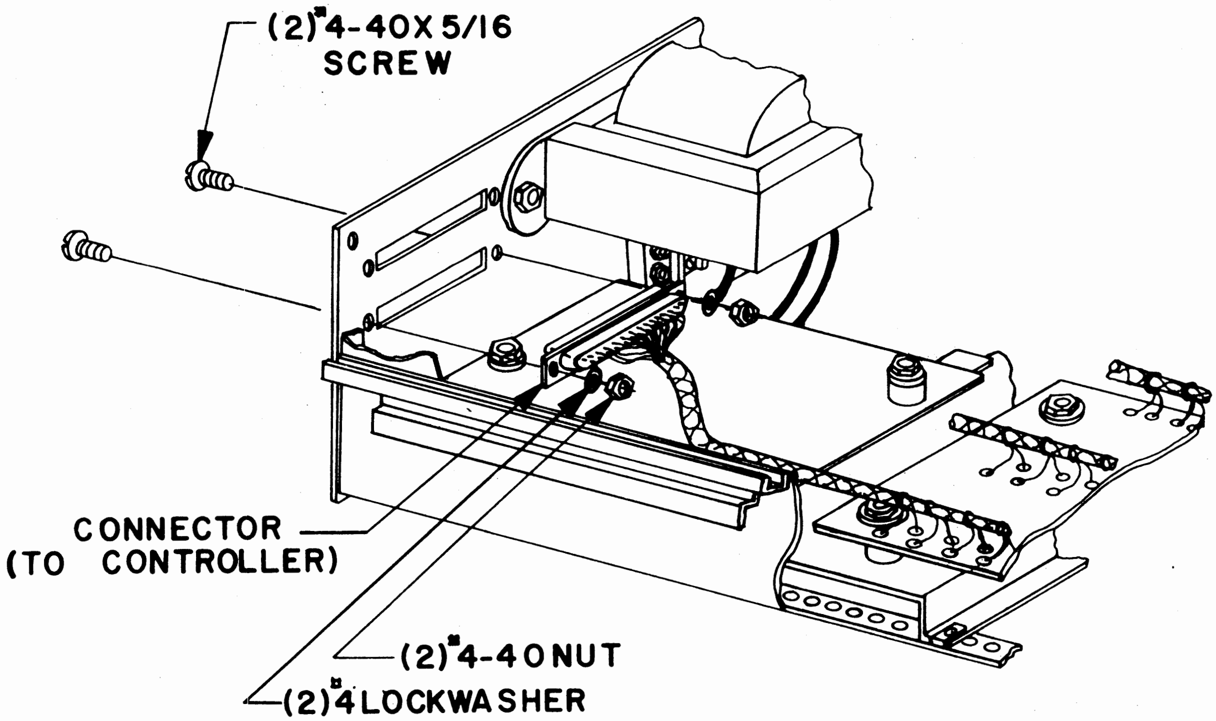
POWER SUPPLY WIRING DIAGRAM

CONNECTOR MOUNTING

Refering to the drawing below, mount the two 37-pin connectors to the back panel as shown.

Be sure to mount the male connector into the slot labeled "TO" and the female connector into the slot labeled "FROM".

On both connectors pin 1 should be towards the top.

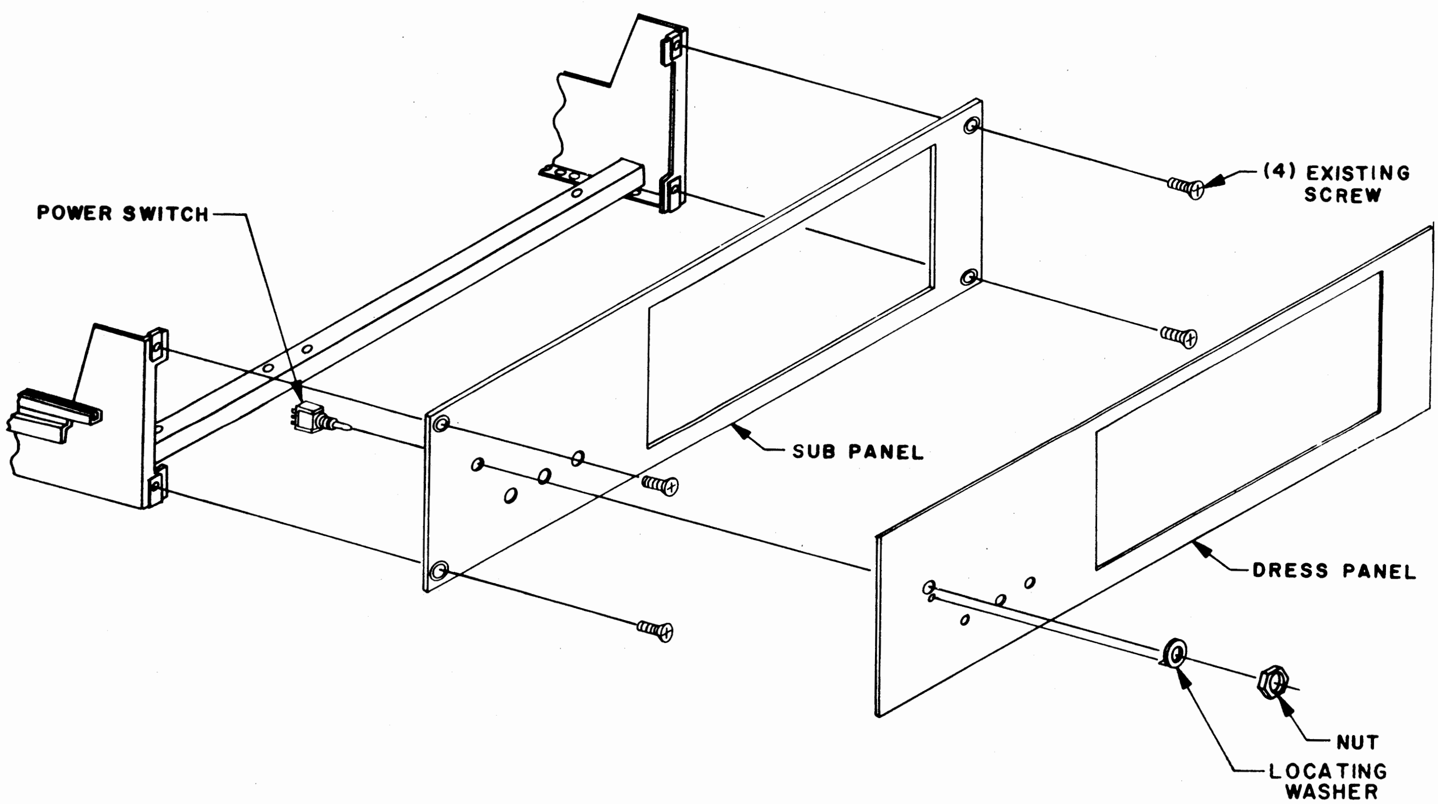


FRONT PANEL MOUNTING

The front sub panel and dress panel can now be re-installed. Use the same four screws previously used to mount the sub panel to re-mount it to the chassis as shown in the drawing below.

Note when setting the dress panel in place that it is a "floating" panel. Installing the power switch, as shown, at this time will temporarily hold it in place.

Be sure the lettering on the dress panel is facing outwards.



LED INSTALLATION

There are three RL-21 Light-Emitting-Diodes (LED's) to be installed on the Disk Buffer Board. These LED's have a cathode and anode lead on each of them which must be properly oriented for installation on the board. The diagram below shows you how to determine the cathode and anode leads of an RL-21. Hold the LED up to a light and you will be able to see inside. The larger of the two elements inside the plastic casing is the cathode.

The silk-screen on the board itself has the cathode leads for the three LED's marked with a "K". The anode lead is marked with an "A". When you install these components, make sure that the cathode leads are in the pads marked "K" and the anode leads in the pads marked "A". Improper orientation when installing LED's may cause permanent damage to the component.

As is shown in the drawing on this page, these three components also require special spacing and bending of the leads in order to fit the unit properly.

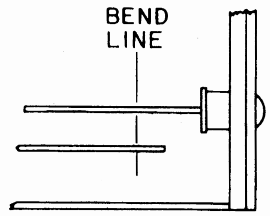
1) Set the LED's in place one at a time and bend as necessary to fit as shown in drawing [3].

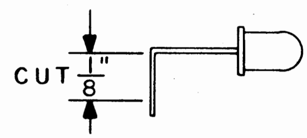
2) Cut the leads as shown in [2] and place the LED's on the board properly.

3) Solder them in place from the top side of the board. LED's are very heat sensitive, so use a minimum of heat for the shortest amount of time possible to make the connection.

When properly installed, the LED's should fit as shown in the drawing below.

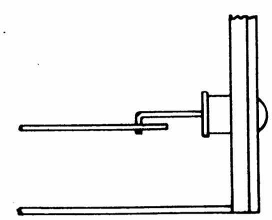
[1] SET THE LED IN PLACE AND MARK THE LEADS





[2] CUT THE EXCESS LEAD TO LEAVE 1/8 INCH

[3] SOLDER TO FIT IN PLACE AS SHOWN



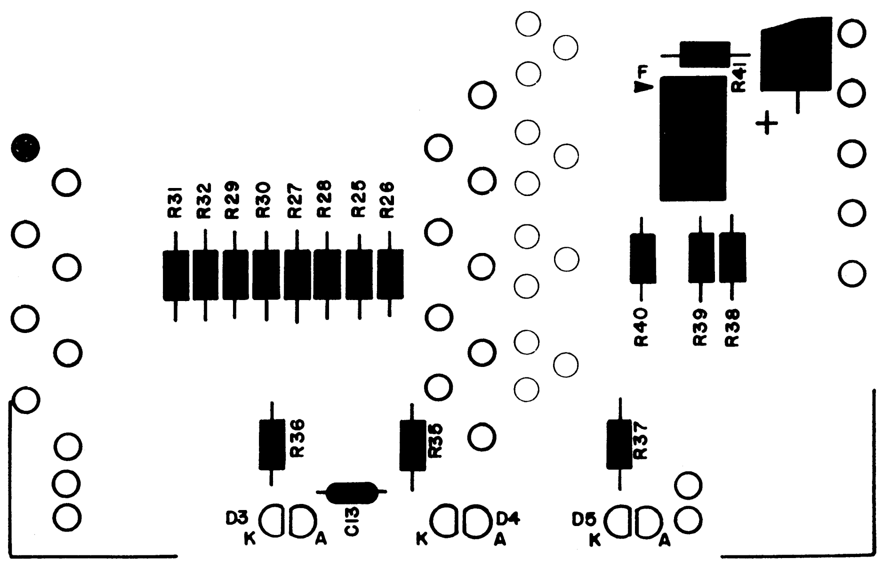
WARNING: RL-21 LED's are very sensitive to heat. Use a minimum application of heat with your iron when making these solder connections.

LED Installation

( ) D3 = RL-21 LED

( ) D4 = RL-21 LED

( ) D5 = RL-21 LED



DISK DRIVE UNIT INSTALLATION

The Disk Drive unit itself can now be installed into the chassis.

1) The first step in this process is to set the chassis on end, with the front panel facing upwards.

2) Remove the screws and rubber feet that were factory installed on the bottom of the drive unit.

3) Being careful not to catch any of the wires or cables, slowly lower the drive unit into the chassis. Refer to the drawing on the following page for the proper orientation.

4) Refering again to the drawing on the following page, insert the two mounting screws and lockwashers on the front side of the drive unit. Do not tighten the screws down at this time.

5) Refering to the same drawing, install the spacer bar and mounting hardware for the rear end of the drive unit.

   Tighten all four mounting screws firmly.

6) The 44-pin edge connector should now be plugged into the rear of the drive unit. Line up the connector with the finger pads on the units PC board and align the plastic key between pins 5 & 6 with the slot in the board. Push the connector firmly into place.

Insert Page

ALTAIR FLOPPY DISK

Disk Drive Assembly Procedure

addendum to page 74, Disk Drive Unit Installation

A. Before beginning the steps listed on page 74, the mounting holes in the Disk Drive Unit must be threaded. Use the following procedure to thread the four mounting holes:

1. Place the unit upside down. Place a strip of masking tape under each mounting hole to catch any metal particles.

2. Install a #6-32 x 3/4" self tapping screw (MITS part number 100957, Bag 7) into each hole.

3. Remove the screws and the masking tape.

B. Step #2 of the instructions given on page 74 may be omitted.

C. If difficulty is encountered while installing the Disk Drive Unit into the chassis, remove the upper right-hand flat heat screw near the bezel on the Disk Drive Unit. When the Disk Drive Unit has been properly installed, be sure to put the flat head screw back into place.

Insert Page

ALTAIR FLOPPY DISK

Disk Drive Assembly Manual

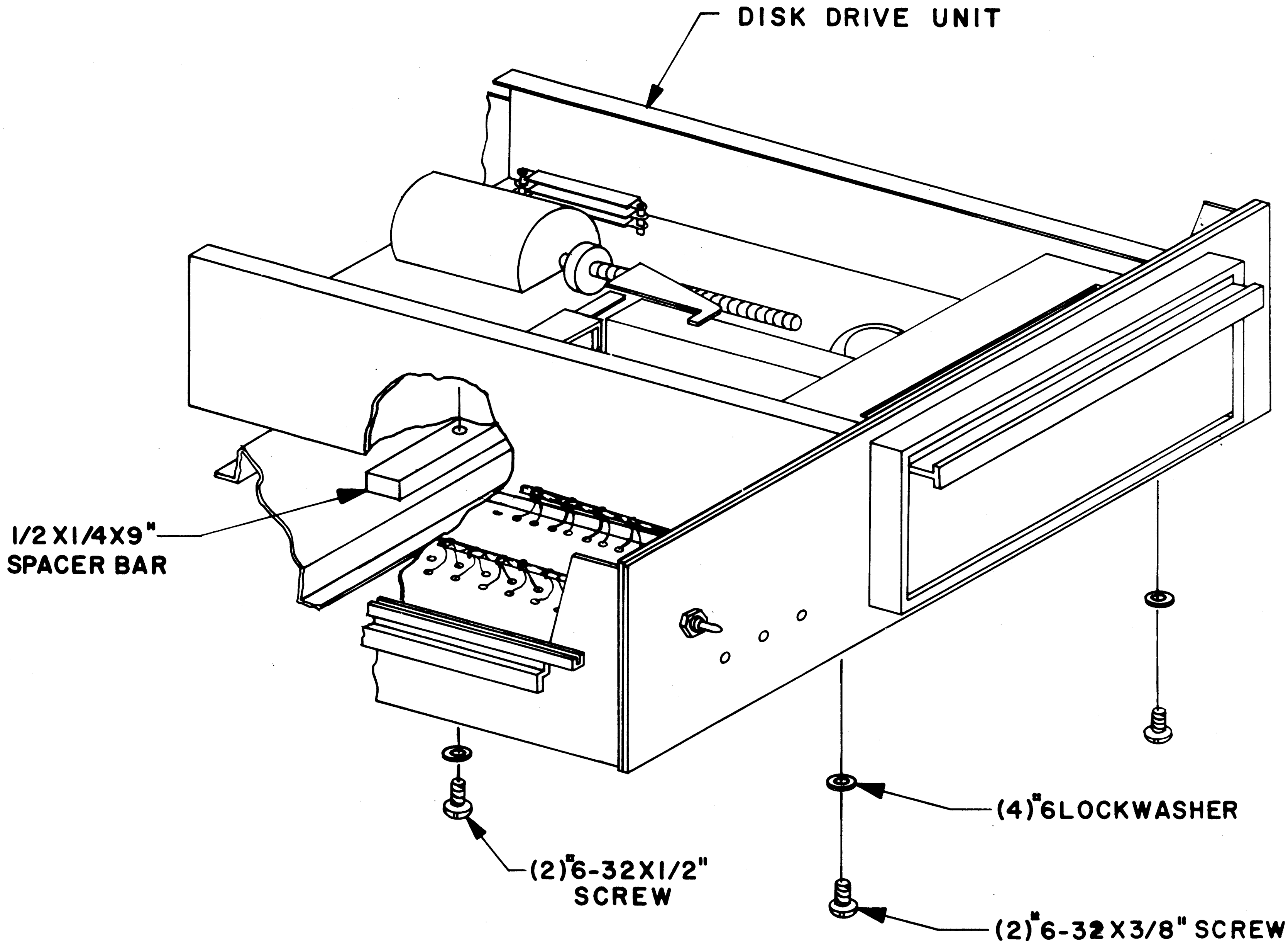
Addendum to page 75, Disk Drive Unit Installation

The instructions on page 75 refer to a 1/2" x 1/4" x 9" spacer bar (Part No. 101841) that is to be installed in the rear of the chassis. This spacer bar has been replaced with either a 1/2" x 1/4" x 1" rectangular spacer or a 5/8" x 1/4" round spacer. The new spacer will be installed in the right rear mounting hole only.

This spacer allows the PERTEC FD-400 to be mounted at three points on the chassis, instead of four, thus avoiding the possibility of warping the FD-400 drive chassis.

The part number for the new spacer is 101841.

MITS, Inc.
August, 1976


DISK DRIVE UNIT
1/2 X1/4X9"
SPACER BAR
(2)#6-32X1/2"
SCREW
(4)#6LOCKWASHER
(2)#6-32X3/8" SCREW

DISK DRIVE PRELIMINARY CHECK OUT

1. With no diskette in drive and the chassis unit not installed in cabinet, and no address jumpers installed, turn power on.

   A) Fan and disk drive motor should turn.

   B) Power indicator should light.

2. If voltmeter is available, measure:

   A) +24 volt supply at + end of C3 (with respect to chassis) on the power supply board.

   B) +5 volt supply at + end of C6 on the power supply board.

   C) -5 volt supply at point "J" of the power supply board.

   All voltages should be within 5% of rated output. If the disk drive motor does not start up, or the power indicator does not light, or the power supply voltages are wrong, consult the Theory of Operation and recheck wiring.

3. A) With a cliplead, ground to chassis wire #13 (Disk Enable) on the left edge of the buffer board (Pin 13 of "To Controller").

   The Disk Enable light should come on.

   B) Now open disk drive door. The drive motor should stop and Disk Enable light should turn off. Close the door and the motor should start up. 5-10 seconds later, the Disk Enable light should turn on (timing controlled by IC G).

   C) With another cup lead, test the mechanical disk functions by grounding (on the left edge of board)

      1. Wire #8 (Head Load)
         The Head Load solenoid should energize as long as #8 is grounded, and Head Load light should turn on.

      2. Wire #6 (Step In)
         The track stepping motor shaft should turn as point #6 is intermittantly grounded, simulating stepping pulses. The head carriage should move towards the front of the Disk Drive.

      3. Wire #7 (Step Out)
         The track stepping motor shaft should turn as Point #7 is intermittantly grounded, simulating stepping pulses. The head carriage should move towards the rear of the Disk Drive.

   This completes the preliminary check out of the Disk Drive.

   Remove the clip leads, and install the disk address jumpers as indicated on page 77 .

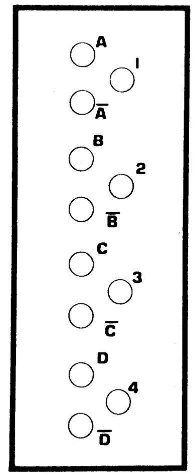
ADDRESS SELECTION

There are four jumper wires to be installed on the buffer board in order to select the I/0 address.

Use component leads saved earlier for this purpose. Install them from the silk-screened side of the board and solder them on either side.

To comply with MITS software, the board should be jumpered to address Ø unless it is a part of a multiple disk drive system.

Refering to the silk-screen drawing on the right, jumper as follows for address Ø:



PAD	TO	PAD
1	----	$\overline{A}$
2	----	$\overline{B}$
3	----	$\overline{C}$
4	----	$\overline{D}$

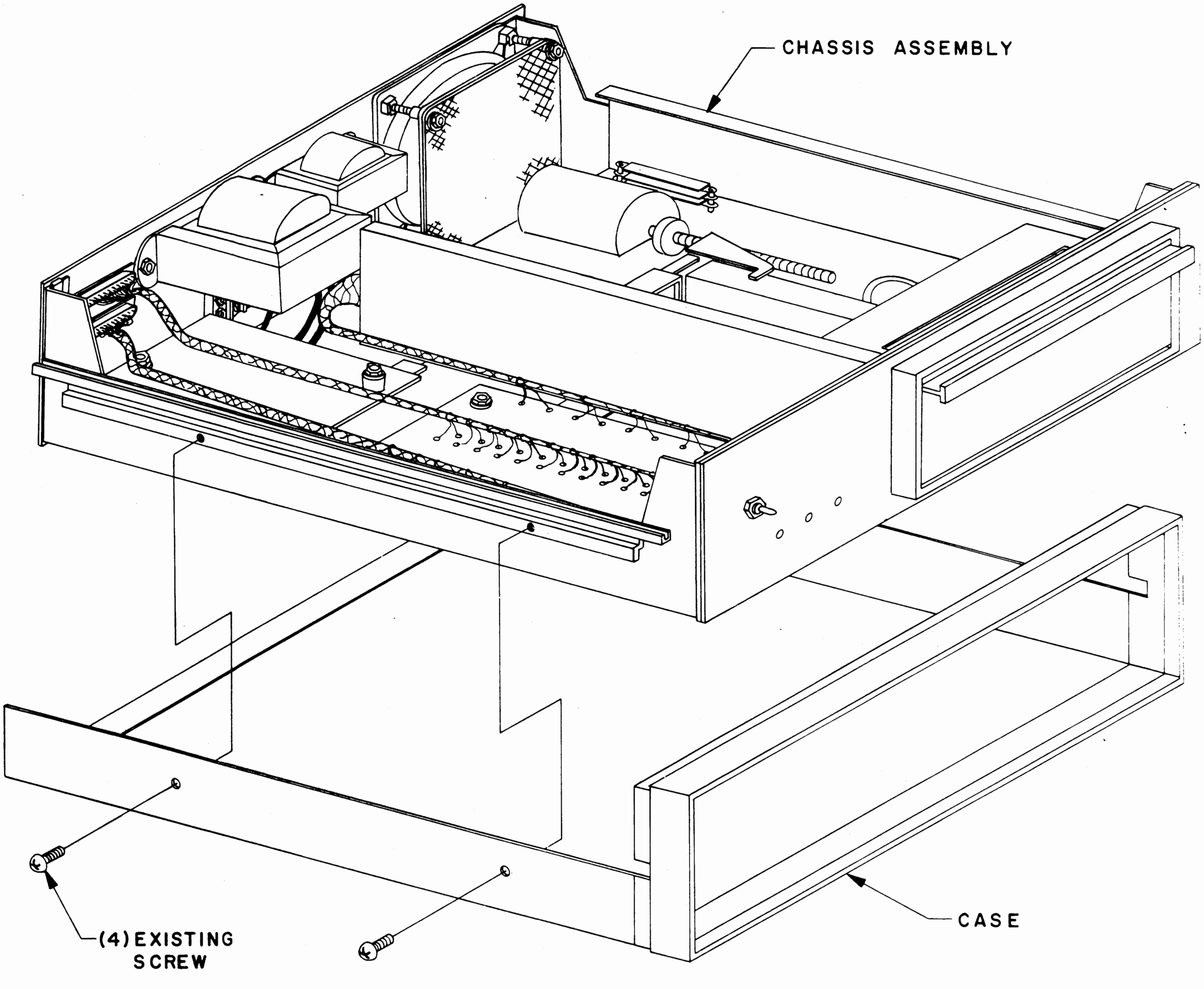
Consult the jumper chart in the Theory of Operation section if a different address is desired.

FINAL ASSEMBLY

The chassis assembly can now be installed into the outer case.

Refer to the drawing on the following page and mount the chassis as shown.

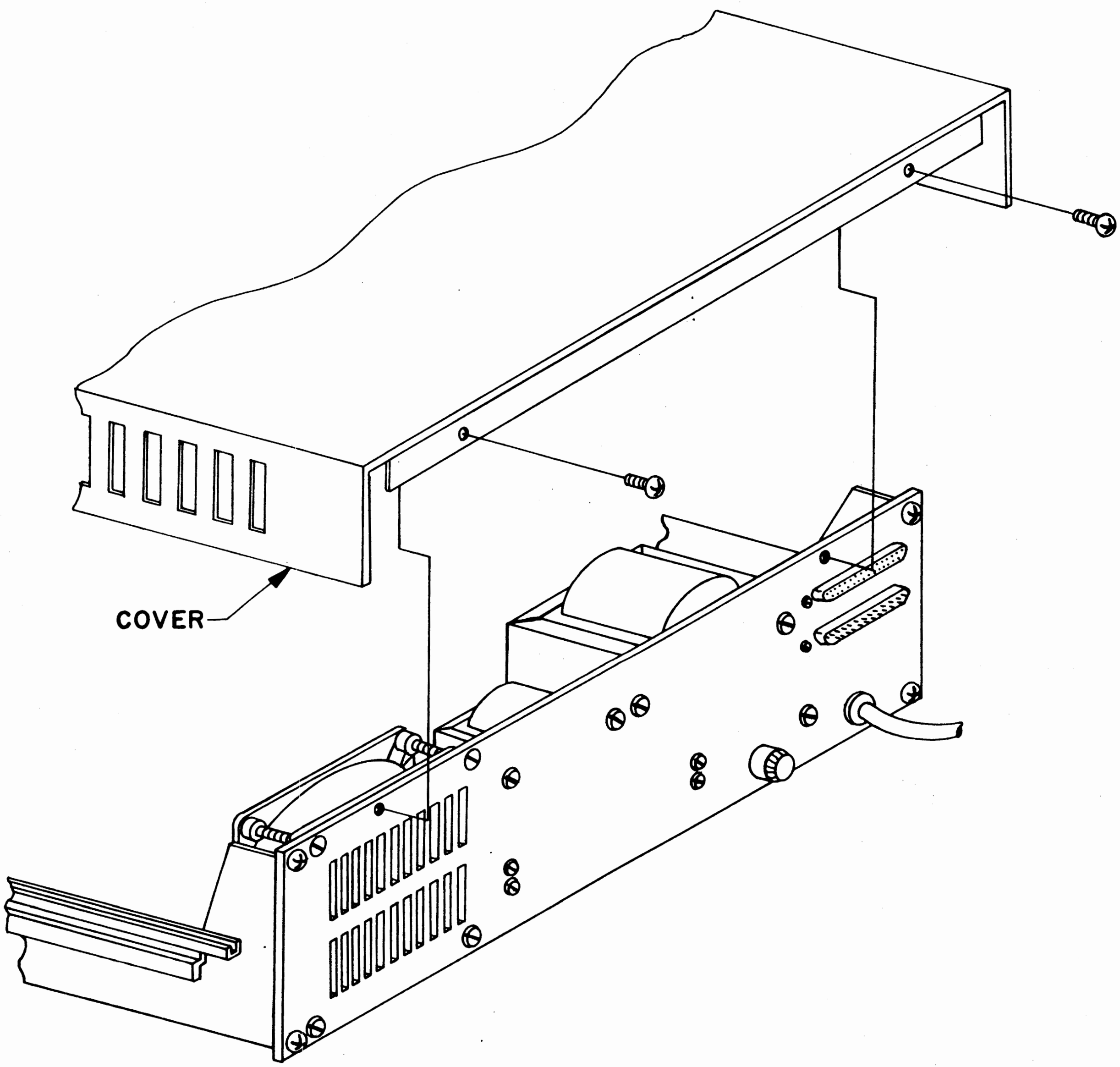
To insert it, start by setting it slightly towards the back of the case, and then slide it forward until the screw holes align. Tighten the four screws firmly.

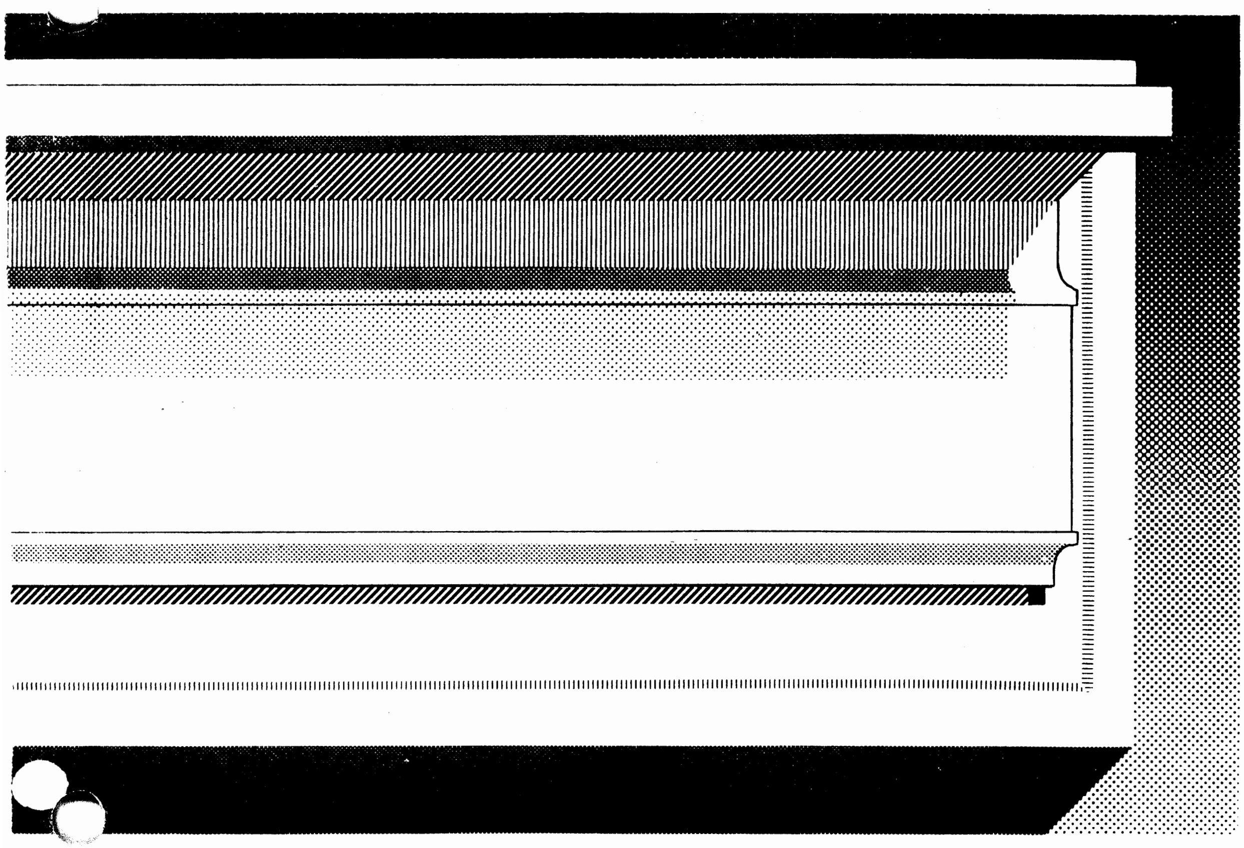

CHASSIS ASSEMBLY
CASE
(4) EXISTING SCREW

CASE TOP INSTALLATION

Re-install the case top onto the unit as shown below. Do not, however, use the same screws which held it originally.

Use #6-32 x 1/4" screws to secure the case top.





disk controller assembly procedure

DISK CONTROLLER ASSEMBLY

The Disk Controller will now be assembled. This consists of two PC boards and interconnecting cables.

The Disk Controller mounts directly into the computer main-chassis and uses two slots.

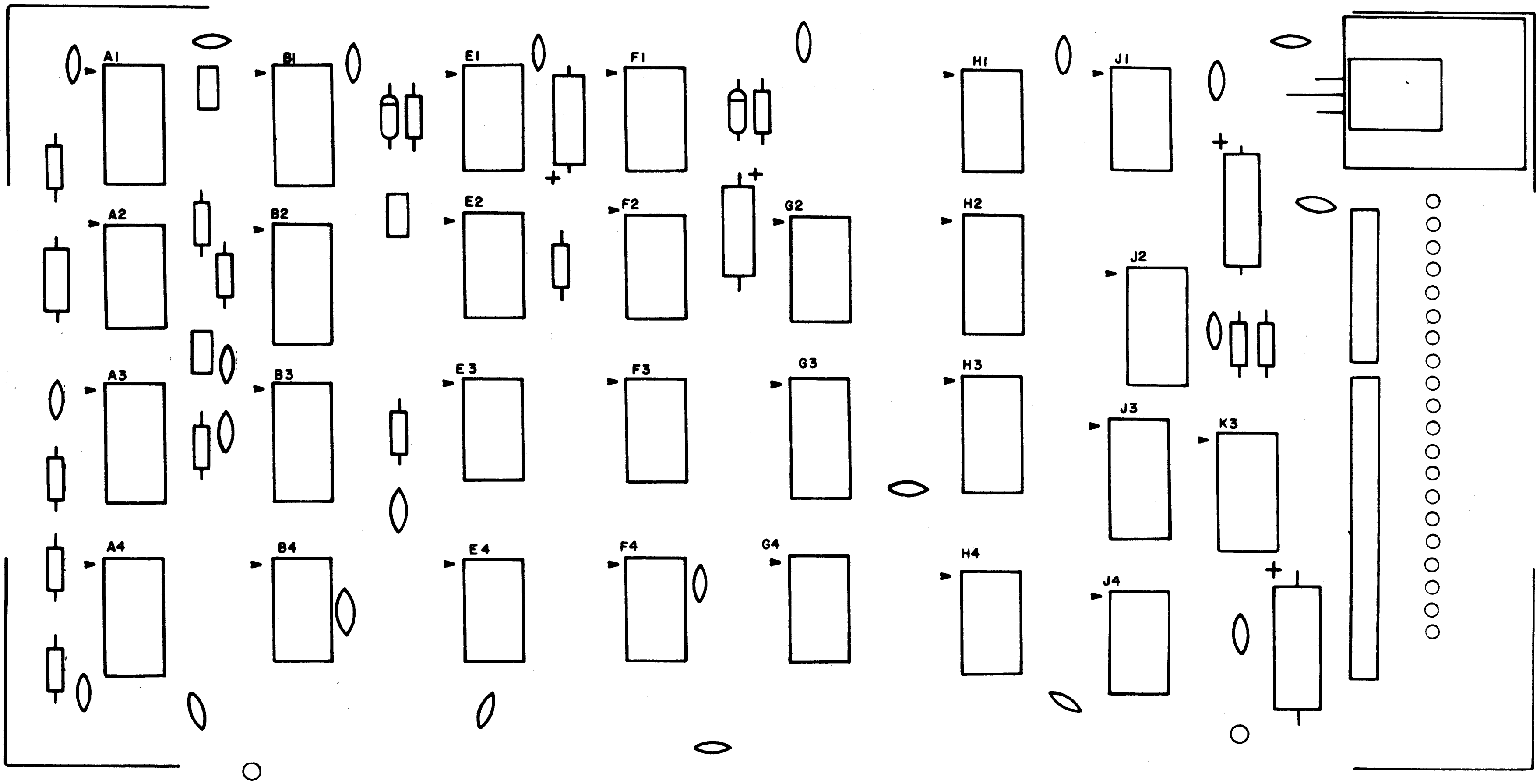
Controller Board #2 will be assembled first.

IC Installation

Install the following 28 ICs according to the instructions on page 4 .

ICs

Silk Screen Designation	Number	Silk Screen Designation	Number
( ) A1	74123	( ) F3	74L02
( ) A2	74L73	( ) F4	74L02
( ) A3	93L16	( ) G2	74L04
( ) A4	93L16	( ) G3	74L75
( ) B1	74123	( ) G4	74L04
( ) B2	74123	( ) H1	74L02
( ) B3	74123	( ) H2	74166
( ) B4	74L04	( ) H3	74L75
( ) E1	74L00	( ) H4	74L04
( ) E2	74L73	( ) J1	74L02
( ) E3	74L00	( ) J2	8T98
( ) E4	74L10	( ) J3	74L75
( ) F1	74L02	( ) J4	74L74
( ) F2	74L73	( ) K3	8T97


A1
B1
E1
F1
H1
J1
A2
B2
E2
F2
G2
H2
J2
A3
B3
E3
F3
G3
H3
J3
K3
A4
B4
E4
F4
G4
H4
J4
+

Resistor Installation

Install the following 13 resistors according to the instructions on page **5** .

RESISTORS

(X) R1, Brown-Black-Orange, 1/4 or 1/2 W.

(X) R2, Brown-Black-Orange, 1/4 or 1/2 W.

(X) R3, Orange-White-Orange, 1/4 or 1/2 W.

(X) R4, Brown-Black-Orange, 1/4 or 1/2 W.

(X) R5, Brown-Green-Orange, 1/4 or 1/2 W.

(X) R6, Red-Red-Brown, 1/4 or 1/2 W.

(X) R7, Orange-Orange-Brown, 1/4 or 1/2 W.

(X) R8, Brown-Green-Orange, 1/4 or 1/2 W.

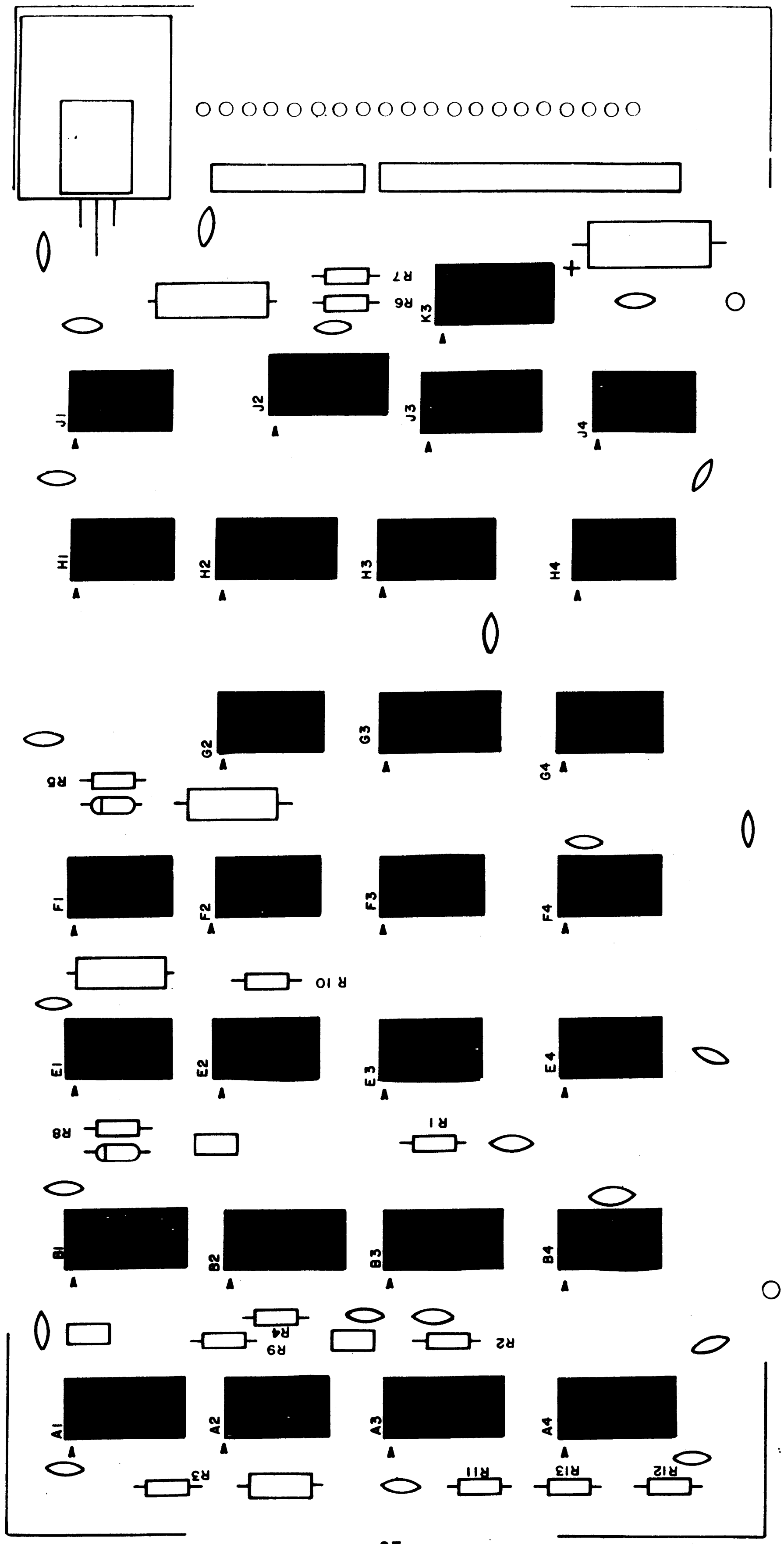
(X) R9, Blue-Gray-Red, 1/4 or 1/2 W.

(X) R10, Brown-Blue-Orange, 1/4 or 1/2 W.

(X) R11, Brown-Black-Red, 1/4 or 1/2 W.

(X) R12, Brown-Black-Red, 1/4 or 1/2 W.

(X) R13, Brown-Black-Red, 1/4 or 1/2 W.



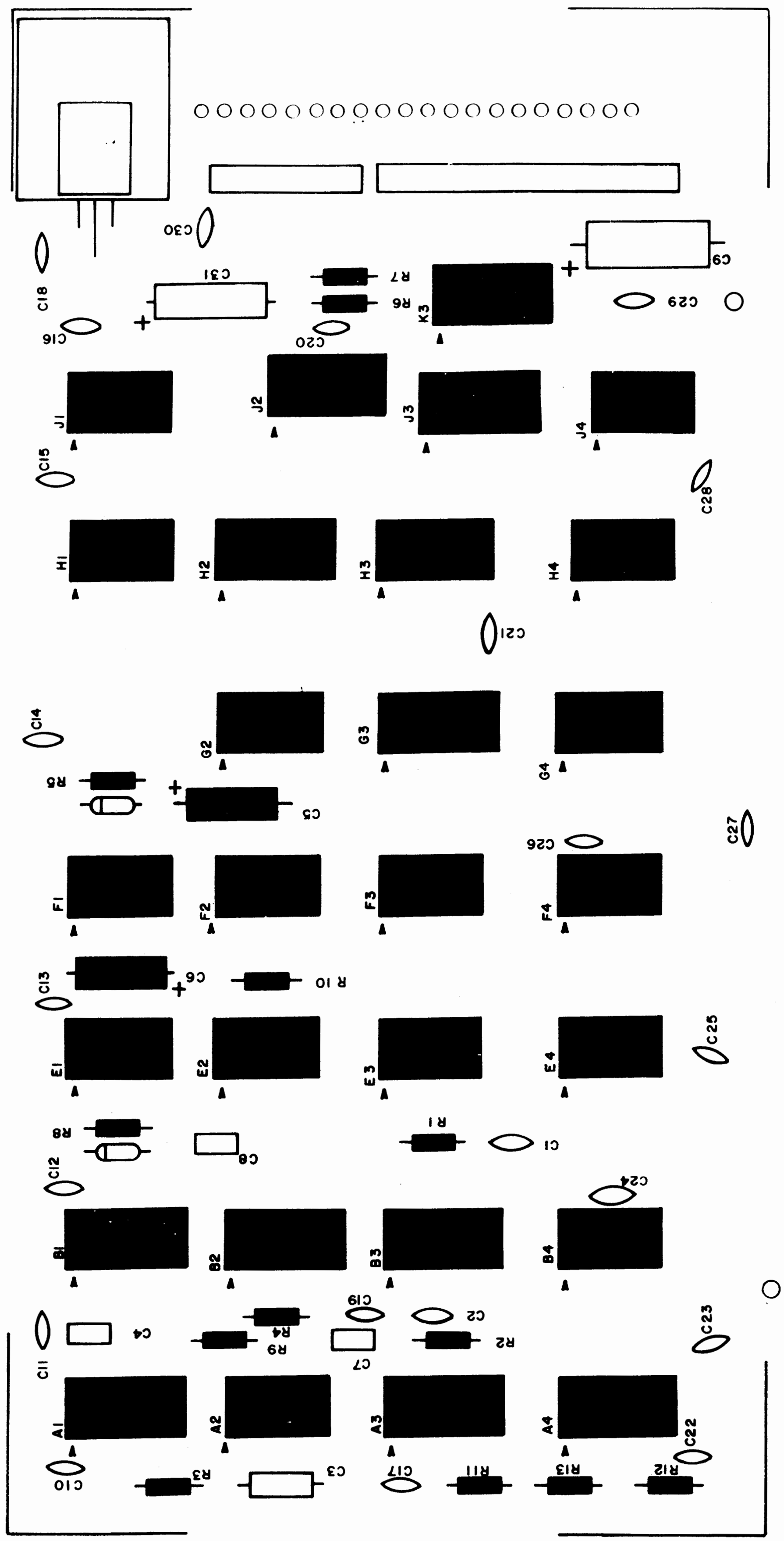
R7
R6
K3
+
J1
J2
J3
J4
H1
H2
H3
H4
G2
G3
G4
R5
F1
F2
F3
F4
R 10
E1
E2
E3
E4
R8
R 1
B1
B2
B3
B4
R4
R9
R2
A1
A2
A3
A4
R3
R11
R13
R12

Capacitor Installation

Install the following 31 capacitors according to the instructions on page **6** . Note that all capacitors are installed in the same manner, <u>except</u> for electrolytic capacitors.

CAPACITORS

- ( ) C1, .001 uf
- ( ) C2, .001 uf
- ( ) C3, 1.0 uf
- ( ) C4, .22 uf
- ( ) C5, electrolytic, 4.7 uf
- ( ) C6, electrolytic, 10 uf
- ( ) C7, .1 uf
- ( ) C8, .1 uf
- ( ) C9, electrolytic, 35 uf
- ( ) C10, .1 uf
- ( ) C11, .1 uf
- ( ) C12, .1 uf
- ( ) C13, .1 uf
- ( ) C14, .1 uf
- ( ) C15, .1 uf
- ( ) C16, .1 uf
- ( ) C17, .1 uf
- ( ) C18, .1 uf
- ( ) C19, .1 uf
- ( ) C20, .1 uf
- ( ) C21, .1 uf
- ( ) C22, .1 uf
- ( ) C23, .1 uf
- ( ) C24, .1 uf
- ( ) C25, .1 uf
- ( ) C26, .1 uf
- ( ) C27, .1 uf
- ( ) C28, .1 uf
- ( ) C29, .1 uf
- ( ) C30, .1 uf
- ( ) C31, electrolytic, 35 uf


C30
C18
C31
R7
R6
K3
C9
C29
C16
C20
J1
J2
J3
J4
C15
C28
H1
H2
H3
H4
C21
C14
G2
G3
G4
R5
C5
C27
C26
F1
F2
F3
F4
C13
C6
R 10
E1
E2
E3
E4
C25
R8
C8
R1
C1
C12
C24
B1
B2
B3
B4
C11
C4
R9
R4
C19
C7
C2
R2
C23
A1
A2
A3
A4
C22
C10
R3
C3
C17
R11
R13
R12

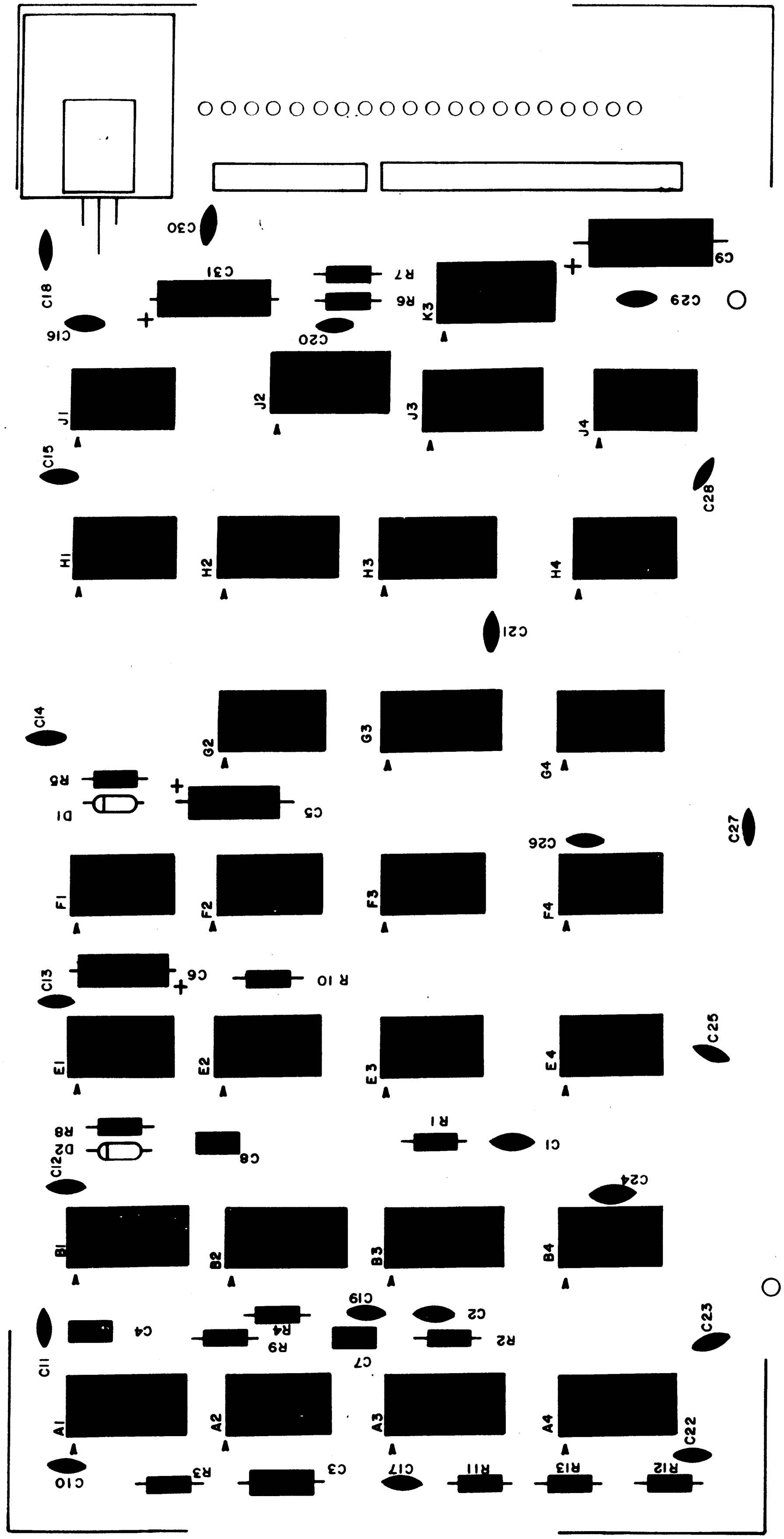
Diode Installation

Install the following two diodes according to the instructions on page 7 .

DIODES

( ) D1, 1N914

( ) D2, 1N914

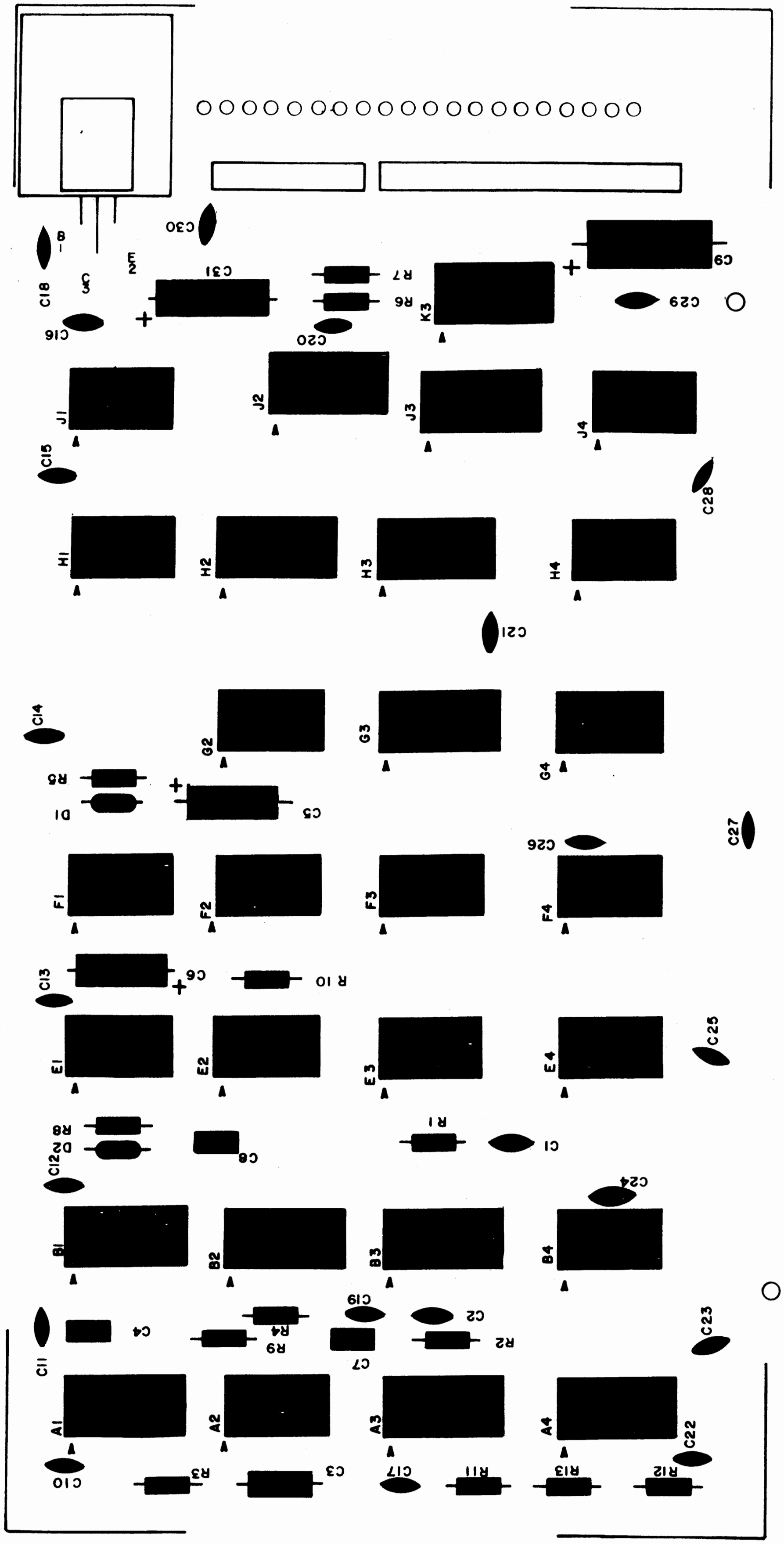

C30
C18
C31
R7
R6
K3
C9
C29
C16
C20
J1
J2
J3
J4
C15
C28
H1
H2
H3
H4
C21
C14
G2
G3
G4
R5
D1
C5
C27
C26
F1
F2
F3
F4
C13
C6
R 10
C25
E1
E2
E3
E4
R8
D2
C8
R1
C1
C12
C24
B1
B2
B3
B4
C11
C4
R4
R9
C19
C7
C2
R2
C23
A1
A2
A3
A4
C22
C10
R3
C3
C17
R11
R13
R12

Voltage Regulator Installation

Install the **voltage** regulator according to the instructions on page **32**.

VOLTAGE REGULATOR

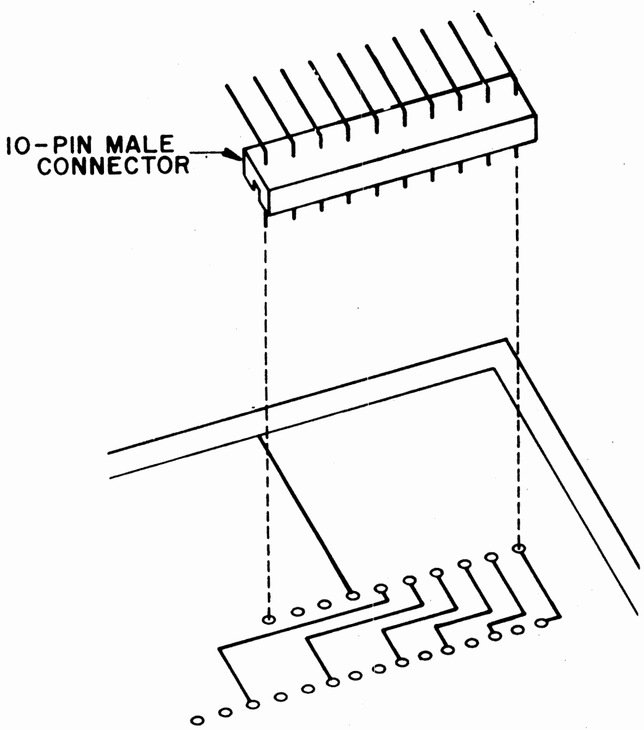
7805



Connector Installation

There are two "boxes" marked on the silk-screen. These are to indicate the positions for a 10-pin and a 20-pin male connector.

The drawing below illustrates the installation of a typical connector of this type.



Refering to the drawing, install the two male connectors onto the silk-screened side of the board. The long 90° bent pins should point towards the right side of the board. The 10-pin connector goes between "CC1" & "CC10"; while the 20-pin connector goes between "CD1" & "CD20".

Two pins should now be cut off. These are the 2nd pin from the top on the 10-pin connector, and the 4th from the top on the 20-pin connector. Cut them off right at the plastic body of the connector. (These pins are both labeled "KEY" on the silk-screen.)

There is a row of 20 pads along the right edge of the board labeled CB1 through CB20.

Remove 10 twisted-pairs of wire from an 8 inch length of ribbon cable. Leave the two wires in each pair twisted together. Strip 1/4 inch of insulation from both ends of all of the wires and tin the exposed portions.

Beginning with the bottom pad on the board, connect one of the twisted-pairs to pads CB1 & CB2. Continue up the row of pads, connecting a twisted-pair to each two pads as you go along.

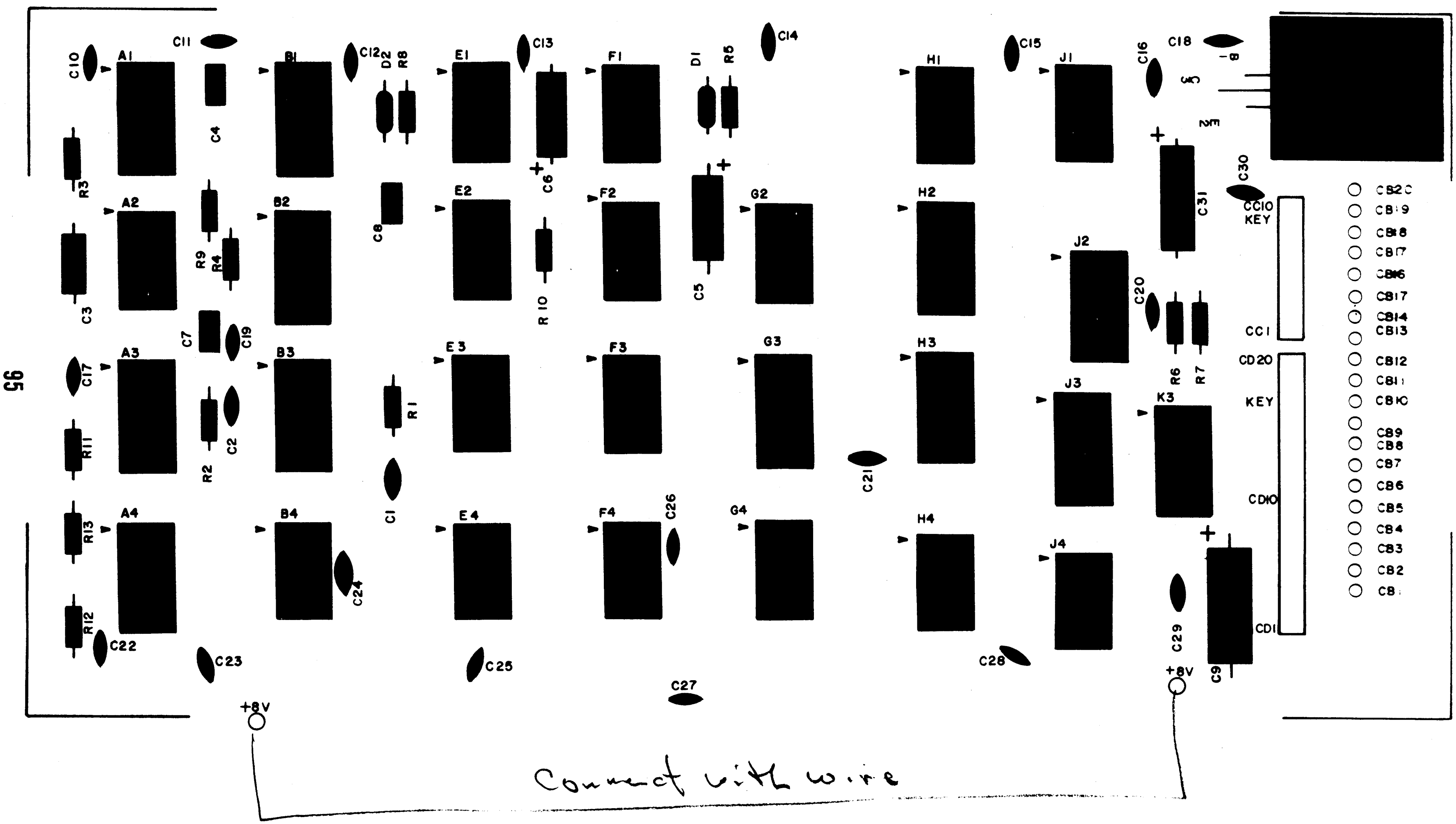
NOTE: The twisted-pairs each have one wire the same color in each of them (usually black or white). Make the connection to pad CB1 with this wire on the 1st pair, and use this wire for the 1st connection on each of the following pairs as you go up the row of 20 pads.

Insert all of the wires from the silk-screened side of the board and solder them of the bottom side. Clip off any excess lead lengths.

Cut the free ends of all 20 wires so that only 1/8 inch of tinned wire is exposed beyond the insulation.

Jumper Installation

Use a length of wire to jumper together the two pads labeled +8V on the bottom edge of the board. Keep it as short as possible and install it on the silk-screened side.


A1
A2
A3
A4
B1
B2
B3
B4
E1
E2
E3
E4
F1
F2
F3
F4
G2
G3
G4
H1
H2
H3
H4
J1
J2
J3
J4
K3
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
C14
C15
C16
C17
C18
C19
C20
C21
C22
C23
C24
C25
C26
C27
C28
C29
C30
C31
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
D1
D2
CC10
KEY
CC1
CD20
KEY
CD10
CD1
+8V
+8V
Connect with wire

A 20-pin female connector will now be attached to the free ends of the 20 wires.

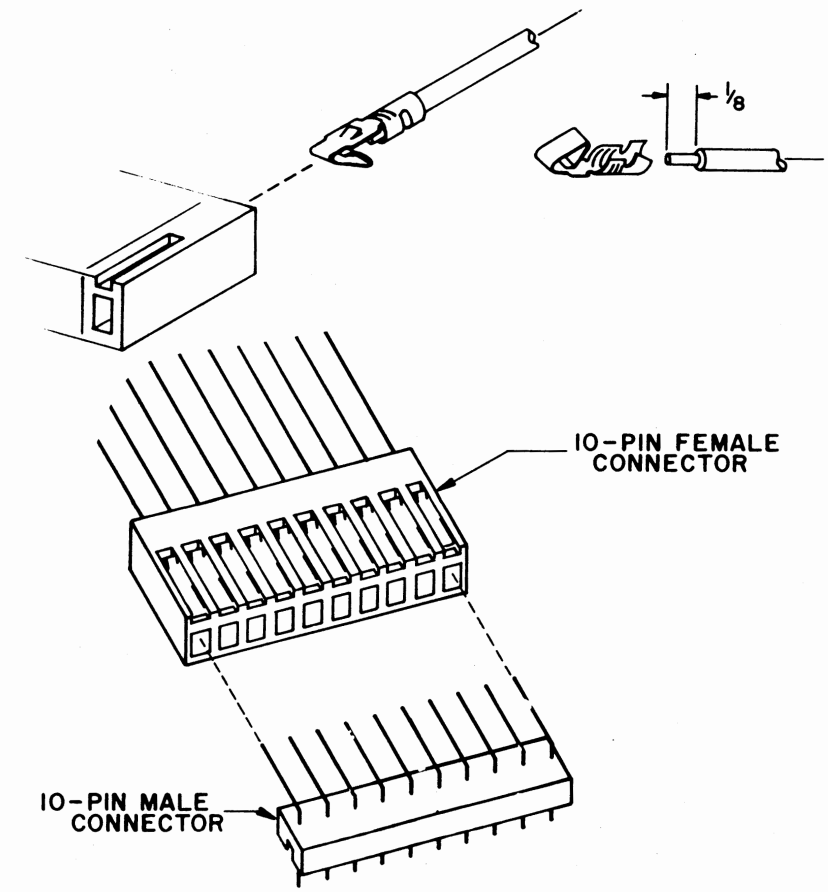
First, connector pins must be attached to the ends of all of the wires. The drawings below illustrate a typical connector of this type, and the method for attaching and inserting the pins.

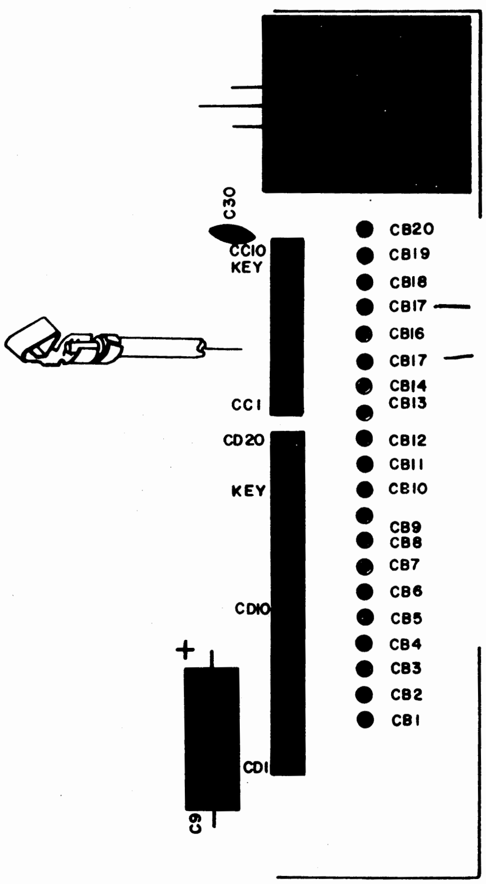
Connect a pin to each of the wires* as shown, and solder them carefully into place. Do not use too much solder or the pins will not fit into the connector properly.

NOTE: Two of the wires, both labeled CB17 on the PC board (see silk-screen), should be attached to a single pin.

Pins 1 & 20 are marked on the plastic body of the female connector. Refering to the silk-screen, insert the pins into the connector so that pad CB1 goes to pin 1, CB2 to pin 2, CB3 to pin 3, etc., being sure not to insert any wires into pin 15 on the connector. A plastic key should be inserted into pin 15 of the female connector, inserting it from the opposite side as the wires.

Place a tie-wrap approximately in the center between the connector and the board to hold the wires together. Place another tie-wrap around the wires and also through the holes in the PC board just to the right of the 20 pads.





IC Installation

Install the following 31 ICs according to the instructions on page 4 .

ICs

Silk Screen Designation	Number	Silk Screen Designation	Number
( ) A1	74123	( ) F2	74L73
( ) A2	74L02	( ) F3	74L73
( ) A3	74L20	( ) F4	74123
( ) A4	74L10	( ) F5	74L30
( ) A5	74L10	( ) G1	74164
( ) B1	93L16	( ) G2	74L00
( ) B2	74L74	( ) G3	74L75
( ) B3	74L73	( ) G4	7493
( ) B4	74L11	( ) G5	74L04
( ) B5	74L04	( ) H1	74L75
( ) E1	74123	( ) H2	8T97
( ) E2	74L00	( ) H3	8T97
( ) E3	74L73	( ) H4	8T97
( ) E4	74L04	( ) H5	8T97
( ) E5	74L00	( ) J3	74L04
( ) F1	74123		

Insert Page

ALTAIR FLOPPY DISK

Disk Controller Assembly Procedure

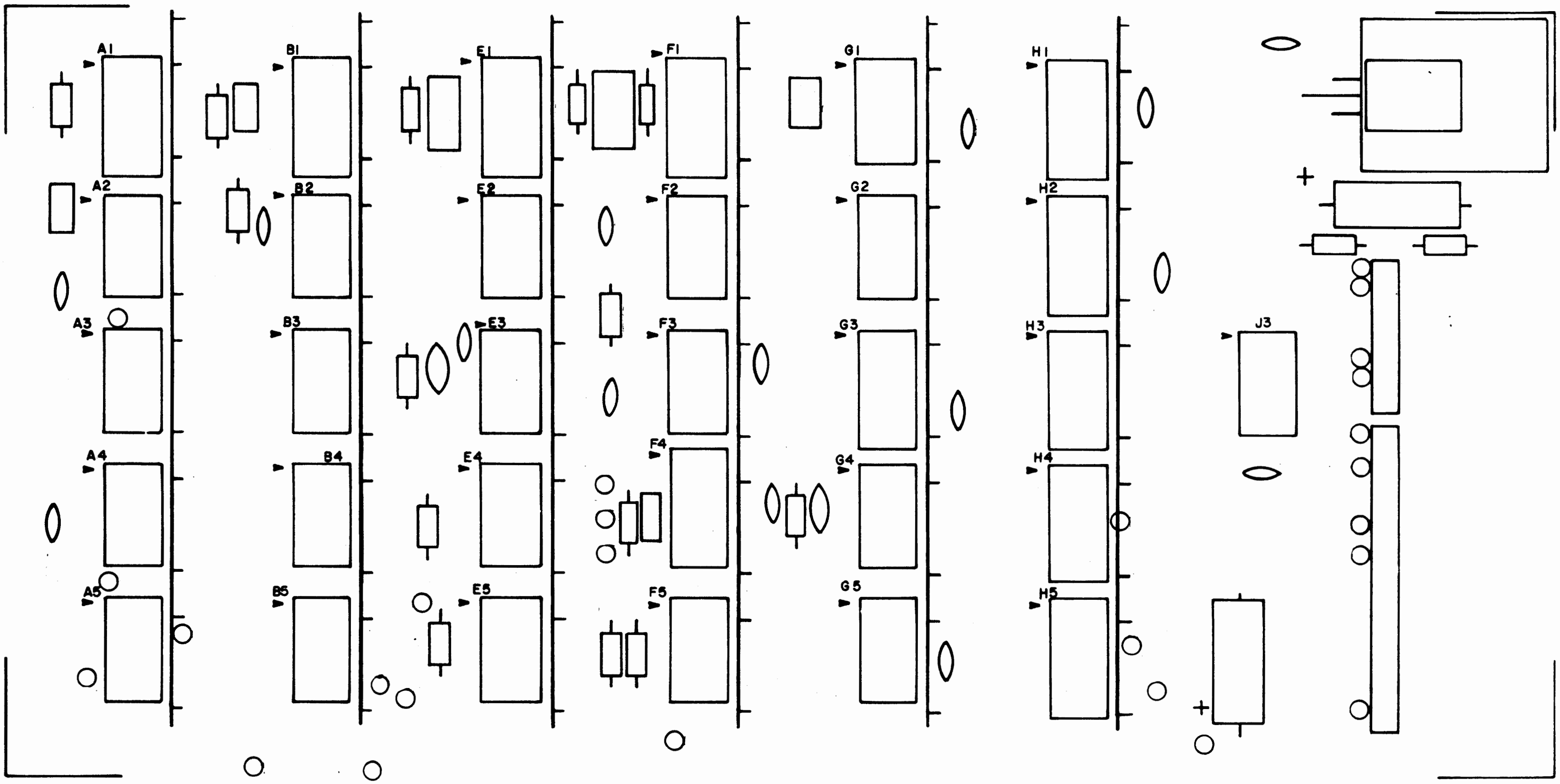
Addendum to Page 98, IC INSTALLATION

Before installing IC "B3" on Disk Controller Board #1, bend pin 7 up so that it does not go into the PC Board.

After all of the ICs have been installed, connect a jumper wire from pin 7 of IC "B3" to the pad labelled "SSC" (pin 9 of IC "B5"). (There should be nothing connected directly to the pad under pin 7 of IC B3.)

Make a note on the schematic for Disk Controller Board #1, sheet 1 of 3, for IC "B3", pin 7. The "J" input of the flip-flop (pin 7) now connects to pin 9 of IC "B5" on sheet 2 of 3 ($\overline{HS}$ - not head status).

MITS, Inc.
August, 1976


A1
A2
A3
A4
A5
B1
B2
B3
B4
B5
E1
E2
E3
E4
E5
F1
F2
F3
F4
F5
G1
G2
G3
G4
G5
H1
H2
H3
H4
H5
J3
+
+

Resistor Installation

Install the following 16 resistors according to the instructions on page 5 .

RESISTORS

( ) R1, Orange-Orange-Brown, 1/4 or 1/2 W.

( ) R2, Red-Red-Brown, 1/4 or 1/2 W.

( ) R3, Brown-Black-Orange, 1/4 or 1/2 W.

( ) R4, Red-Black-Orange, 1/4 or 1/2 W.

( ) R5, Brown-Black-Orange, 1/4 or 1/2 W.

( ) R6, Red-Black-Orange, 1/4 or 1/2 W.

( ) R7, Green-Blue-Red, 1/4 or 1/2 W.

( ) R8, Brown-Black-Orange, 1/4 or 1/2 W.

( ) R9, Orange-Orange-Brown, 1/4 or 1/2 W.

( ) R10, Red-Red-Brown, 1/4 or 1/2 W.

( ) R11, Brown-Black-Orange, 1/4 or 1/2 W.

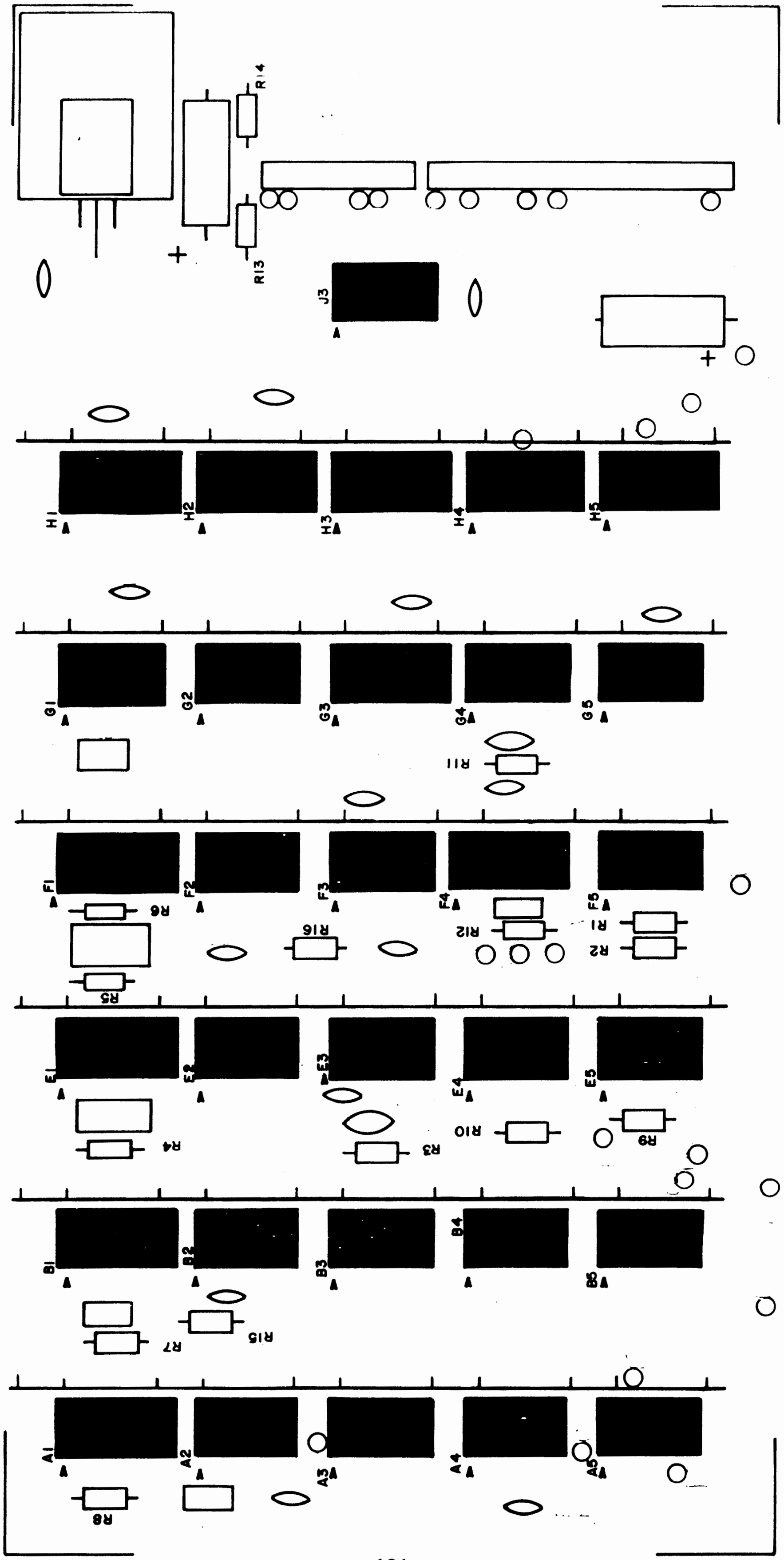
( ) R12, Red-Black-Orange, 1/4 or 1/2 W.

( ) R13, Red-Red-Brown, 1/4 or 1/2 W.

( ) R14, Orange-Orange-Brown, 1/4 or 1/2 W.

( ) R15, Brown-Black-Red, 1/4 or 1/2 W.

( ) R16, Brown-Black-Red, 1/4 or 1/2 W.


R14
R13
J3
H1
H2
H3
H4
H5
G1
G2
G3
G4
G5
R11
F1
F2
F3
F4
F5
R6
R16
R12
R1
R2
R5
E1
E2
E3
E4
E5
R10
R9
R4
R3
B1
B2
B3
B4
B5
R7
R15
A1
A2
A3
A4
A5
R8

Capacitor Installation

Install the following 25 capacitors according to the instructions on page 6 . Note that all capacitors are installed in the same manner, except for electrolytic capacitors.

CAPACITORS

(X) C1, .1 uf

(X) C2, .68 uf

(X) C3, .047 uf

(X) C4, .68 uf

(X) C5, 430 pf

(X) C6, 910 pf

(X) C7, electrolytic, 33 uf

(X) C8, .01 uf

(X) C9, .047 uf

( ) C10, .1 uf

( ) C11, .1 uf

( ) C12, .1 uf

( ) C13, .1 uf

( ) C14, .1 uf

( ) C15, .1 uf

( ) C16, .1 uf

( ) C17, .1 uf

( ) C18, .1 uf

( ) C19, .1 uf

( ) C20, .1 uf

( ) C21, .1 uf

( ) C22, .1 uf

( ) C23, .1 uf

( ) C24, .1 uf

(✓) C25, electrolytic, 35 uf

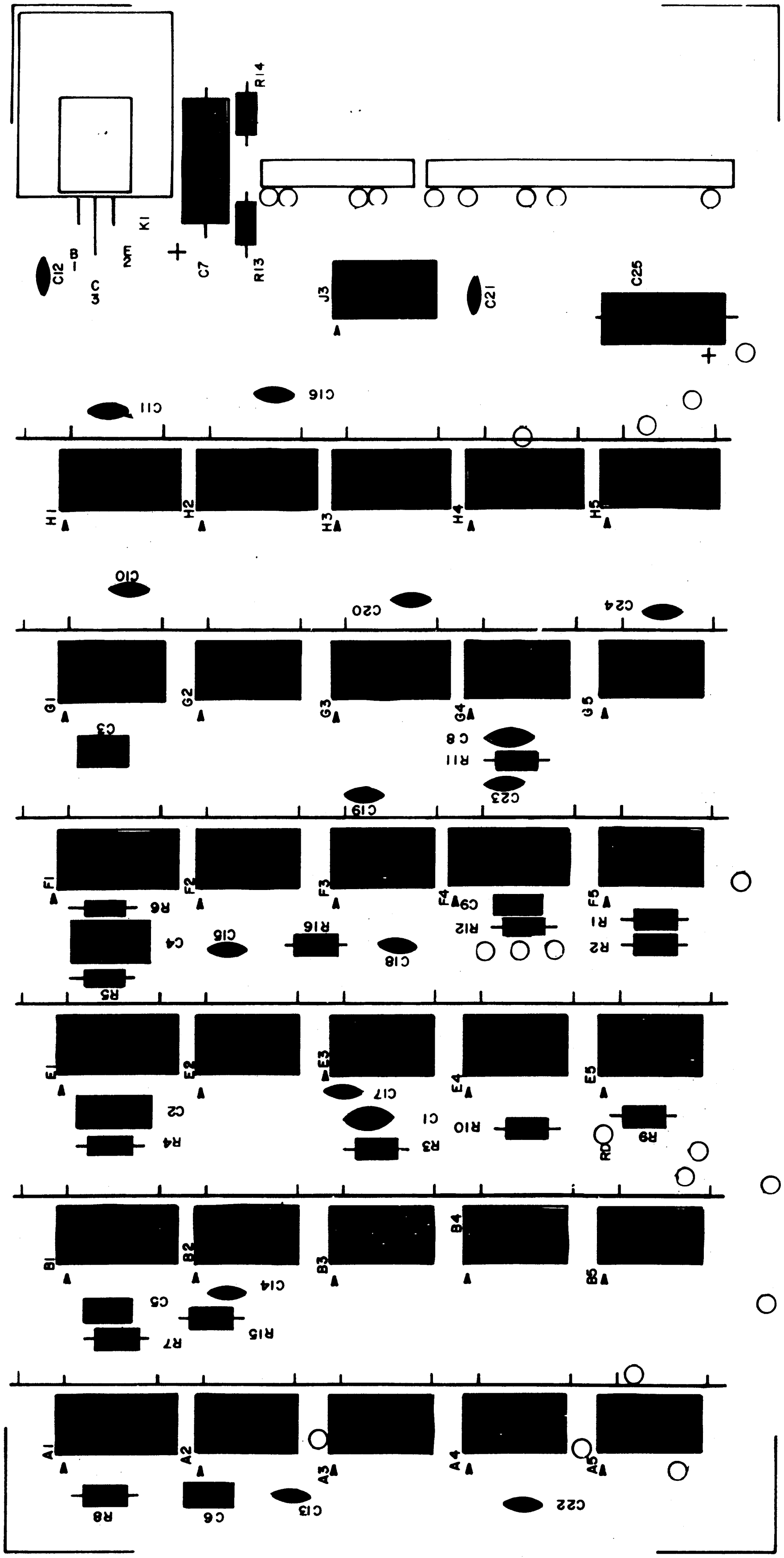
R14
C12
C7
R13
J3
C21
C25
C11
C16
H1
H2
H3
H4
H5
C10
C20
C24
G1
G2
G3
G4
G5
C3
C8
R11
C23
C19
F1
F2
F3
F4
F5
C9
R6
R16
R12
R1
R2
C4
C15
C18
R5
E1
E2
E3
E4
E5
C17
C2
C1
R10
R9
R4
R3
B1
B2
B3
B4
B5
C14
C5
R15
R7
A1
A2
A3
A4
A5
R8
C6
C13
C22

Voltage Regulator Installation

Install the voltage regulator according to the instructions on page **32**.

VOLTAGE REGULATOR

( ) K1, 7805



K1
C12
C7
R14
R13
J3
C21
C25
C11
C16
H1
H2
H3
H4
H5
C10
C20
C24
G1
G2
G3
G4
G5
C3
C8
R11
C23
C19
F1
F2
F3
F4
F5
C9
R6
C4
C15
R16
C18
R12
R1
R2
R5
E1
E2
E3
E4
E5
C17
C2
C1
R10
R9
R4
R3
B1
B2
B3
B4
B5
C14
C5
R15
R7
A1
A2
A3
A4
A5
R8
C6
C13
C22

Jumper Installation

There are 13 jumper wires to be installed on board #1.

Install these jumper wires by inserting them on the silk-screened side of the board and soldering them on the back side. Clip off any excess lead length.

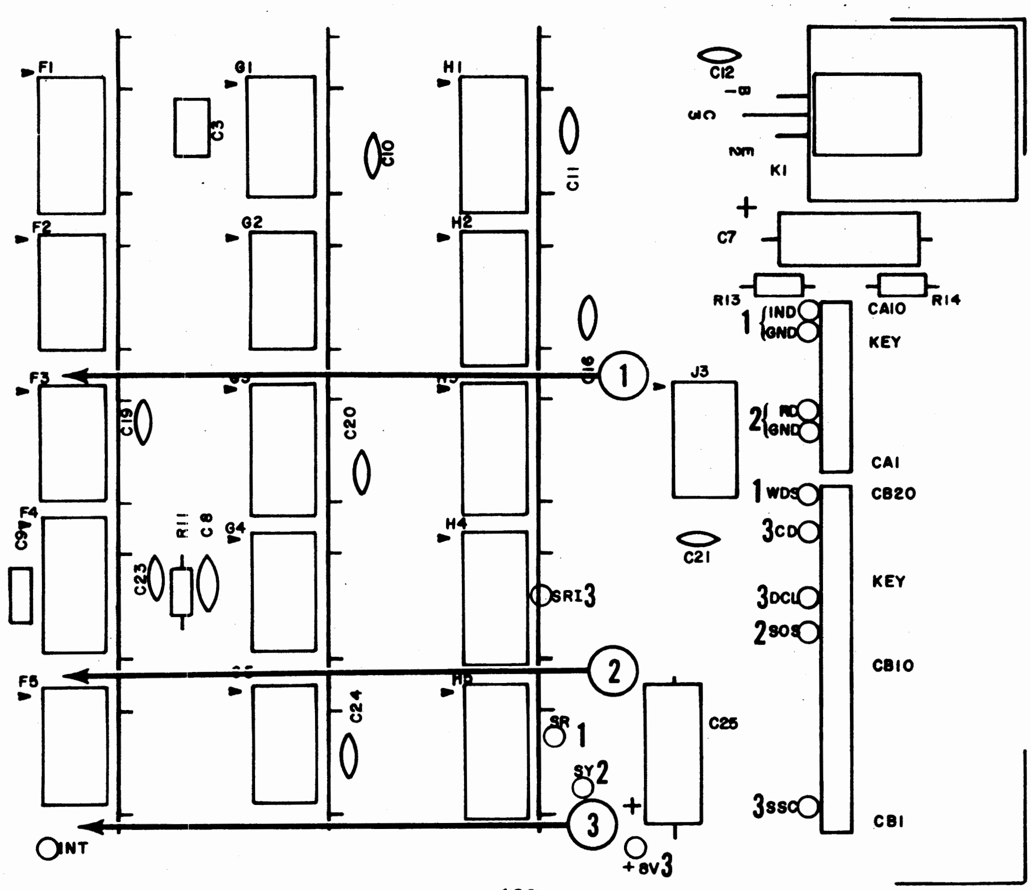
The drawing below shows the proper way to route the wires across the board. Pay close attention to this as it is very important. Pads labeled 1 below route through arrow 1, 2 through 2, and 3 through 3.

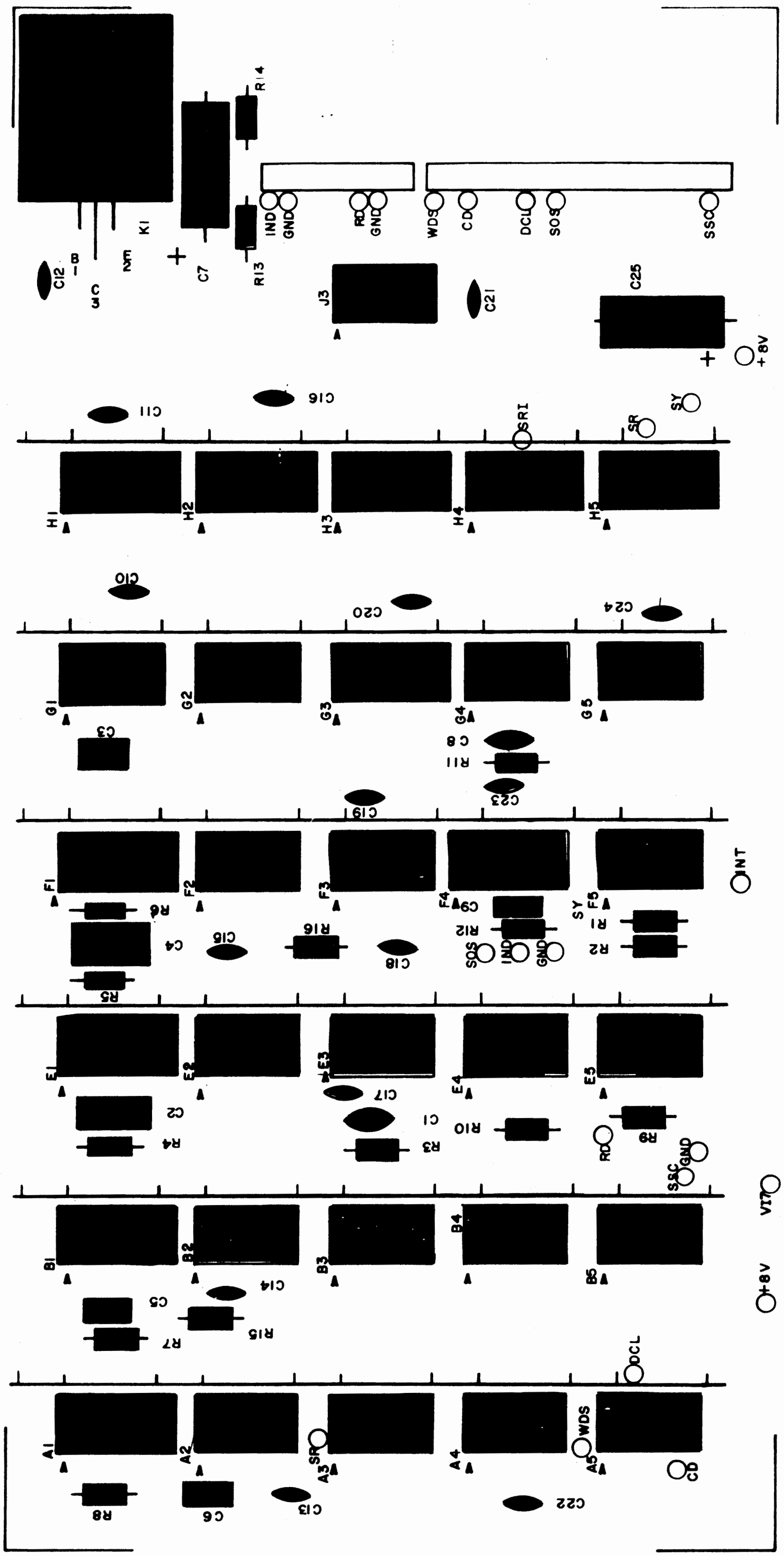
Cut the wires to the necessary length, and install them through the paths as shown. Use ribbon cable wires for the two twisted pair connections. The "GND" pad for the twisted pairs is the one closest to the other connection stated.

Connect the following jumpers:

IND to IND
GND to GND
RD to RD
GND to GND
WDS to WDS
CD to CD
DCL to DCL
SOS to SOS
SSC to SSC
+8V to +8V
SY to SY
SR to SR
SRI to INT*

*or to VI7 (see Theory of Operation)

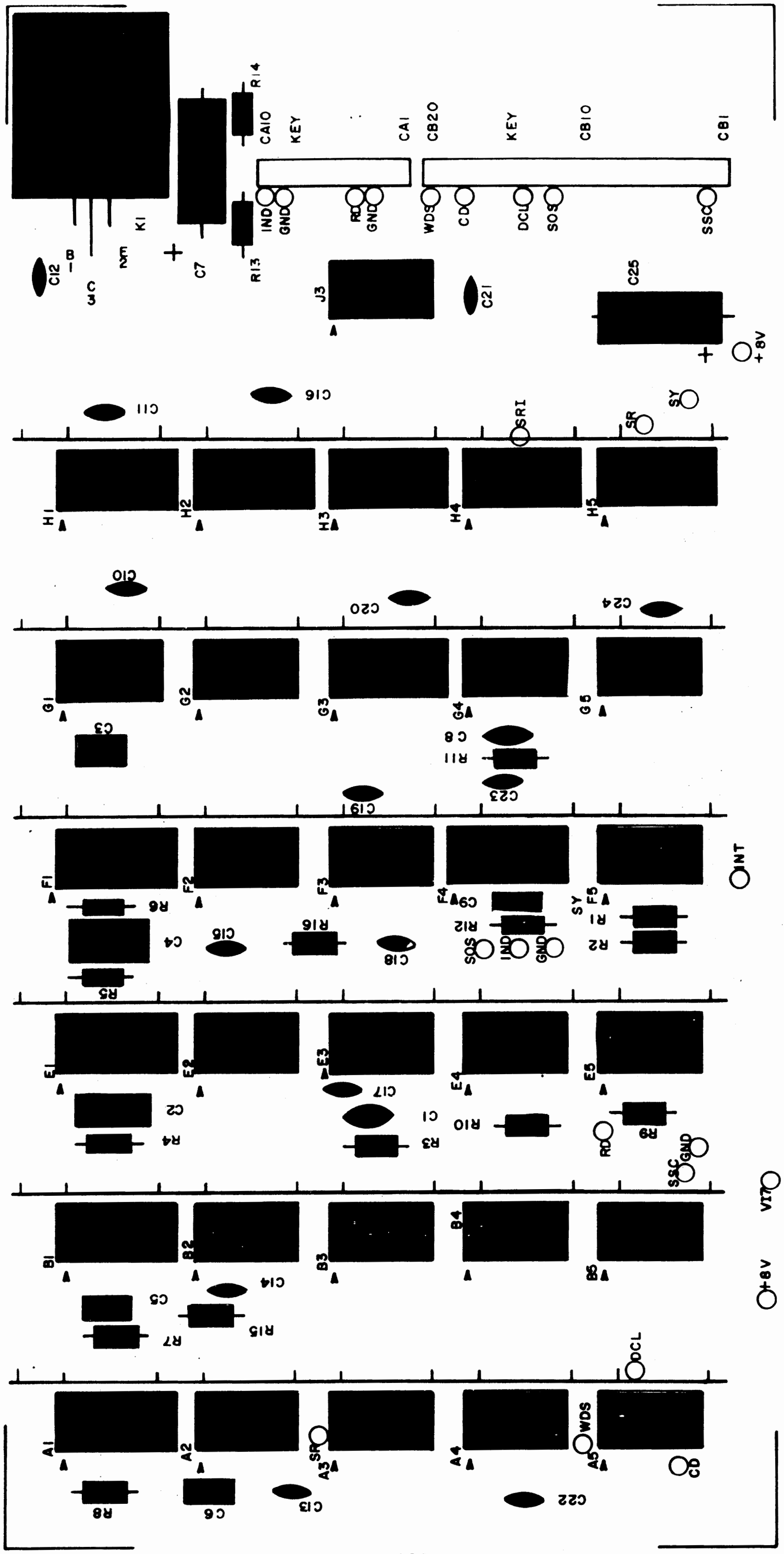



K1
C7
R13
R14
C12
J3
C21
C25
+8V
IND
GND
RD
WDS
CD
DCL
SOS
SSC
C11
C16
SRI
SR
SY
H1
H2
H3
H4
H5
C10
C20
C24
G1
G2
G3
G4
G5
C3
C8
R11
C19
C23
F1
F2
F3
F4
F5
INT
R6
C4
C15
R16
C18
C9
R12
R1
R2
R5
E1
E2
E3
E4
E5
C2
C17
C1
R10
R9
R4
R3
VIT
B1
B2
B3
B4
B5
C5
C14
R15
R7
A1
A2
A3
A4
A5
R8
C6
C13
C22

Connector Installation

Install a 10-pin and a 20-pin female connector onto the board in the same manner as described on page **94** for board #2.

NOTE: The only exception to the above statement is that pin 6 is to be cut off instead of pin 4 on the 20-pin connector.



CA10
KEY
CA1
CB20
KEY
CB10
CB1
IND
GND
RD
GND
WDS
CD
DCL
SOS
SSC
R14
R13
C7
K1
C12
C21
C25
J3
+8V
C11
C16
SRI
SR
SY
H1
H2
H3
H4
H5
C10
C20
C24
G1
G2
G3
G4
G5
C3
C8
R11
C23
C19
F1
F2
F3
F4
F5
INT
R6
C9
R12
R1
R2
C4
C15
R16
C18
SOS
IND
GND
R5
E1
E2
E3
E4
E5
C17
C2
C1
R10
R9
RD
GND
SSC
R4
R3
B1
B2
B3
B4
B5
+8V
C14
C5
R15
R7
DCL
A1
A2
A3
A4
A5
SR
WDS
CD
R8
C6
C13
C22

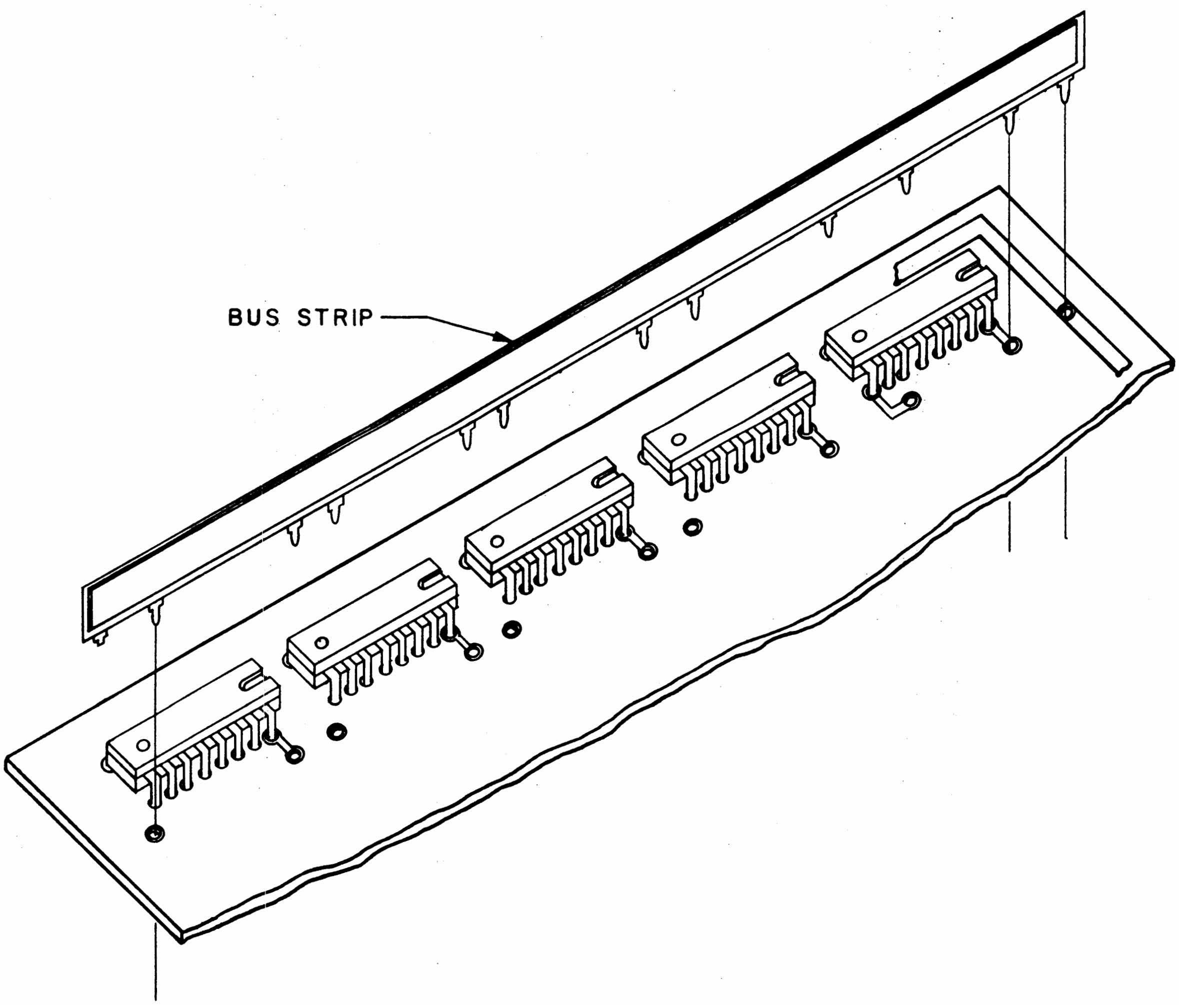
Bus Strip Installation

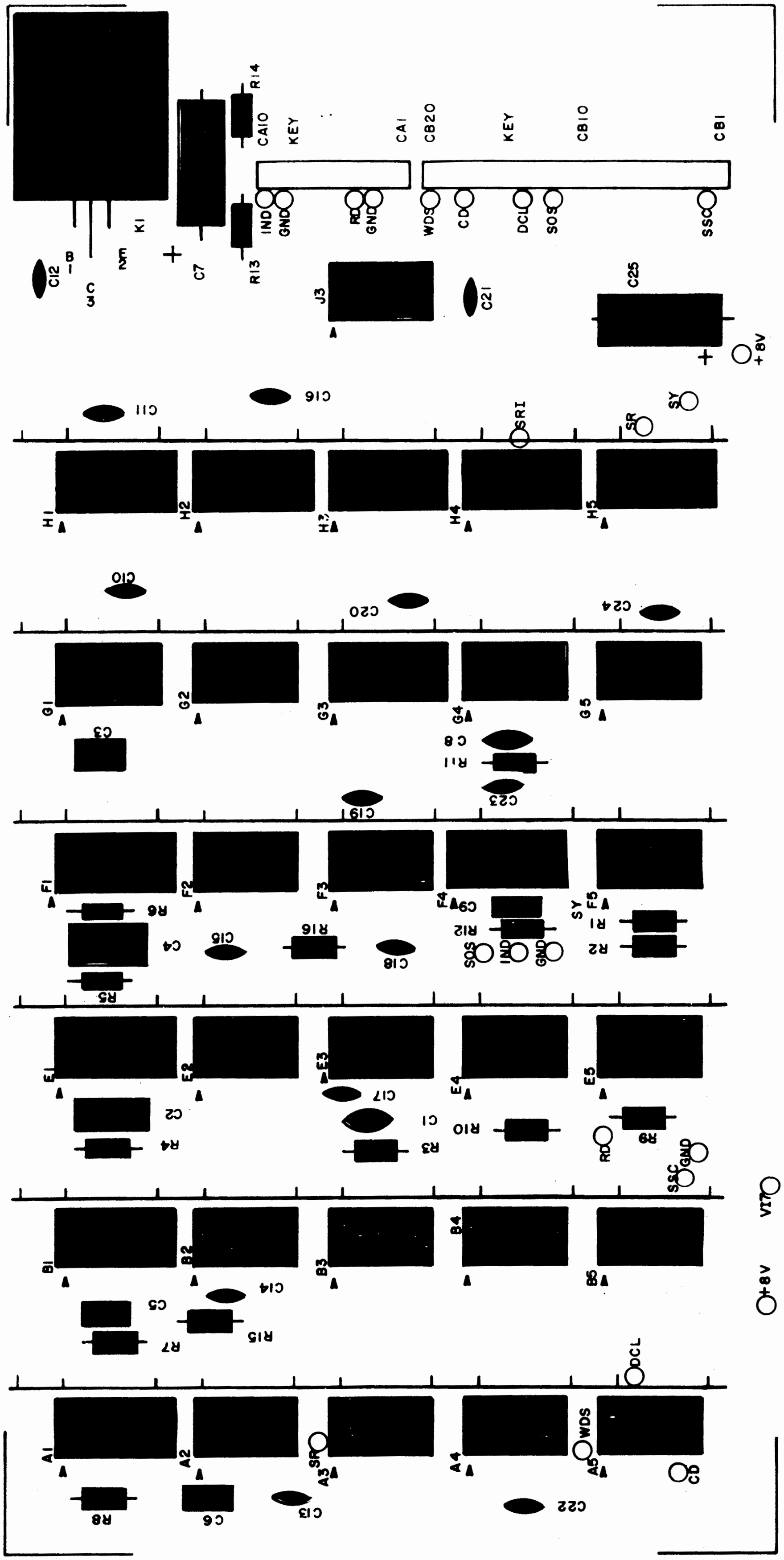
The drawing below illustrates the method for installing the 6 bus strips onto the board.

Note that the last pin (on the bottom side of the board) is to be cut off before installing the strips.

Be careful when installing these strips, that you do not push the strips down tight enough to damage the jumper wires or to short any of the PC lands.

Insert them as shown below and solder them on the non-silk-screened side of the board.





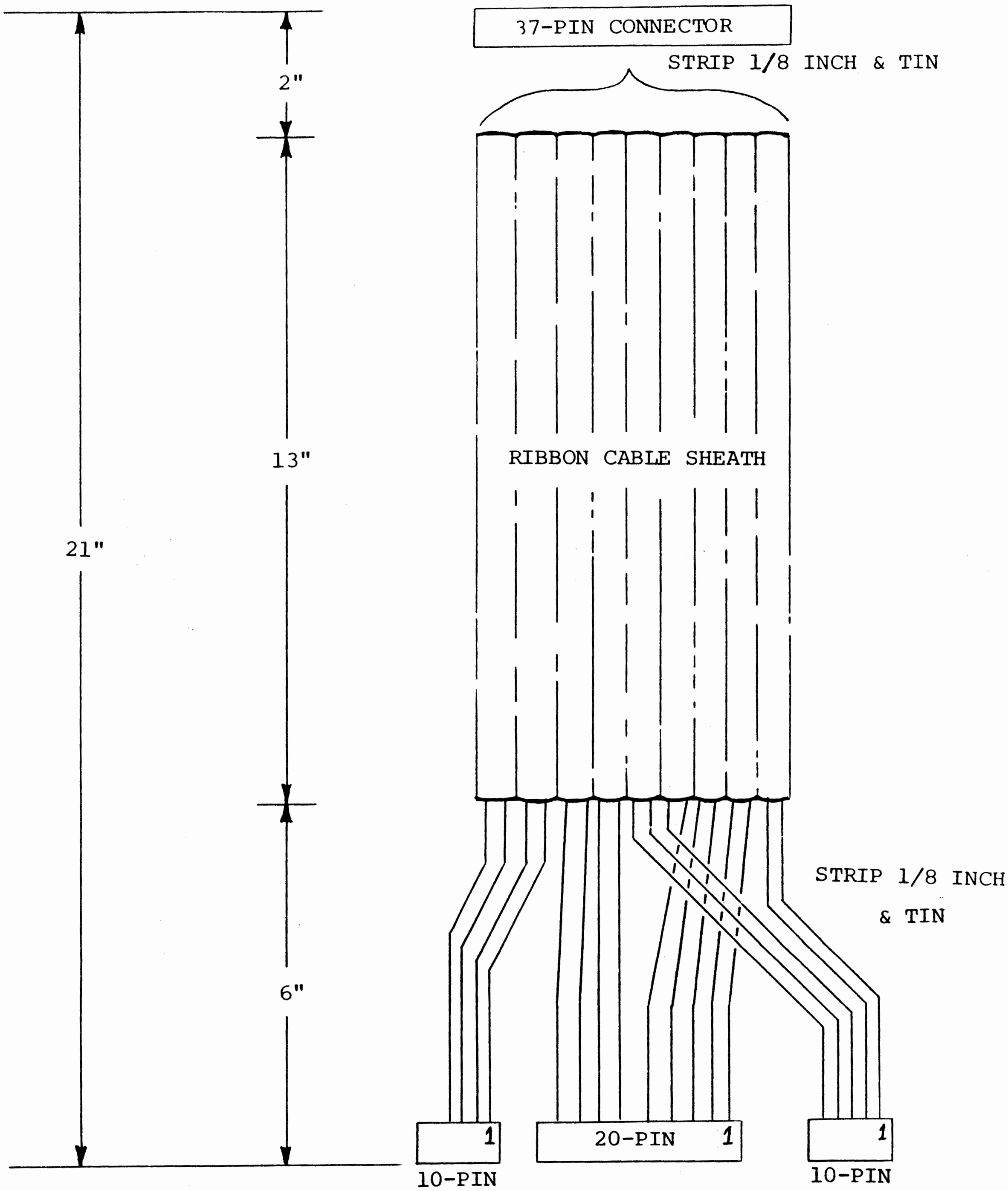
Controller Cable Assembly

Refering to the drawing on the following page, and to the previous instructions beginning on page 44, cut a 21 inch length of ribbon cable and prepare it as shown in the drawing.

The 37-pin connector shown at the top of the drawing is one of the FEMALE connectors included with your kit. The 10 & 20 pin connectors shown at the bottom of the drawing are of the same type as that on page 97 (female connectors), and should be assembled in the same manner.

Use the drawing on the following page, and the chart and drawing following after that, to construct this cable in the same relative manner as the previous ribbon cables.

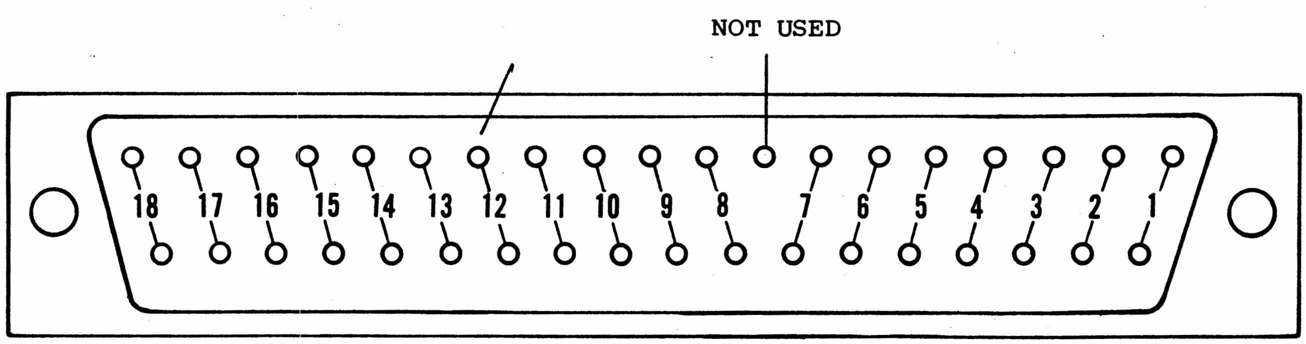
DISK CONTROLLER CABLE



The drawing below illustrates the pin positions where each of the 18 twisted-pairs should be attached to the 37-pin connector. Be sure to use a female connector. This portion of the assembly is essentially identical with that shown on page 51.

Use the orientation for this process shown on page 113. It would be adviseable to connect the varied colored wires from each pair to pins 1 through 19, and the same colored wire from each pair to pins 20 through 37.

37-PIN FEMALE CONNECTOR

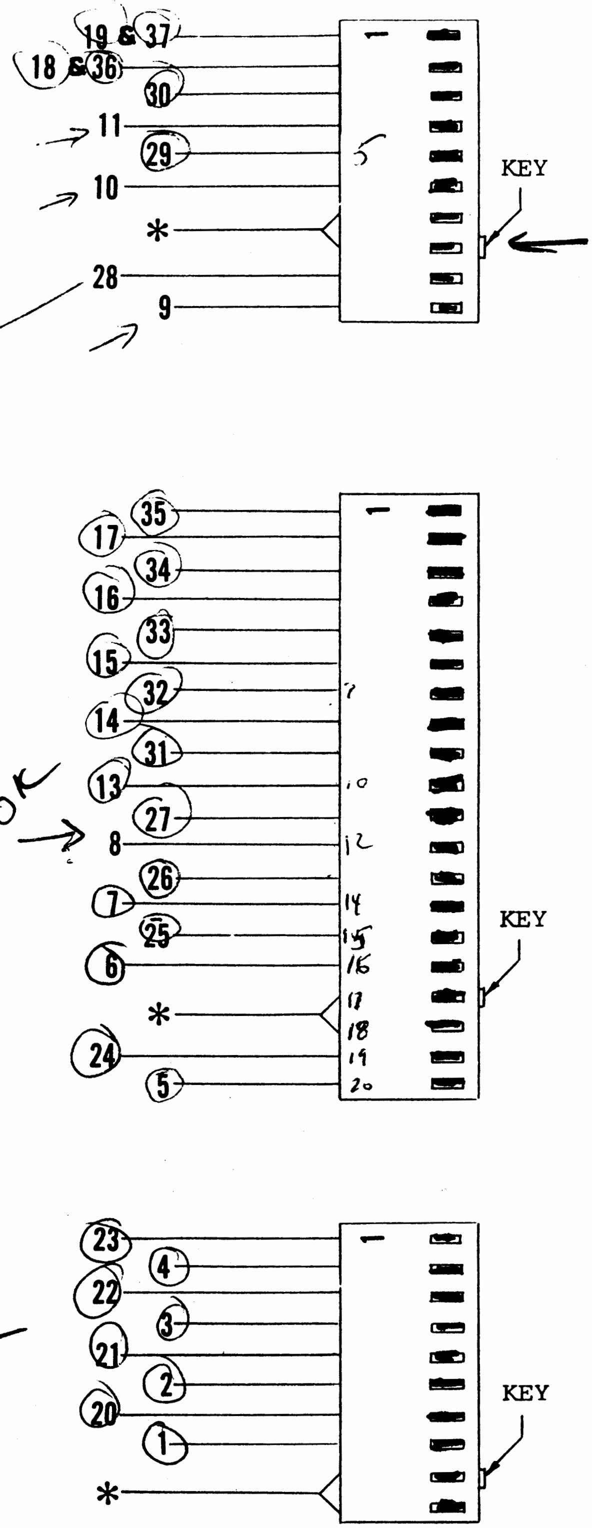


The drawing on the right illustrates the same three female connectors as shown on the bottom of the drawing on page 113. The orientation in the drawing on the right is the same as that on page 113, only rotated 90° counterclockwise.

The first step in this assembly process is to attach connector pins to the ends of each of the wires. Do this in the same manner as described on page 97. Note that two of the twisted-pairs have both of their wires attached to a single connector pin.

Once this is completed, the pins can be inserted into the female connectors. The numbers in the drawing on the right refer to the 37-pin connector pin numbers. Use the same procedure as with the previous ribbon cables and insert the pins into the connectors, correlating the 37-pin connector pin numbers on the right with the with the proper wires and positions on the 3 female connectors.

Insert the the plastic keys in the positions shown. Be sure to insert them from the opposite side that the wires are inserted from.



* NO WIRE CONNECTION

CONTROLLER/DRIVE INTERCONNECT CABLE ASSEMBLY

There is one more cable to be assembled for the disk system. This cable will be used to connect the Disk Drive unit with the ALTAIR containing the controller.

1) The first step is to cut a 6 foot length of ribbon cable and remove 2 inches of the cable sheath from each end.

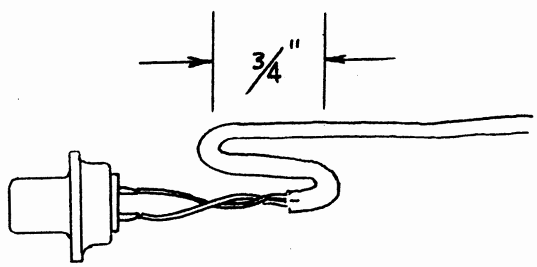
2) There are two grey plastic connector covers included in your kit. Slip one of these over each end of the cable, with the small holes towards the center of the cable and the larger holes towards the free ends. Push the covers down at least a foot so that they will not interfere with the rest of this procedure.

3) Strip 1/8 inch of insulation from both ends of each of the cable wires and tin the exposed portion.

4) Prepare the two remaining 37-pin connectors (one male & one female) in the same manner as the previous 37-pin connectors.

5) For this cable the connections will simply run pin-to-pin. That is, connect pin 1 of the male connector to pin 1 of the female connector. BE SURE NOT TO CONNECT ANY WIRES TO PIN 12 OF EITHER CONNECTOR.

6) Once all 36 wires have been connected on both ends, push the ends of the cable into a fold as shown on the right, and secure it with a double wrap of masking tape. Keep the fold as close as possible to the connector itself.



7) Push the connector covers into place over the two connectors. Do not use any of the hardware supplied with the covers by the factory. Simply mount the 37-pin connectors to the covers using standard 4-40 X 5/16 " screws.

DISK/COMPUTER INTERFACE

Refer to the preliminary documentation release included with this manual for a description of how to hook-up and operate this system.

The above mentioned documentation includes an abreviated version of both the theory and the operation of the ALTAIR FLOPPY DISK SYSTEM.

An updated, complete version of this documention will be sent at a later date, as described in the front of this manual.

DISK CONTROLLER CHECK OUT
WITH DISK DRIVE

A) Preliminary Test

This tests the primary functions of the Disk Drive and Disk Controller.

Enter the following program and then single step through (with Controller and Drive connected).

Address	Instruction		Note
000,000	076	MVI A	NOTE 1
1	000	Disk Drive Addr (Ø)	NOTE 1
2	323	Output	NOTE 1
3	010	Disk Enable Channel	NOTE 1
4	076	MVI A	NOTE 2
5	004	Head Load (Bit D2=1)	NOTE 2
6	323	Output	NOTE 2
7	011	Disk Control Channel	NOTE 2
10	333	Input	NOTE 3
11	011	Sector Position Channel	NOTE 3
12	333	Input	NOTE 4
13	010	Disk Status Channel	NOTE 4

Note 1

Disk Drive should be enabled at the end of these 4 instructions.

Note 2

Disk Drive Head should be loaded at the end of these 4 instructions.

Note 3

After single stepping these two instructions, the ALTAIR data lights should indicate as follows:

DØ on all the time
D1 on all the time (flashing very fast)
D2 on all the time (flashing very fast)
D3 flashing very fast
D4 flashing slower
D5 flashing slowest
D6 on-not used
D7 on-not used

The flashing lights indicate the index/sector circuits are functioning properly.

Note 4

The last two instructions, when single stepped through, indicate the status or the disk on the data lights as follows:

D∅ - (ENWD) - On
D1 - (MH) - Off
D2 - (HS) - Off
D3 - Not used - Off
D4 - Not used - off
D5 - (INTE) - On if "INTE" on front panel off
D6 - (TRACK ∅) - Off if disk head on track ∅
D7 - (NRDA) - Flickering, half on - indicates that read circuit is OK.

B) Testing Individual Functions

To test individual disk functions, an output of the correct data pattern must be done on Channel 011.

For example, to step the head in, use this program. Note--The disk must be enabled before doing any disk functions.

Address	Instruction	
000,000	076	MVI A
1	000	Disk Drive Addr.
2	323	Output
3	010	Disk Enable Chan.
4	333	Input
5	377	From Sense SW
6	323	Output
7	011	Disk Control Channel

Set Sense Switch 8 up, others down when single stepping this program. Change switch pattern to control other functions.

SERVICE

Should you have a problem with your unit, it can be returned to MITS for repair. If it is still under warranty any defective part will be replaced free of charge. The purchaser is responsible for all postage. In no case should a unit be shipped back without the outer case fully assembled.

If you need to return the unit to us for any reason, remove the top cover of the drive unit and install the wood block over the door mechanism as it was shipped to you. Secure cover and pack the unit in a sturdy cardboard container and surround it on all sides with a thick layer of packing material. You can use shredded newspaper, foamed plastic or excelsior. The packed carton should be neatly sealed with gummed tape and tied with a stout cord. Be sure to tape a letter containing your name and address, a description of the malfunction, and the original invoice (if the unit is still under warranty) to the outside of the box.

Mail the carton by parcel post or UPS--for extra fast service, ship by air parcel post. Be sure to insure the package.

SHIP TO: MITS, Inc.
2450 Alamo SE
Albuquerque, NM 87106

All warranties are void if any changes have been made to the basic design of the machine or if the internal workings have been tampered with in any way.

ALTAIR DISK TEST PROGRAMS

Reprinted from Computer Notes, April, 1976

Listed below are some Altair Disk Test programs that will check out all the normal functions of the Disk Drive. These check-out procedures will also be included in the Altair Disk Theory of Operation manual.

A. Disk Read/Write Test Program

This program writes data on disk on sector Ø of the track it is positioned on, then reads the data back, stores it in memory, then outputs it to an I/O device. It is used for testing all read/write functions.

WRITE: The number of write data bytes is set by the position of the sense switches (maximum of $22Ø_8$). Write data consists of:

1st byte = 377_8 (D7 = 1 - sync bit)
2nd byte = data on sense switch
3rd byte = 2nd - 1
4th byte = 2nd - 2
.
.
.
"n"th byte = 001
last byte = 000

If sense switch is set to ØØØ, program will stop.

READ: The read data is stored in memory, starting at address $001,236_8$ and consists of the data written by the write program

OUTPUT: After the read program, the data is outputted to a terminal (Teletype, CRT, etc.). The output program is set to output on channel 1. To obtain a useful output pattern, change the sense switches until a desirable pattern is printed. The characters printed will consist of all printable ASCII characters in reversed order (as in 987654321 and zyxwvu . . .). This pattern repeats itself and is easily observed for errors.

B. Stepping Program

This program steps the disk head out 77 times to track Ø and then in 77 times to track 76, continuously repeating with the computer in the run mode.

This program is useful for testing the disk enable, MH status, track Ø status, and stepping functions of the disk.

While stepping with this program, the head is unloaded, so it may be run continuously without wear on the read/write head surface. A squeaking sound caused by the head load mechanism is normal in this test.

To loop with the read/write program, see next section.

For stepping program, disk drive address of ØØØ is used. To change disk drive tested, the address is contained in location (001,001).

Looping With Stepping Program

To check the read/write and step functions simultaneously, the two programs may be run together by changing:

1) Data in locations (000,154) and (000,155) to 037, 001 as indicated.

2) Data in location (001,034) to 303 as indicated.

Start the program at (001,000), the start of the stepping program.

The disk head will step out to track Ø.

The head will then load and a write/read will occur. The head will then unload and output will take place. After output, the head will step in once, starting the write/read sequence again. After this repeats 76 times, the head is stepped out to track Ø, and it begins again. **

NOTE: **

1) For read/write program, disk drive address of ØØØ is used. To change disk drive tested, the address is contained in location (000,001) and (000,150).

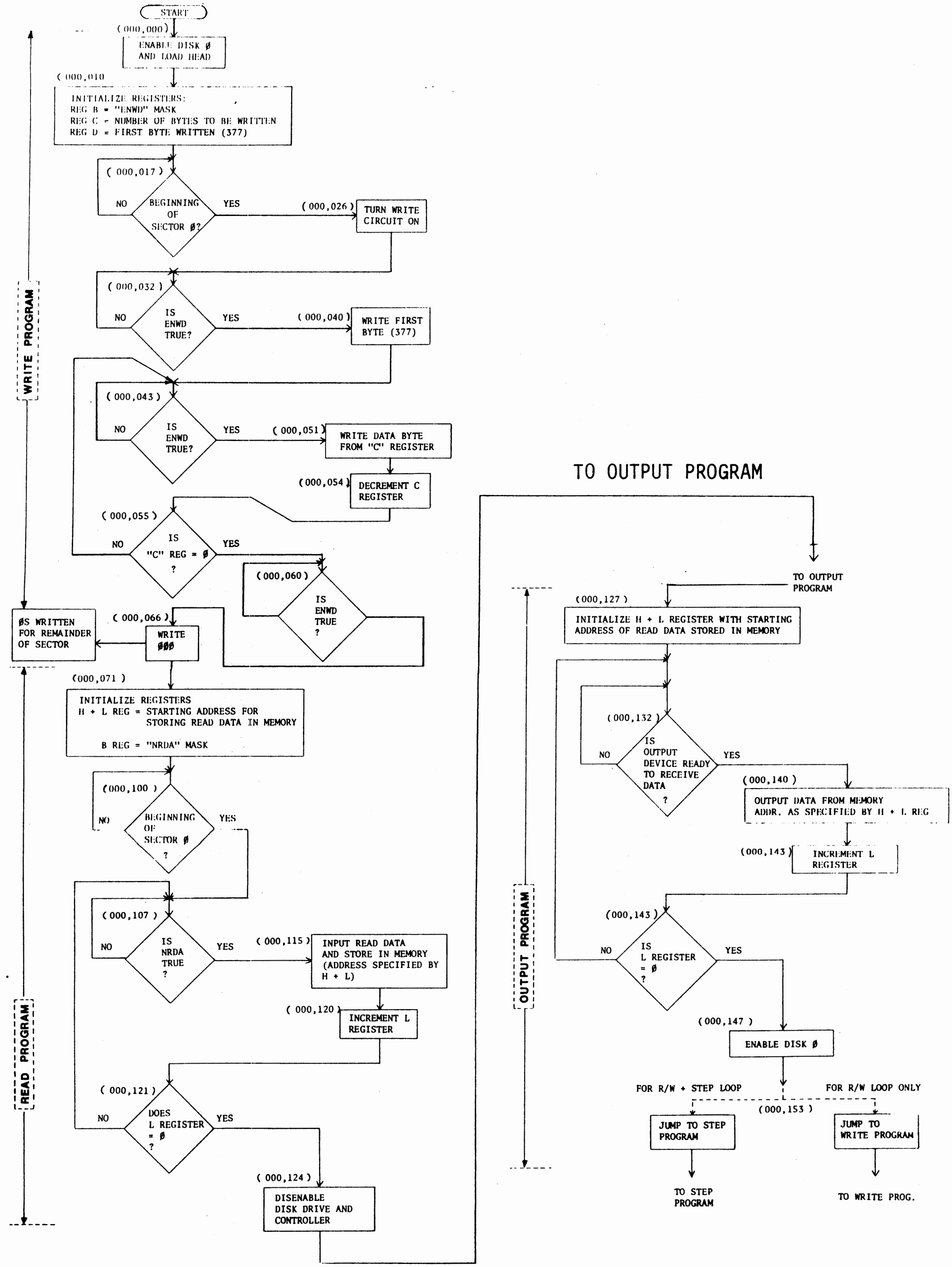
2) Output device addresses are in locations (000,133) (status) and (000,141) (data).

READ/WRITE/OUTPUT PROGRAM

TAG	MNEMONIC	ADDRESS	OCTAL CODE	EXPLANATION
	MVI(A)	000,000	076	
		1	000	Disk drive address
	OUT	2	323	
		3	010	Disk controller enable channel
LDHD	MVI(A)	4	076	
		5	004	Load head bit
	OUT	6	323	
		7	011	Disk function control channel
WRTLP	IN	10	333	Input # of bytes to be written
		11	377	Sense switch
	MOV(C)←(A)	12	117	Store in "C" reg.
	MVI(D)	13	026	Store in "D" reg.
		14	377	First write byte
	MVI(B)	15	006	Store in "B" reg.
		16	001	"ENWD" status mask
WSECT	IN	17	333	Write sector test
		20	011	Sector position channel
	CPI	21	376	
		22	300	∅ sector
	JNZ	23	302	Jump if not start of ∅ sect.
		24	017	to "WSECT"
		25	000	
	MVI(A)	26	076	
		27	200	Write enable bit
		30	323	
		31	011	Disk function control channel
FBYT		32	333	First byte test
		33	010	Disk status channel
	ANA(A)/(B)	34	240	Test for "ENWD" status
	JNZ	35	302	Jump if "ENWD" false (=1)
		36	032	to "FBYT"
		37	000	
	MOV(A)(D)	40	172	Move 377 into accum.
	OUT	41	323	Output first byte
		42	012	Disk data channel
INDAT	IN	43	333	Start of write data sequence
		44	010	Disk status channel
	ANA	45	240	Test for "ENWD" status
	JNZ	46	302	Jump if "ENWD" false (=1)
		47	043	to "WDAT"
		50	000	
	MOV(A)←(C)	51	171	Move "DATA" byte to accum.
	OUT	52	323	
		53	012	Disk data channel
	DCR(C)	54	015	Decrement "DATA" byte
	JNZ	55	302	Jump if data byte = ∅.
		56	043	to "WDAT", write another byte
		57	000	
WZT	IN	60	333	Start of zero byte
		61	010	Output sequence
	ANA(A)←(B)	62	240	Test "ENWD" (last byte written)
	JNZ	63	302	Jump if "ENWD" false
		64	060	To WZT
		65	000	
	XRA(A)(A)	66	257	Zeros accumulator
	OUT	67	323	Output zero byte
		70	012	Disk data channel (end of write, start of read)
	LXI	71	041	Load H+L reg. with:
		72	236	Starting addr. to store read data
		73	001	

TAG	MNEMONIC	ADDRESS	OCTAL CODE	EXPLANATION
	MVI(B)	74	006	Store in "B" reg.
		75	200	"NRDA" mask
	NOP	76	000	
	NOP	77	000	
RSECT	IN	100	333	Read sector test
		101	011	Sector position channel
	CPI	102	376	
		103	300	∅ sector
	JNZ	104	302	Jump if not start of ∅ sect.
		105	100	to "RSECT"
		106	000	
RDTST	IN	107	333	Start of "NRDA" test
		110	010	Disk status channel
	ANA(A)/(B)	111	240	Test for "NRDA" status
	JNZ	112	302	Jump if "NRDA" false (=1)
		113	107	to "RDTST"
		114	000	
	IN	115	333	Input read data
		116	012	Disk data channel
	MOV(M)←(A)	117	167	Store data in memory (H+L)
	INR(L)	120	054	Increment L reg. (mem addr)
	JNZ	121	302	Jump if L reg. ≠ 0
		122	107	to RDTST
		123	000	
	MOV(A)←(D)	124	172	Move 377 byte to accum.
	OUT	125	323	Disenable disk by output logic 1 on
		126	010	D7 to disk enable chan. (end of read start of output)
	LXI(H+L)	127	041	Load H+L with:
		130	236	Starting addr of data stored by read program
		131	001	
OTST	IN	132	333	Test output device for busy
		133	000	Status chan. of terminal
	RLC	134	007	Test bit ∅, rotate into carry
	JC	135	332	Jump if carry (bit ∅ = 1)
		136	132	to "OTST"
		137	000	
	MOV(A)←(M)	140	176	Move data from mem(H+L)
	OUT	141	323	Output data
		142	001	Data channel for term
	INR(L)	143	054	Increment L register
	JNZ	144	302	Jump if L reg ≠ 0, output another byte
		145	132	to "OTST"
		146	000	
	MVI(A)	147	076	
		150	000	Enable disk
	OUT	151	323	
		152	010	
	JMP	153	303	
NOTE		*154	004	To "LDHD"
		*155	000	
		156		
		157		

*--For R/W-step loop change
Data at (000,154) to 037
Data at (000,155) to 001


START
(000,000)
ENABLE DISK Ø AND LOAD HEAD
(000,010
INITIALIZE REGISTERS:
REG B = "ENWD" MASK
REG C = NUMBER OF BYTES TO BE WRITTEN
REG D = FIRST BYTE WRITTEN (377)
(000,017)
BEGINNING OF SECTOR Ø?
NO
YES
(000,026)
TURN WRITE CIRCUIT ON
(000,032)
IS ENWD TRUE?
NO
YES
(000,040)
WRITE FIRST BYTE (377)
(000,043)
IS ENWD TRUE?
NO
YES
(000,051)
WRITE DATA BYTE FROM "C" REGISTER
(000,054)
DECREMENT C REGISTER
(000,055)
IS "C" REG = Ø ?
NO
YES
(000,060)
IS ENWD TRUE ?
(000,066)
WRITE ØØØ
ØS WRITTEN FOR REMAINDER OF SECTOR
WRITE PROGRAM
(000,071)
INITIALIZE REGISTERS
H + L REG = STARTING ADDRESS FOR STORING READ DATA IN MEMORY
B REG = "NRDA" MASK
(000,100)
BEGINNING OF SECTOR Ø ?
NO
YES
(000,107)
IS NRDA TRUE ?
NO
YES
(000,115)
INPUT READ DATA AND STORE IN MEMORY (ADDRESS SPECIFIED BY H + L)
(000,120)
INCREMENT L REGISTER
(000,121)
DOES L REGISTER = Ø ?
NO
YES
(000,124)
DISENABLE DISK DRIVE AND CONTROLLER
READ PROGRAM
TO OUTPUT PROGRAM
TO OUTPUT PROGRAM
(000,127)
INITIALIZE H + L REGISTER WITH STARTING ADDRESS OF READ DATA STORED IN MEMORY
(000,132)
IS OUTPUT DEVICE READY TO RECEIVE DATA ?
NO
YES
(000,140)
OUTPUT DATA FROM MEMORY ADDR. AS SPECIFIED BY H + L REG
(000,143)
INCREMENT L REGISTER
(000,143)
IS L REGISTER = Ø ?
NO
YES
(000,147)
ENABLE DISK Ø
FOR R/W + STEP LOOP
FOR R/W LOOP ONLY
(000,153)
JUMP TO STEP PROGRAM
JUMP TO WRITE PROGRAM
TO STEP PROGRAM
TO WRITE PROG.
OUTPUT PROGRAM

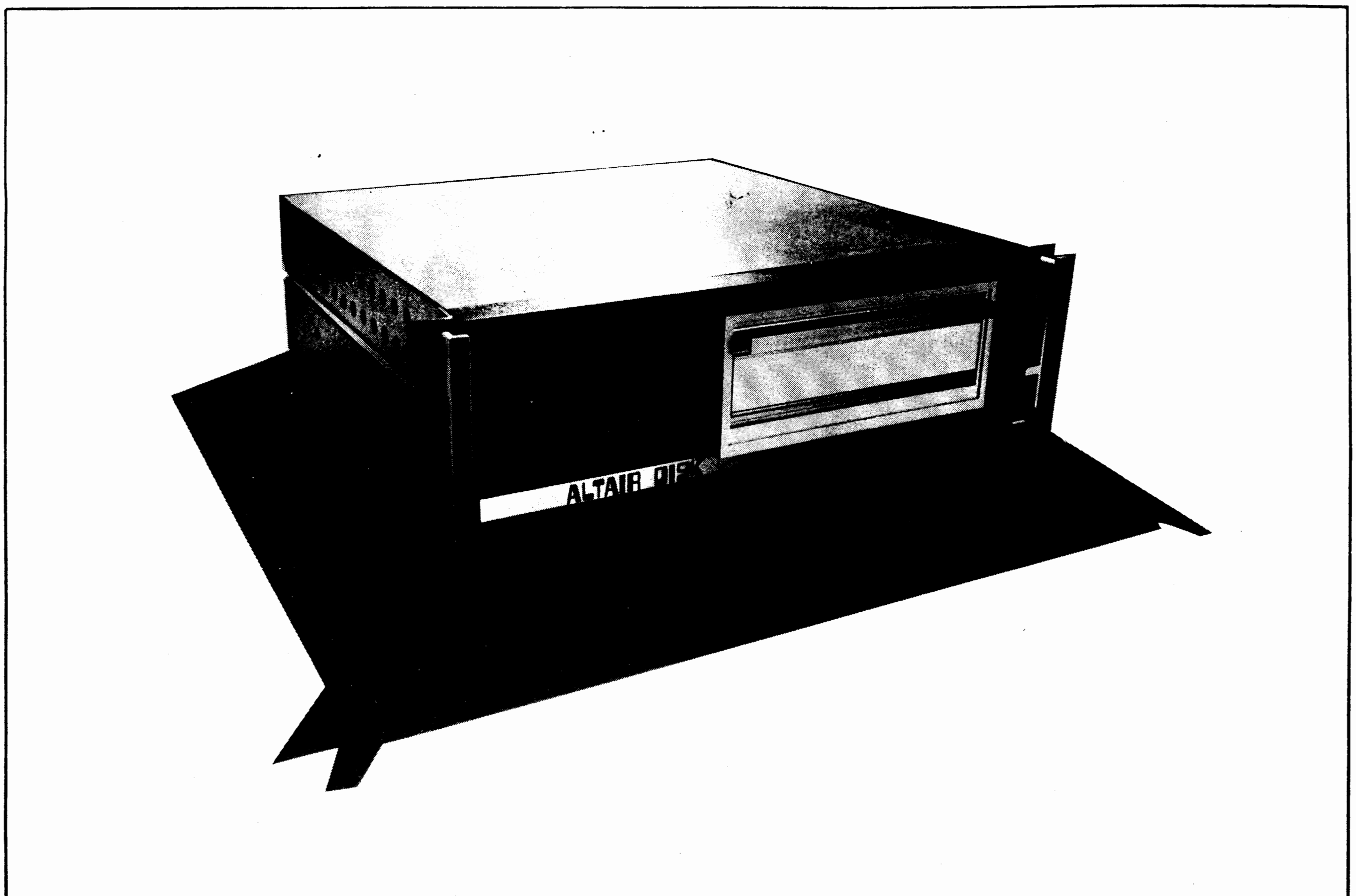
mits

2450 Alamo SE
Albuquerque, NM 87106

PRELIMINARY DOCUMENTATION RELEASE

THE FOLLOWING INFORMATION IS A PRELIMINARY RELEASE ONLY.

THE COMPLETE THEORY OF OPERATION WILL BE ADDED TO THE ASSEMBLY MANUAL AT A LATER DATE. THE OPERATORS MANUAL AND DOS DOCUMENTATION WILL BE DEVELOPED IN A SINGLE MANUAL ALSO. BOTH OF THESE, AND ANY UPDATES TO THIS ENTIRE SYSTEM DOCUMENTATION WILL BE SENT IMMEDIATELY UPON THEIR PRINTING DATE.



THE ALTAIR FLOPPY DISK SYSTEM

The *ALTAIR Disk* offers the advantage of nonvolatile memory, plus relatively fast access to data. The *ALTAIR Disk Controller* consists of two PC boards (over 60 I.C.s) that fit in the *ALTAIR* chassis. They inter-connect to each other with 20 wires and connect to the disk through a 37-pin connector mounted on the back of the *ALTAIR.* Data is transferred to and from the disk serially at 250K bits/sec. The disk controller converts the serial data to and from 8-bit parallel words (one word every 32 μ sec). The *ALTAIR CPU* transfers the data, word by word to and from memory, depending on whether the disk is reading or writing. The disk controller also controls all mechanical functions of the disk as well as presenting disk status to the computer. All timing functions are done by hardware to free the computer for other tasks. Since the *floppy diskette* is divided into 32 sectors, a hardware interrupt system can be enabled to notify the CPU at the beginning of each sector. Power consumption is approximately 1.1 amperes from the +8v (VCC) line for the two boards.

The Disk Drive unit, using a PERTEC FD400 mounted in an Optima case (5½" high—same depth and width as computer), includes a *power supply PC board* and a *Buffer/Address/Line Driver P.C. Board.* A cooling fan maintains low ambient temperature for continuous operation. The disk drive cabinet has two 37-pin connectors on the back panel, one is the input from the disk controller, the other is the output to additional disk drives. Up to 16 drives may be attached to one controller.

The 88-DCDD consists of the disk controller and one disk drive with an interconnect cable. The 88-Disk is one disk drive for adding storage capability to the 88-DCDD and includes the interconnect cable.

The *ALTAIR Disk Format* allows storage of over 300,000 bytes. Since the disk is hard sectored (32 sectors for each track), we write 137 bytes on each sector, 9 of which are used internally (track#, checksum) leaving 128 data bytes per sector, 4096 per track. One floppy diskette is supplied with each drive; extra floppies are available for purchase. A *software driver* for the floppy disk is available at no charge and is supplied with the disk as a source listing. The disk operating system—which has a complete file structure and utilities for copying, deleting and sorting files—costs extra. *Extended BASIC,* which uses random and sequential file access for the floppy disk, is also available.

Specifications

Rotational Speed	**360 rpm (166.7 ms/rev)**
Access Times	**Track to track, 10 ms** **Head settle, 20 ms** **Head load, 40 ms** **Average time to read or write, 400 ms** **Worst case, 1 sec**
Head Life	**Over 10,000 hours of head to disk contact**
Disk Life	**Over 1 million passes/track**
Data Transfer Rate	**250K bits/sec**
Power Consumption	**117VAC 110W**
Diskette	**Hard sectored, 32 sectors + index, Dysan 101 floppy disk, 77 tracks**

DISK OPERATORS MANUAL

I. DESCRIPTION OF SYSTEM

A) DISK SPEC SHEET

B) DISK SYSTEM BLOCK DIAGRAM DESCRIPTION

1. CONTROLLER BOARD 1:

Controller Board 1 does all input functions to the ALTAIR bus (Read Data, Sector Data, Status Information), as well as Control Addressing of all Disk to ALTAIR I/O.

2. CONTROLLER BOARD 2:

Controller Board 2 performs all output functions from the ALTAIR bus (Write Data, Disk Control, Disk Enable and Drive Selection).

3. INTERCONNECT CABLE:

An 18 pair flat cable with two 37 pin connectors, a male on one end, a female on the other. This cable connects the Disk Drive to the ALTAIR Disk Controller and "Daisy Chains" one Disk Drive to another in multiple Disk systems.

4. DISK DRIVE CABINET:

a) POWER SUPPLY:

The Disk Drive Cabinet contains a power supply for powering the Disk Buffer and Disk Drive.

b) THE DISK BUFFER:

The Disk Buffer board contains the necessary line drivers and receivers for interconnection with long cables to the Disk Drive. In addition, it contains the Disk Drive Address circuitry that allows the Controller to select one of 16 Disk Drives.

The Disk Buffer board also contains the line drivers for connection of multiple Disk Systems.

c) THE DISK DRIVE:

The Disk Drive, a Pertec FD-400, contains the mechanism and electronics that actually reads and writes data on the Diskette.

II. CONNECTION OF DISK SYSTEM:

A) CONTROLLER BOARDS:

1. Items Supplied:

   a) CONTROLLER BOARD 1 (white vert strips)
   b) CONTROLLER BOARD 2 (with short cable wired to it)
   c) CONTROLLER CABLE (with 37 pin on one end, 3 Molex connectors on the other end)
   d) Connector Mounting Bracket and Hardware

2. Connection of Controller Boards

   a) Take cover off ALTAIR (power off!)
   b) Feed Molex (flat) connector ends of Controller cable through hole in back of ALTAIR on connector panel: (37 pin connector outside chassis, molex connectors inside chassis).
   c) Lay board 1 flat in front of you on the ALTAIR chassis with components up and stab connector to your right (as facing the front of the ALTAIR).
   d) Take the short wired cable of board 2 and connect it to the 20 pin connector on board 1 (note polarization key of connector and missing pin on the PC board).
   e) Place board 2 flat, to the left of board 1.
   f) Connect 20 pin Molex connector on the Controller cable to the 20 pin connector on board 2. Note Keying.
   g) Take the 10 pin connector on the Controller cable with the orange and yellow wires connected to it and connect it to the 10 pin connector on board 2. Note Keying.
   h) Take the remaining 10 pin connector on the Controller cable with white and gray wires on it and connect it to the 10 pin connector on board 1. Note Keying.
   i) Take both boards, hold together and slide into slots, with board 1 on right, board 2 on the left. Be sure wires from connector go out between card guides, and do not catch on card guides.
   j) Push cards firmly into connector in ALTAIR mother board.
   k) Install 37 pin connector in bracket and on back of ALTAIR, straddling 2 connector holes. Use #4-40 x 5/16 screws, lockwashers and #4-40 nuts.

B) DISK DRIVE CONNECTION TO ALTAIR:

Take the 6 ft. flat cable with 1 male and 1 female connector; connect male end to Disk Controller connector on ALTAIR, and female end to connector on the Disk Drive marked "To Controller".

C) MULTIPLE DISK DRIVE CONNECTION:

1. With multiple Disk Drives, the Disks should have sequential addresses (i.e., for a 3 drive system you should have Disks with addresses Ø, 1, and 2). They may be connected in any order. There serial # sticker has the Disk Address written on it. The Disk Address is determined by four jumper wires in the Disk Buffer P.C. card inside the Drive, and may be changed.

2. Connect the Disks by using the 6 ft. flat cable. Connect the male connector to the connector marked "From Next Disk" on the Disk Drive connected to the Controller. The other end of the cable connects to the next Disk Drive connector marked "To Controller". This procedure is repeated for added Disk Drive.

III. USING THE DISK DRIVE:

A) DISKETTE INFORMATION:

1. Always keep Diskette in envelope when not in use.
2. Keep Diskette away from heat, magnetic fields (flourescent lights, power transformers, etc.) and dust and dirt.
3. Never touch recording surface of Diskette (opposite label side).
4. Always mark your Diskette with what is on them. Use adhesive labels, but don't write on them after they are attached to the Diskette.
5. The Diskette used is hard Sectored (32 Sector holes, 1 index hole). Blank Diskettes are available from MITS for $15.00 each. The Diskettes are <u>not</u> IBM compatible.

B) OPERATING THE DISK DRIVE:

1. Open door to Disk Drive by pulling out and down.
2. Insert Diskette into Drive with label side up, making sure it catches on retaining tab.
3. Close door to Disk Drive.
4. If Disk power is on, wait 10 seconds, after closing door before activating any programs to access the Disk. Wait 10 seconds after turning power on with Diskette in Drive before activating any programs to access the Disk. This is to allow motor speed to stabilize.
5. <u>NEVER</u>: open Disk Drive door or turn power off when Disk Enable and Head Load lights are on. There would be a good possibility that you would interrupt the software during a write function, and destroy data on the Diskette.
6. Consult software documentation on methods used to load basic or use software. For applications where the user wishes to write his own software. See last section, "Controller I/O Information".

ALTAIR DISK CONTROLLER I/O INFORMATION

A. Address codes for I/O

		Address		Mode	
1.	DSTEN	Ø1Ø	08H	Out	Select, latches and enables controller and disk drive.
2.	DSTEN	Ø1Ø		In	Indicates status of disk drive and controller.
3.	POSCL	Ø11	09H	Out	Controls disk function.
4.	POSCL	Ø11		In	Indicates sector position of disk.
5.	DDATA	Ø12	ØAH	Out	Write data.
6.	DDATA	Ø12		In	Read data.

B. Definitions (In order as listed above)

1. Selection of Disk Drive "OUT" on CH #Ø1Ø DSTEN

DØ LSB D1 D2 D3 MSB	Enables 1 of 16 drives (each drive has a unique address, selected by 4 jumper wires) and enables controller (on disk drive buffer P.C. card).
D4 D5 D6	Not used, don't care.
D7	Clears disk control if set to 1 (D0-D6 don't care). Disables disk control. Disk control also cleared by opening door of disk drive or turning disk drive power off.

NOTE: a) If disk drive door is open, drive and controller cannot be enabled.

b) If disk power is off, drive and controller cannot be enabled.

c) If disk interconnect cable is not connected between the controller and the drive, drive and controller cannot be enabled.

10Q 08H

2. Status (Ø1Ø - INP) indicates disk status when drive and controller enabled. Also gives valid "INTE" status (D5) from the ALTAIR bus when controller enabled.

True condition = 0, False = 1.

All false if disk and controller are not enabled, and all false if no disk in drive.

01 D0 - ENWD - Enter new Write data - indicates write circuit is ready for new data byte to be written. It occurs every 32 μs and starts 280 μs after sector true (when Write enabled). It is reset by outputting to the Write data channel (012).

02 D1 - Move Head - Indicates head movement allowed when true (step IN, step OUT,). Goes false for 10 ms, true 1 ms, false 20 ms after step command. May step every 10 ms. Goes false for 40 ms after head load. Goes false during Write and 475 μs after Write to allow completion of trim erase.

04 D2 - HS - Head Status - True 40 ms after head loaded or step command (if stepping with head already loaded). Indicates when head is properly loaded for reading and writing. Also enables sector position channel when true.

D3 - Not Used, = Ø.

D4 - Not Used, = Ø.

20 D5 - INTE - Indicates interrupt enabled.

40 D6 - TRACK 0 - Indicates when head is on outermost track.

80 D7 - NRDA - New read data available - indicates that the read circuit has 1 byte of data ready to be taken from the read data channel (012). After the SYNC* bit is detected, it occurs every 32 μs and is reset by an input instruction on channel Ø12. The byte containing the SYNC bit is the first byte read from the disk.

* See "Write Enable"

3. Control (Ø11 - Out) - Controls Disk operations when disk drive and controller enabled. A true signal, logic 1, on a data line will control the disk as follows:

01 DØ - Step IN - steps disk head in one position to higher numbered track.

02 D1 - Step OUT - steps disk head out one position to lower numbered track.

04 D2 - Head Load - loads head onto disk - enables sector position status.

08 D3 - Head Unload - removes head from disk surface, may be unloaded immediately after "Write Enable" (write and trim erase circuits hold head loaded until through).

10 D4 - IE - Interrupt Enable - enables interrupts to occur when SRØ true (see sector definition).

20 D5 - ID - Interrupt Disable - disables interrupt circuit. Interrupt circuit also disabled by clearing disk control.

40 D6 - HCS - Head Current Switch - must be true when outputting a write instruction with the head on tracks 43-76. This reduces head current and optimizes resolution on inner tracks (automatically reset at end of writing a sector).

80 D7 - Write Enable - initiates write sequence as follows:

1. Disk selected and enabled, head loaded, enabling sector status.
2. (Sector True) detected for desired sector, write circuit enabled by software.
3. 200 μs from Write Enable, trim erase automatically turned on. 280 μs from start of sector, "ENWD" goes true, sync byte written by software.
4. First byte written always has most significant (D7) bit A "1" (SYNC Bit) (most sifnificant bit written first).
5. ENWD goes true every 32 μs. MAX. no. of data bytes per sector 137 (including SYNC).
6. Last or 138th byte written must be a 000. This will be written for the remainder of the sector. Ignore "ENWD" from this point to end of sector.
7. At end of sector, the write circuit automatically disabled, trim erase disabled 475 μs later.

NOTE: a) Write circuit will continue writing last byte outputted on CH #Ø12 to the end of that sector.

b) Head may be unloaded anytime during write cycle if no read or write function is expected after current write cycle. Once Write is enabled, it holds the head loaded for the required time. (For writing and trim erase).

4. Sector Position (Ø11 - INP) with disk drive and controller enabled, and 40 ms after head is loaded, the sector information is as follows:

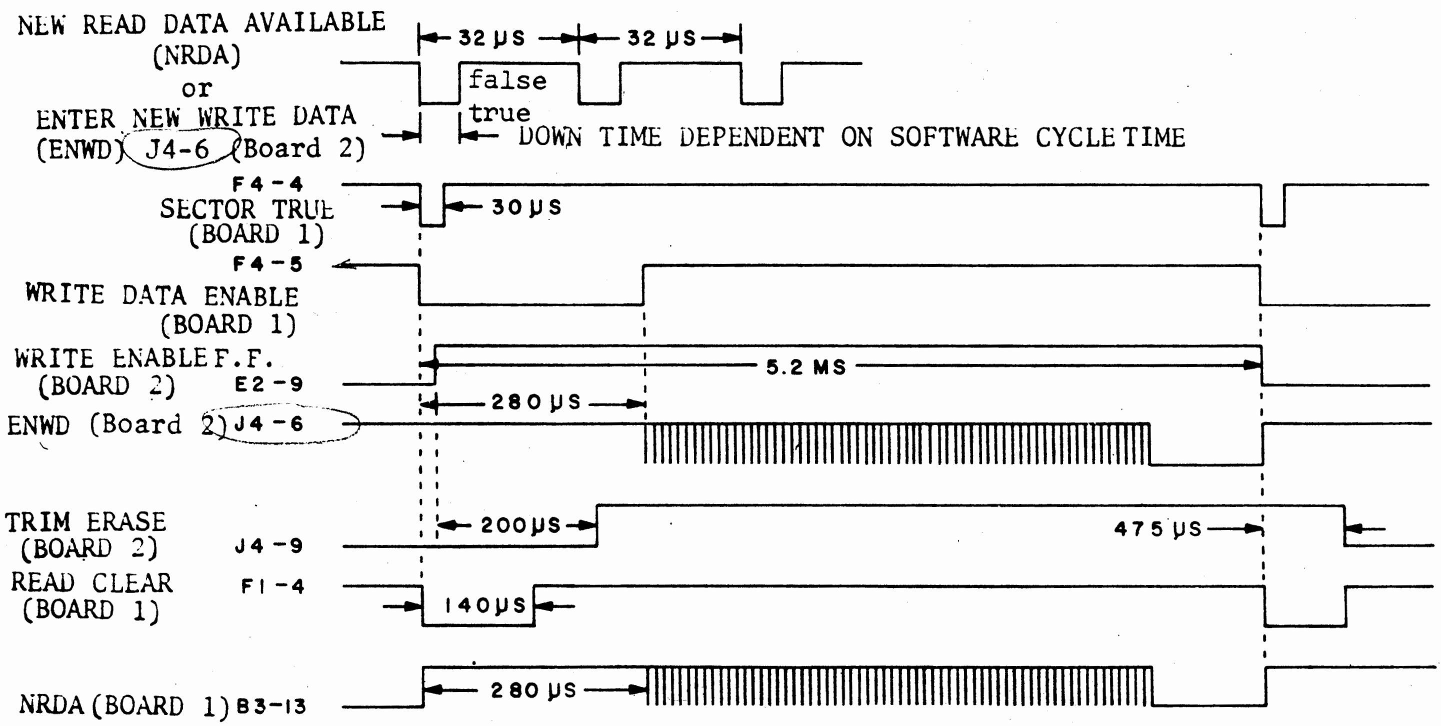
DØ - SRØ - Sector True - True when = 0, and is 30 μs long. The write mode should begin as close as possible to the time that DØ goes true. Write data will be requested 280 μs after DØ goes true. Read data will be available 140 μs after SRØ goes true.

SECTOR #	Ø	1	2	3.................................31
D1-SR1-	Ø	1	Ø	1.................................1
D2-SR2-	Ø	Ø	1	1.................................1
D3-SR3-	Ø	Ø	Ø	Ø.................................1
D4-SR4-	Ø	Ø	Ø	Ø.................................1
D5-SR5-	Ø	Ø	Ø	Ø.................................1

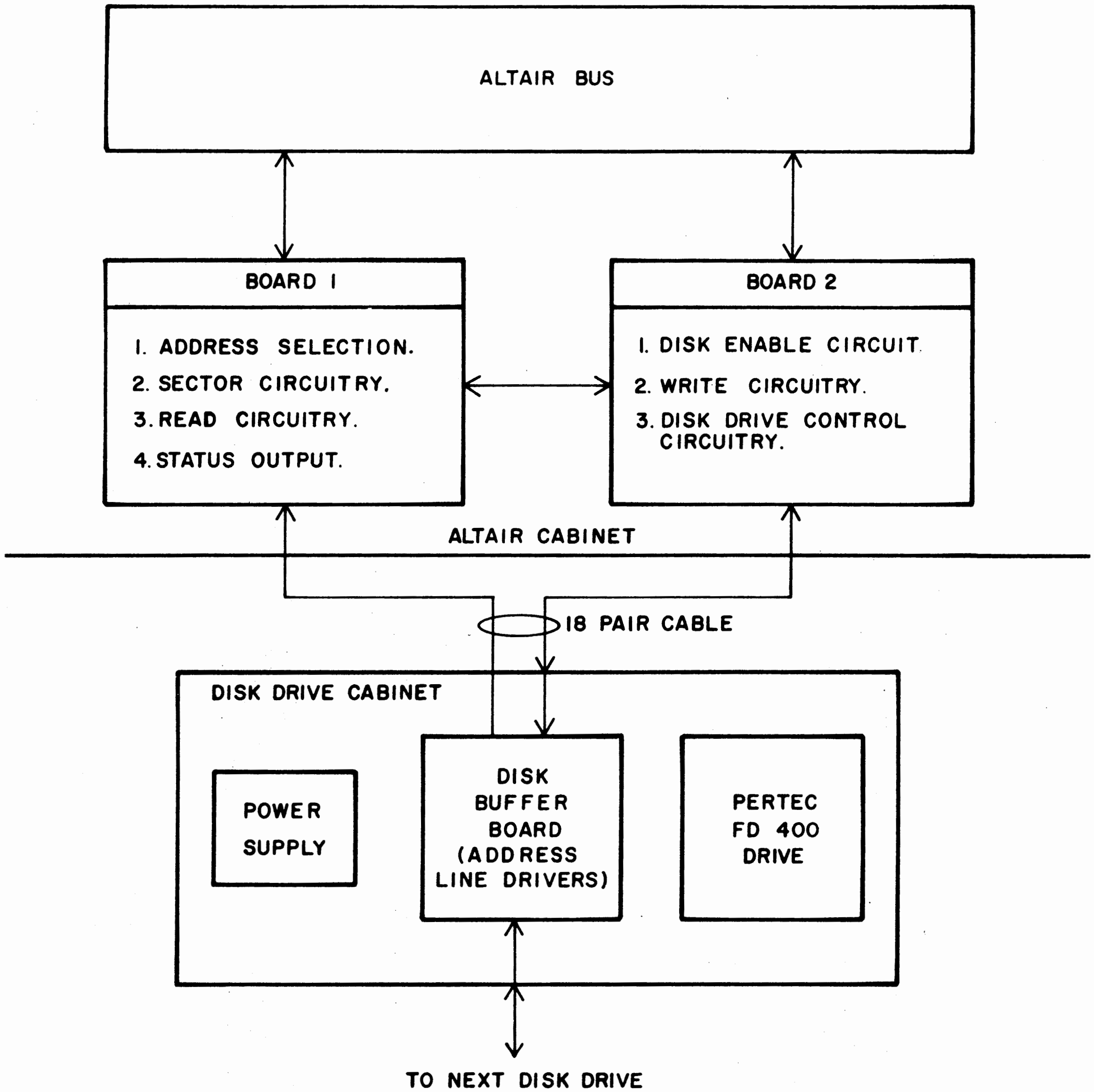
5. Write Data (Ø12-OUT) Outputted on the "ENWD" status request.

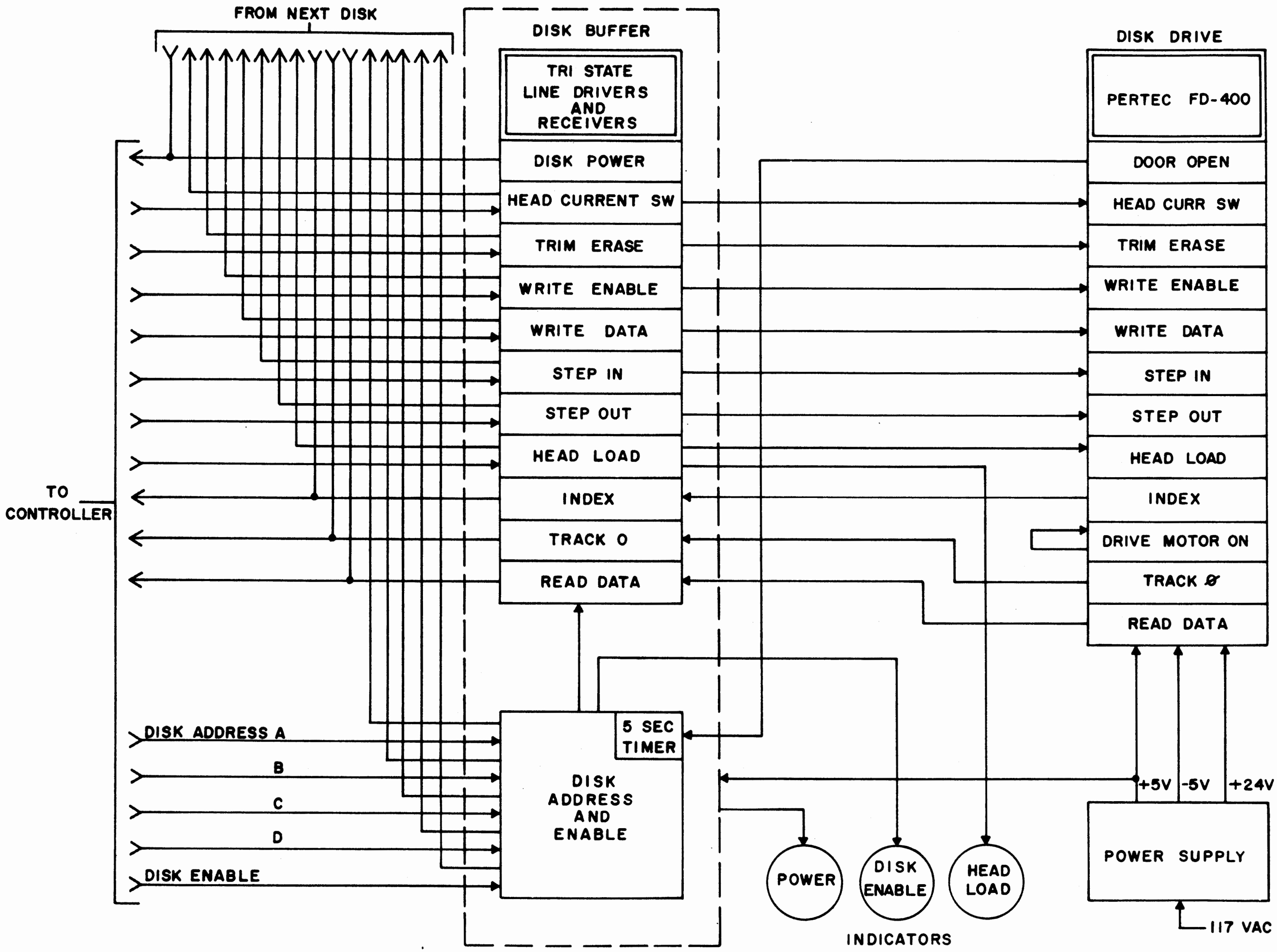
6. Read Data (Ø12-IN) Inputted on the "NRDA" status flag.

READ/WRITE TIMING
DURING READ OR WRITE FUNCTION



DISK SYSTEM BLOCK DIAGRAM

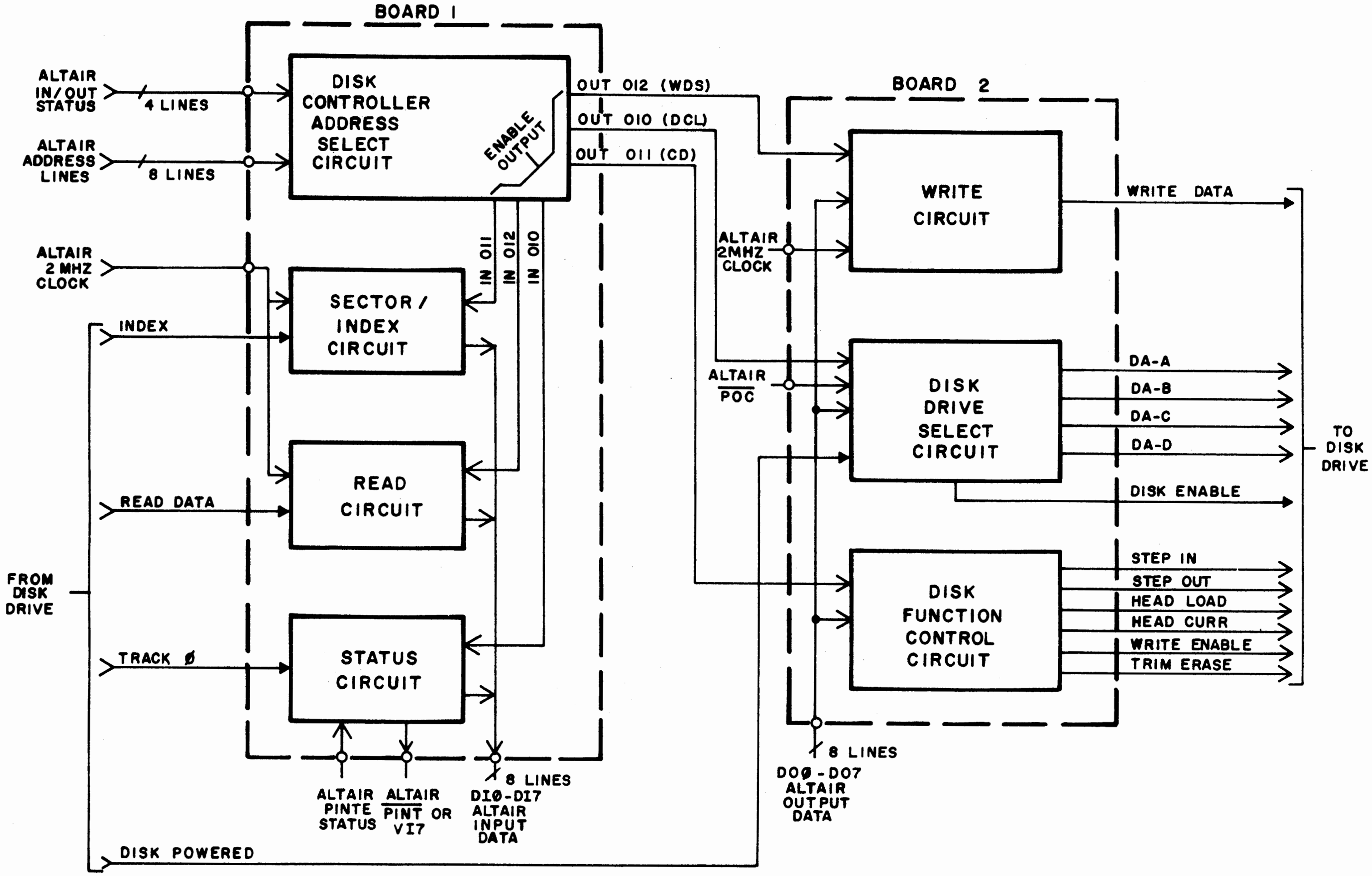




88 DISK BLOCK DIAGRAM

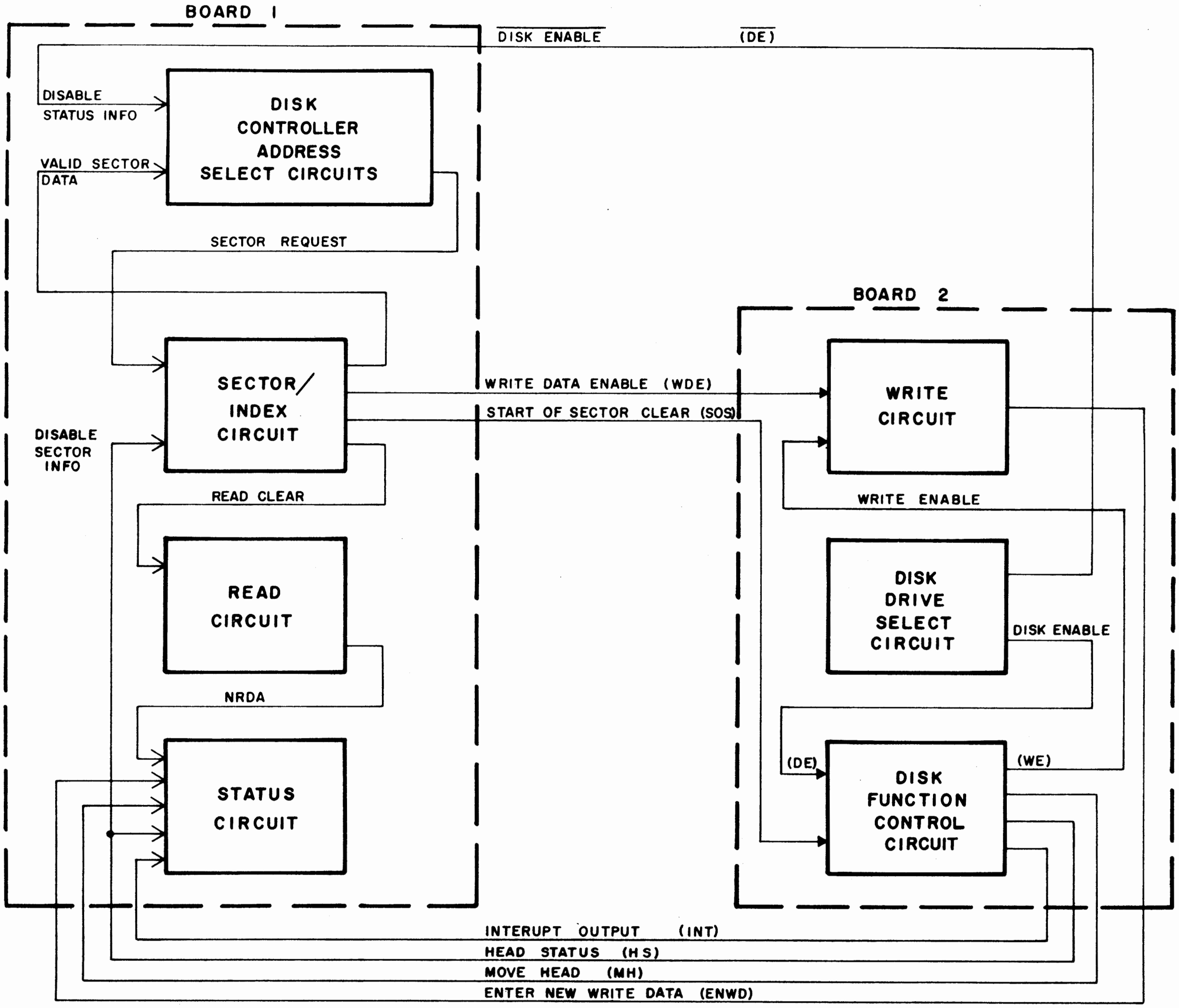
DISK CONTROLLER BLOCK DIAGRAM

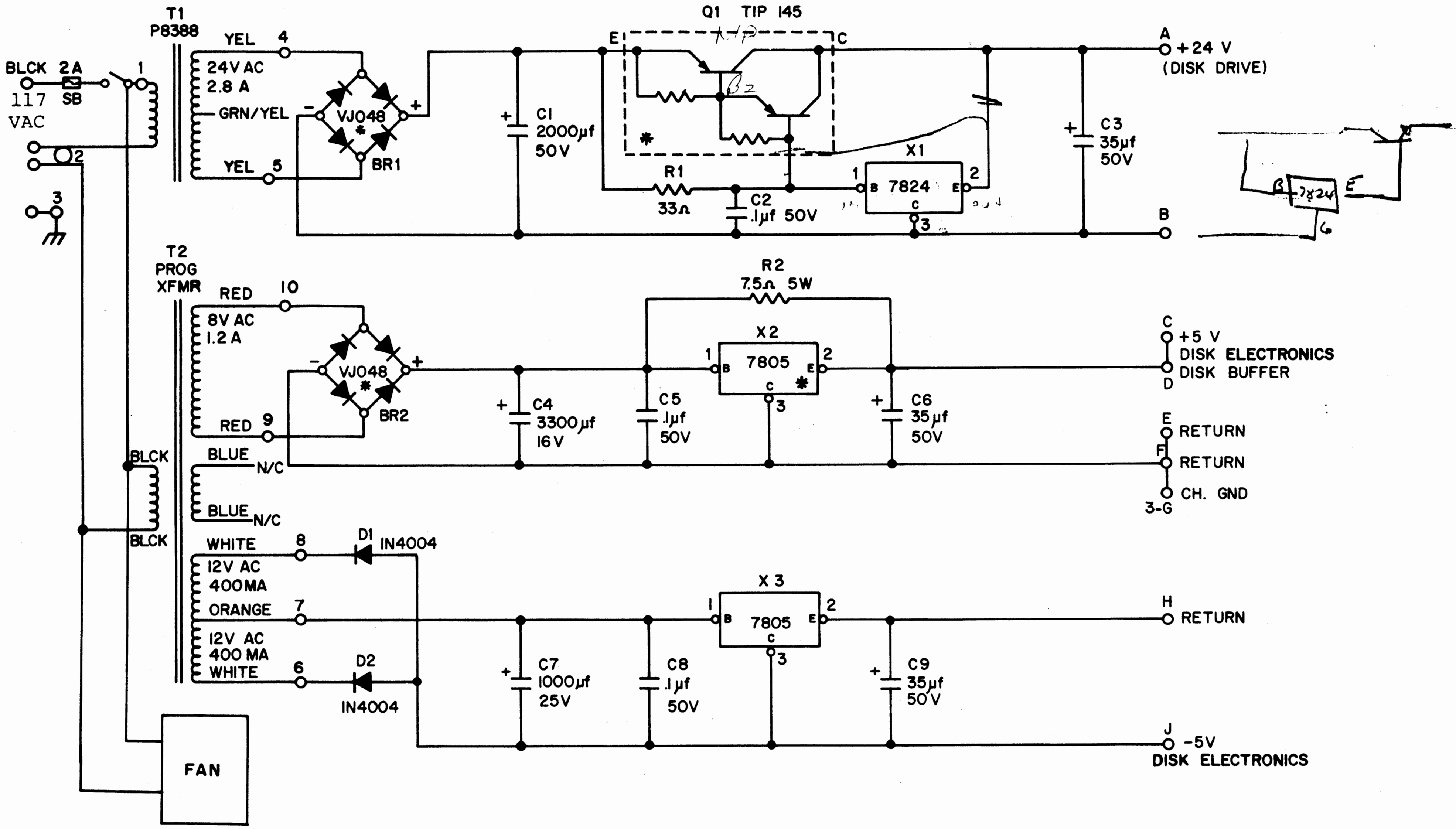
SHEET I EXTERNAL CONNECTIONS AND ADDRESS SELECT



DISK CONTROLLER BLOCK DIAGRAM

SHEET 2 INTERNAL CONNECTIONS





* HEATSINK ON CHASSIS

DISK DRIVE POWER SUPPLY

88-DISC
PARTS LIST
JANUARY, 1976

BAG 1

1	74L30	101082
2	7805	101074
1	7824	101079
4	8T97	101040
1	8T98	101045
1	9601	101033

BAG 2

4	.1mf 12v	100348
3	.1mf 50v	100312
3	33mf 50v	100311
1	500mf 15-25v	100310
1	1000mf 25v	100365
1	2200mf 50v	100376
1	3300mf 16v	100315

BAG 3

3	150 ohm ¼w	101915
17	330 ohm ½w	101926
1	1k ½w	101928
1	39k ½w	101967
1	7.5 ohm 5w	101987
1	33 ohm ½w	101921

BAG 4

17	220 ohm ½w	101925
3	RL21	100702
2	VJ048	100711
2	IN4004	100718
1	TIP 145 or 146	102820
1	IN914	100705
1	Mica Washer & Bushing	

BAG 5

1	12ft. 18 Pair Cable	103066
2	6ft. #20 Black	103062
3	2ft. #20 Orange	103063
2	3ft. #26 White	103060

BAG 6

8	#4-40 x 5/16" Screw	100912
2	#4-40 x ½" Screw Flat Head	100903
2	#4-40 x 1" Screw	100913
10	#4-40 Nut	100932
8	#4 Lock Washer	100941
4	#4 Flat Washer	100940
6	#6-32 x 3/8" Pan Head Screw	100925
6	#6-32 x ½" Pan Head Screw	100918
4	#6-32 x 5/8" Pan Head Screw	100916
2	#6-32 x 3/4" Pan Head Screw	100935
4	#6-32 x 1" Pan Head Screw	100919
4	#6-32 x 2" Flat Head Screw	100937
27	#6-32 Nut	100933
35	#6 Lock Washer	100942
1	#6 Ground Lug	101801
2	.15" Spacer	101823
6	5/16" Spacer	101829
2	.6" Spacer	101824
4	#6 Flat Washer	100943
2	#6-32 x ¼" Screw	100917

BAG 7

1	Heat Sink	101775
1	Heat Sink Spacer 5½"	101835
1	Disk Drive Spacer 9"	101841
1	Right Angle Bracket	101717
1	Strain Relief	101719
1	Terminal Block	101868
30	Insulated Terminals	101803
1	Fuse Holder	101813
2	DC37S Connector	102114
2	DC37P Connector	102115
2	DC37 Connector Cover	101799
1	Toggle Switch ST1-1C	101879
1	44 Pin Edge Conn. & (Key Pin 101660)	101800
15	Fastwrap	103037
1	Heat Sink Grease	
1	Fuse 2ASB 3AG	101762

MISC:

1	Power Cord 3 Wire	101742
1	Disk Mechanism (Pertec)	FD-400
1	Case	100511
1	Disk Rail	101862
1	Fan Filter	101757
1	Fan and (4) clips	101869
1	P-8388 Transformer	102612
1	Programmer Transformer	102609
1	Diskette	101712
1	Power Supply PC Board	100171
1	Buffer PC Board	100172
1	"ALTAIR DISK" Nameplate	101808
1	Serial Number Sticker	101833
1	Assy, Theory, OP Manual	101531

88-DCDD
PARTS LIST
JANUARY, 1976

BAG 1

5	74L00	101080
6	74L02	101072
8	74L04	101073
3	74L10	101081
1	74LS11	101089
1	74L20	101039
1	74L30	101082
7	74L73	101084
2	74LS74	101088
5	74L75	101075
1	7493	101030
8	74123	101060
1	74164	101091
1	74166	101092
3	93L16	101093
5	8T97	101040
1	8T98	101045
2	7805	101074

BAG 2

37	.1mf 12v 20%	100348

BAG 3

1	430pf 500v 5%	100322
1	910pf 500v 5%	100356
2	.001mf 1kv 20%	100328
1	.01mf 16v 20%	100321
2	.047mf 100v 5%	100332
2	.1mf 100v 5%	100339
1	.22mf 100v 5%	100349
2	.68mf 100v 5%	100343
1	1.0mf 100v 5%	100373
1	4.7mf 16v	100351
1	10mf 16v	100350
4	33mf 16v	100326

BAG 4

4	220ohm ½w 5%	101925
4	330ohm ½w 5%	101926
5	1k ½w 5%	101928
1	5.6k ½w 5%	102091
1	6.8k ½w 5%	101931
7	10k ½w 5%	101932
2	15k ½w 5%	102083
1	16k ½w 5%	101942
3	20k ½w 5%	101940
1	39k ½w 5%	101967

BAG 5

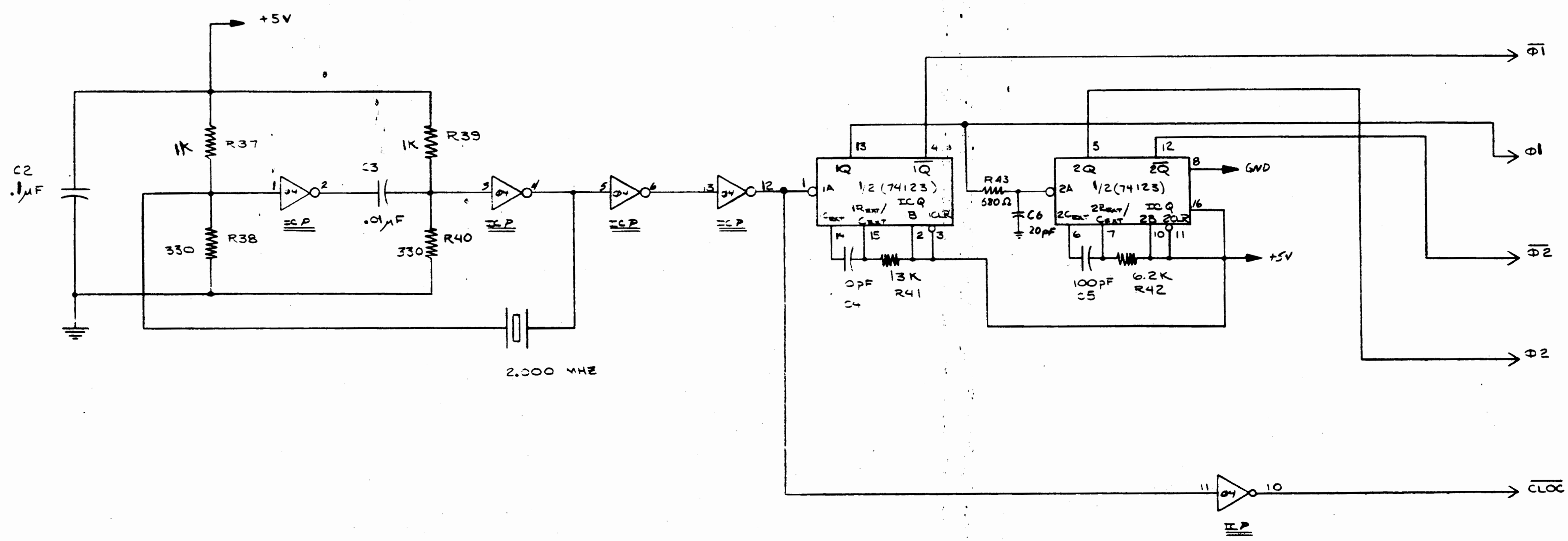
2	IN914	100705
10	#6-32 x 3/8" Screw	100925
2	#6-32 Nut	100933
2	#6 Lock Washer	100942
4	#4-40 x 3/8" Screw	100908
4	#4-40 Nut	100932
4	#4 Lock Washer	100941
1	3ft. 18 Pair Cable	103066
1	37 Pin Adapter Bracket	101795

BAG 6

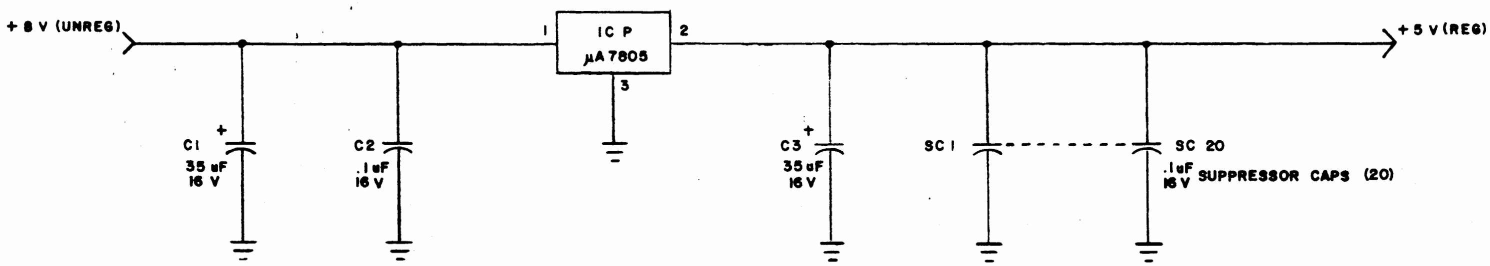
6	Buss Strips	101805
2	100 Pin Edge Connector	101864
1	DC37S Connector	102114
2	10 Pin Right Angle Wafer	101798
2	20 Pin Right Angle Wafer	101788
2	10 Pin Connector	101720
2	20 Pin Connector	101789
70	Terminal Pins	101723
4	Polarizing Keys	101791
2	Fastwrap	103037
1	Heat Sink Grease	
2	Heat Sink (large)	101870
4	Card Guides	101714

MISC:

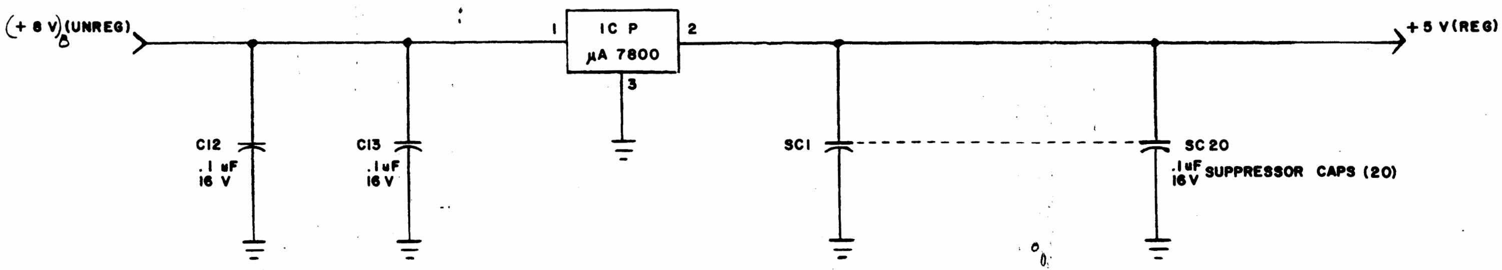
1	Controller PC Board 1	100173
1	Controller PC Board 2	100174



880-102
SYSTEM CLOCK



880-108
1K STATIC MEMORY ON-BOARD REGULATOR



880-104
DISPLAY/CONTROL ON-BOARD REGULATOR

Reprinted from July, 1976, Computer Notes

Disk Hardware Notes

By Tom Durston

If you are having difficulties with your 88-DCDD hardware, follow these guidelines for servicing:

A. Controller Boards:

1. On Controller Board #1 be sure the bus strips are soldered on both the top and bottom of the P.C. Board. Do not apply pressure to bus strips after installation.

2. On Controller Board #1 jumper the top end of R16 (VHB) to the track from pin 7 of IC F2 (on back of card). This ties floating inputs of sector logic high to prevent noise pickup.

3. On Controller Board #1 check the track from Pin 9 of IC H1 where it goes through the board on the plated hole. Some P.C. Cards had shorts to the adjacent track on the back of the card.

4. On Controller Board #1 check jumper wires to be sure there are no shorts to bus strips (insulation on wires melted), and check jumper wires for correct wiring.

5. On both Board 1 and 2 check Stab Connector for shorts on fingers. File at an angle along the length of the Stab Connector and the bevel edge of the card to remove any shorts.

6. Be sure all interconnect cables are wired correctly and the pins are making good contact.

7. Check one shot timing on both boards as follows, using the Disk Test Program that appeared in April '76 Computer Notes, pages 12 and 13.

   a) Controller Board #1:

FUNCTION	IC and PIN #	POSITIVE PULSE WIDTH RANGE
Read Clock Mask	IC A1 Pin 13	0.7us to 1.2us
Read Data Window	IC A1 Pin 5	2.6us to 2.9us
Sector Pulse Mask	IC E1 Pin 13	150us to 600us 370us
Index Pulse Window	IC E1 Pin 5	3.3ms to 4.5ms
Read Clear	IC F1 Pin 13	130us to 150us
Index Pulse Verification	IC F1 Pin 5	3.3ms to 4.5ms
Sector True	IC F4 Pin 13	20us to 40us
Write Data Enable	IC F4 Pin 5	250us to 300us

b) Controller Board #2:

FUNCTION	IC and PIN #	POSITIVE PULSE WIDTH RANGE
Repeat Step OK (Status)	IC A1 Pin 13	0.4ms to 0.8ms
Step Inhibit 1 (Status)	IC A1 Pin 5	9.5ms to 11.5ms
Head Settle	IC B1 Pin 13	35ms to 70ms
Step Inhibit 2 (Status)	IC B1 Pin 5	17ms to 30ms
Trim Erase Start Delay	IC B2 Pin 13	180us to 225us
Trim Erase End Delay	IC B2 Pin 5	420us to 520us
Disk Enable Timer	IC B3 Pin 13	1.5us to 4.5us
Disk Power Disable	IC B3 Pin 5	1.5us to 4.5us

c) If the measured time constants are not within the specified tolerance, vary the resistor value for the one shot affected.

d) We have had difficulty using National 74123 ICs for B3 on Board #2. Replace with Signetics or TI ICs if you suspect problems.

8. If you are using 4K Dynamic cards, be sure they are using only one wait state. See May '76 Computer Notes, pages 9 and 10.

9. Check the Power Supply to be sure the negative peaks of the +8V unregulated do not go below +7V.

B. Disk Drive Chassis:

1. On the Buffer Card the most common difficulty is incorrect wiring or incorrectly installed ICs.

2. On the Power Supply Board be sure X1 and X3 are properly installed as indicated on the errata sheet.

3. If you suspect difficulty with the Disk Drive, DO NOT attempt to service it. Any work done on the Pertec FD-400 will void the warranty. Typical service charges for customer damaged FD-400's are $100.00.

4. Do not plug the FD-400 connector in backwards. Be sure to install the polarizing key as the instructions indicate. Plugging in the connector backwards will destroy 5-10 ICs and will cost at least $100.00 for repair.

5. If you must ship the Pertec FD-400 or complete Disk Drive Unit, reinstall the Disk door block or strap. Any damage to the mechanism as a result of incorrect shipping typically costs the customer $100.00 in repair charges.

6. Our dealers now have Pertec FD-400 service manuals. If you suspect difficulty with the FD-400, contact your nearest dealer for his advice and service.

7. If you can't remedy the difficulty, don't try to save postage by just returning the FD-400 alone. Please return your complete 88-DCDD including Cables, Controller Boards, and Drive Chassis. This will allow us to check your system out completely and save you time, money, and hassle.

26) The new Checksum Loader will display 7647 on the address lights when running properly. When an error occurs (checksum "C"-bad data, memory "M"-data won't store properly, overlay "O"-attempt to load over top of the checksum loader) the address lights will then display 7637. The ASCII error code is stored in the accumulator (A) and is being output on channels 1, 21, and 23.

27) When the tape finishes reading, the MONITOR should start up and print the normal prompt - ? . If you are loading from cassette, STOP the player immediately so other files can be loaded.

Appendix F.

Audio Cassette Users

The following table shows the order and length of files on the cassette of Package II.

Program Name	Time from Start of Tape (in seconds)
MONITOR	13 - 125
ASM	120 - 230
EDT	240 - 310
AM2	320 - 415
DBG	430 - 510

When recording a new file on a cassette, position the cassette after the last file. When using either the editor or assembler to dump out a file, start the recorder a few seconds before flipping sense switch 15. A gap of this type should be inserted between all files on a casette.

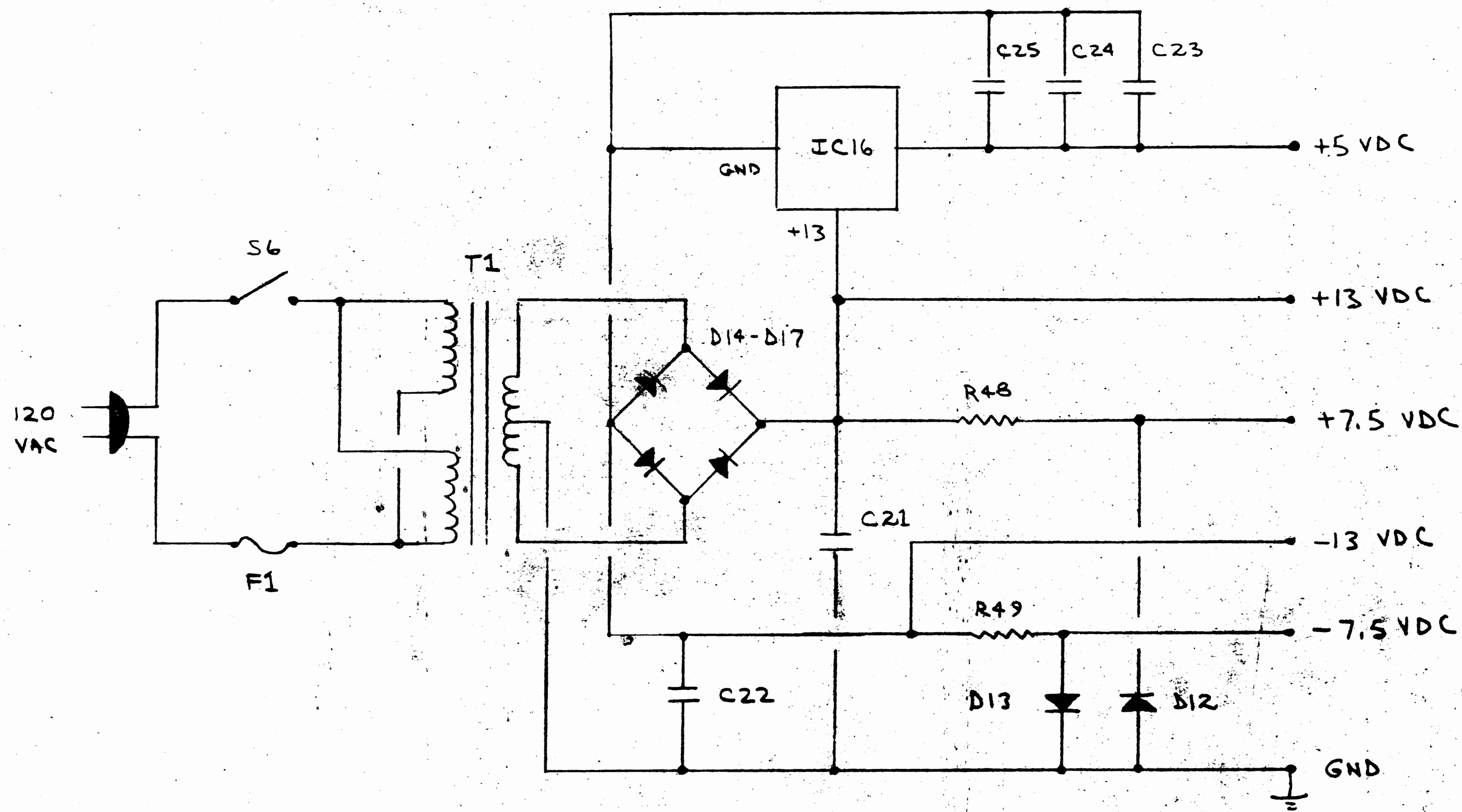
ASCII Line Input

The following describes the action taken for various special characters.

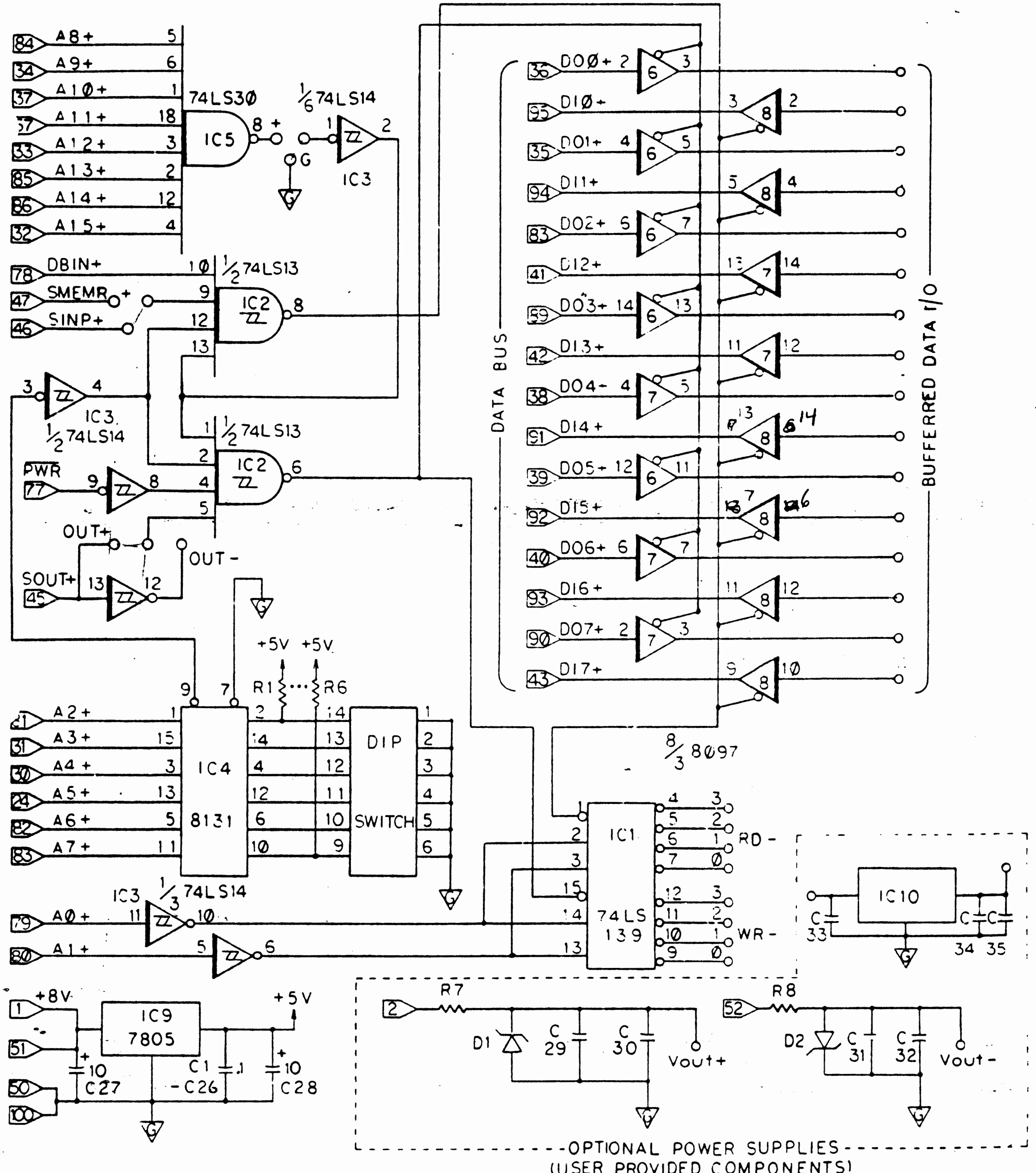
<CR> - Ends a line. The monitor returns to the calling program when typed. It is not counted in the line length returned. A line feed is also written out if input is being echoed.
<LF> - ends a line. Only a line feed is echoed. See above.
<ESCAPE> - Ends a line. $ is echoed. See above.
Octal 0 - Ignored
<Control A> - rubs out complete line typed.
<RUBOUT> - Backspaces one character for each one typed.
<Control> z - End of file, branches to address given in control block.

Interrupt I/O

Package II now supports input interrupts from the terminal device. One I/O card in the Altair can be wired for input interrupts directly to the bus interrupt line(PINT), or to the lowest priority on the vectored interrupt card. If the terminal is set for interrupts, typing a <Control C> will stop execution of a program and return to the monitor. All registers are saved in the register save area as described in the monitor section of this manual.

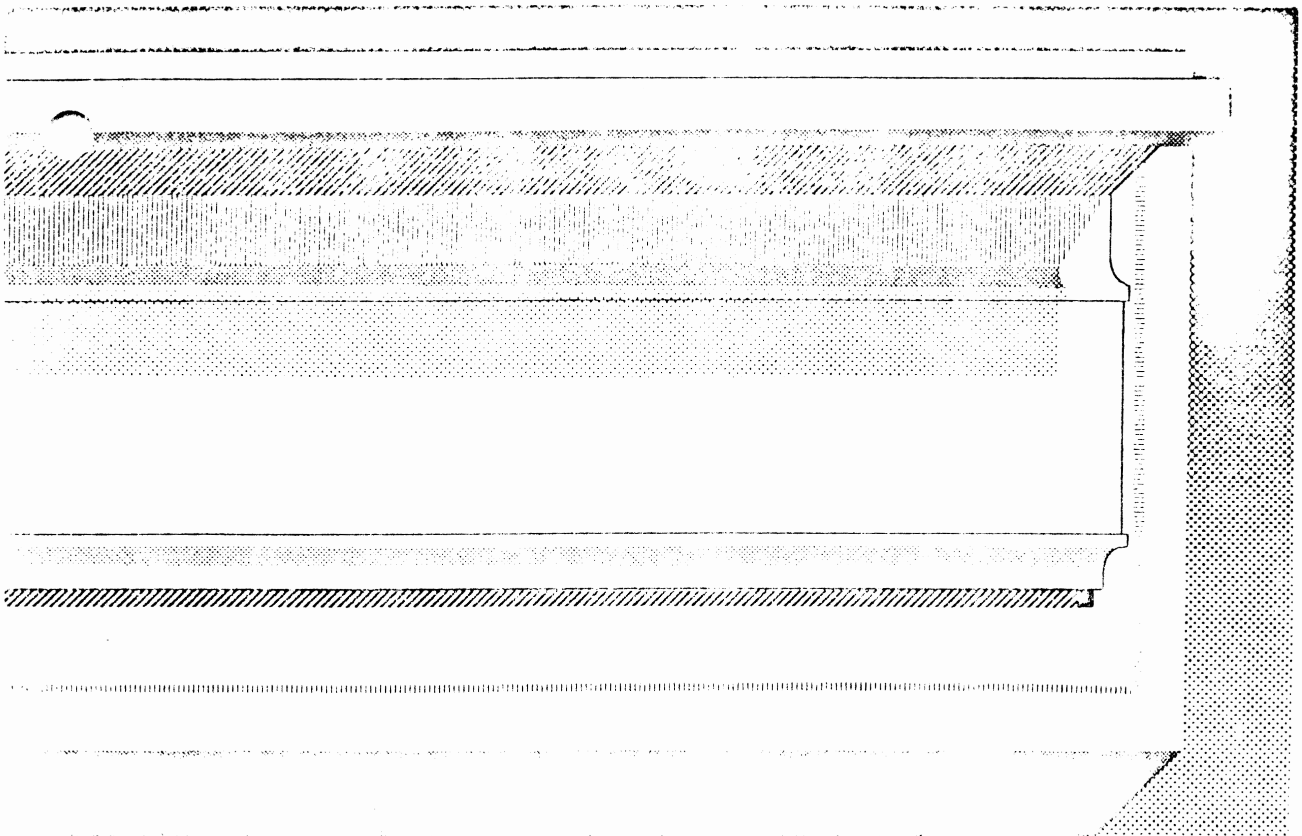


SWTPC AC-30 AUDIO CASSETTE POWER SUPPLY SCHEMATIC



POLY I O SCHEMATIC

mits Altair Floppy Disk



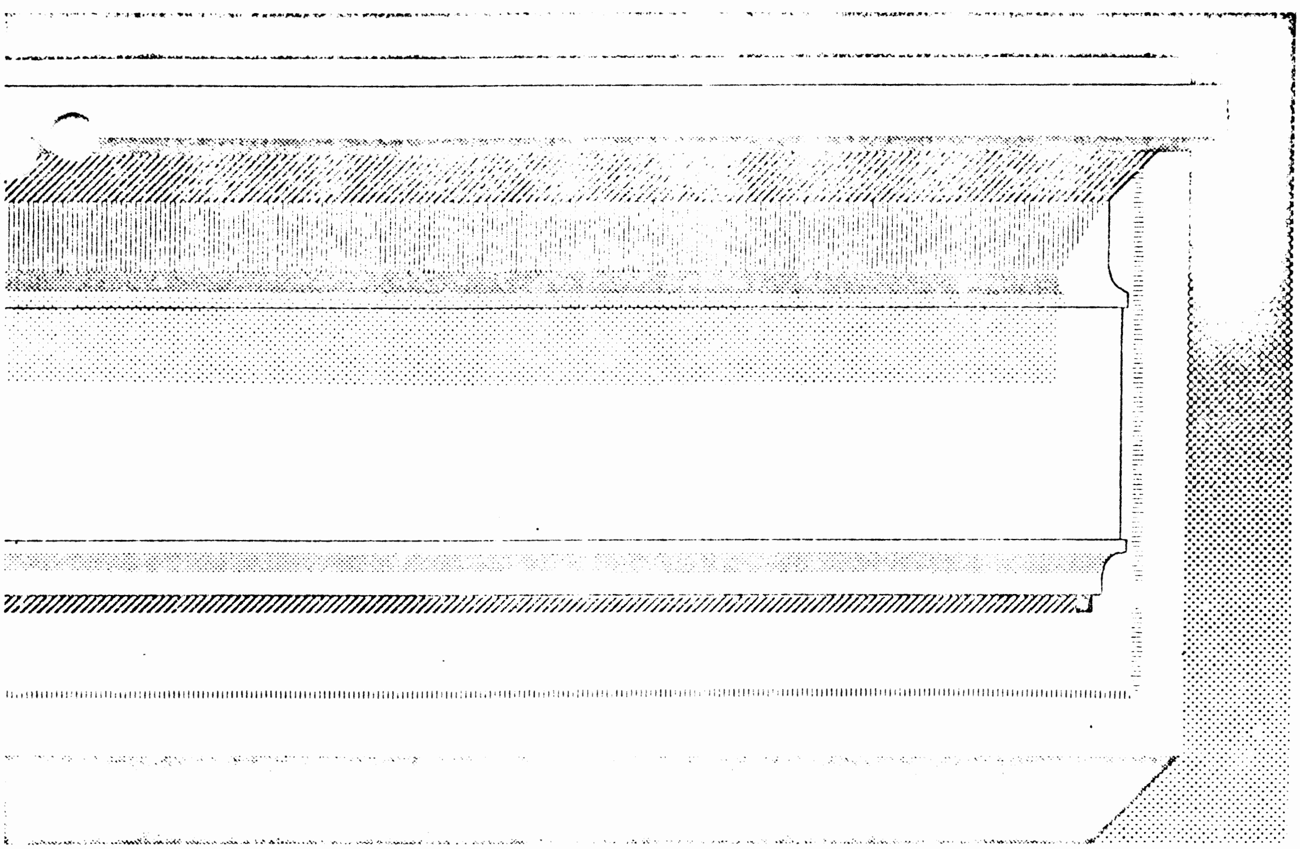
drive & controller • hardware documentation

PRELIMINARY DOCUMENTATION RELEASE

This manual is incomplete in its present form. This page and an additional section will be sent to you within a short period for insertion.

This documentation contains the entire assembly and check-out information for both the disk controller and drive units. The Theory of Operation and some additional information will be in the insertation.

mits Altair Floppy Disk



drive & controller · hardware documentation

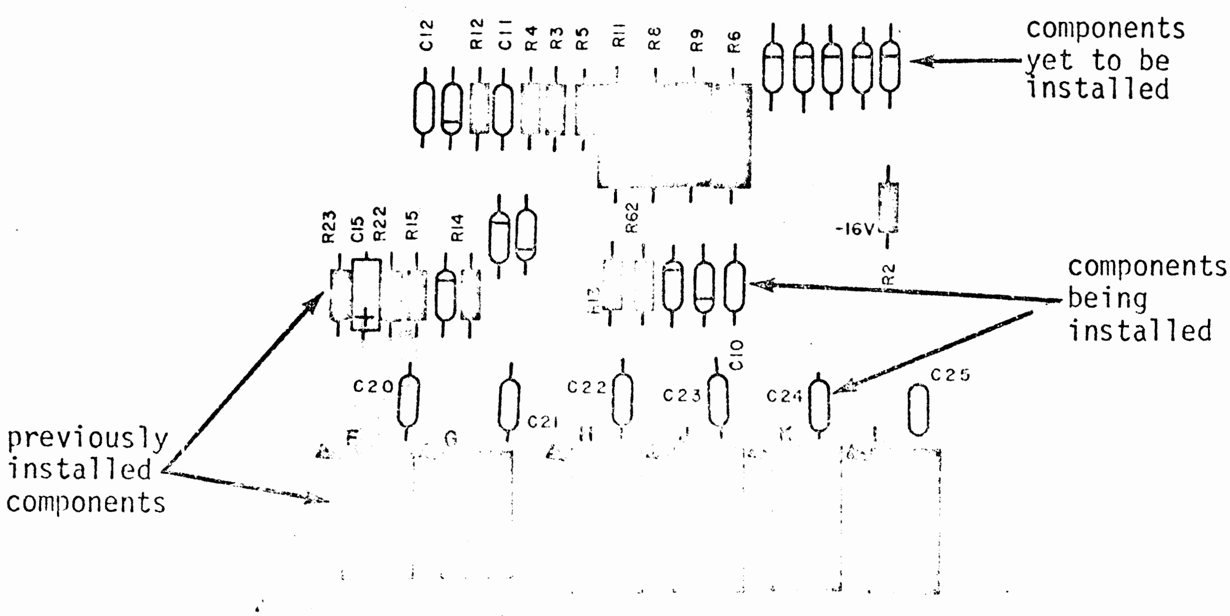
ASSEMBLY HINTS

Before beginning the construction of your unit, it is important that you read the "MITS Kits Assembly Hints" booklet included with your kit. Pay particular attention to the section on soldering, because most problems in the Altair occur as the result of poor soldering. It is essential that you use the correct type of soldering iron. A 25-30 watt iron with a chisel tip (such as an Ungar 776 with a 7155 tip) is recommended in the assembly hints booklet.

Some important warnings are also included in the hints booklet. Read them carefully before you begin work on your unit -- failure to heed these warnings could cause you to void your warranty.

Check the contents of your kit against the enclosed parts list to make sure you have all the required components, hardware and parts. The components are in plastic envelopes; do not open them until you need the components for an assembly step. You will need the tools called for in the "Kits Assembly Hints" booklet.

As you construct your kit, follow the instructions in the order they are presented in the assembly manual. Always complete each section before going on to the next. Two organizational aids are provided throughout the manual to assist you: 1) Boxed-off parts identification lists, with spaces provided to check off the components as they are installed; 2) Reproductions of the silk screens showing a) previously installed components, b) components being installed and c) components yet to be installed. (see below)



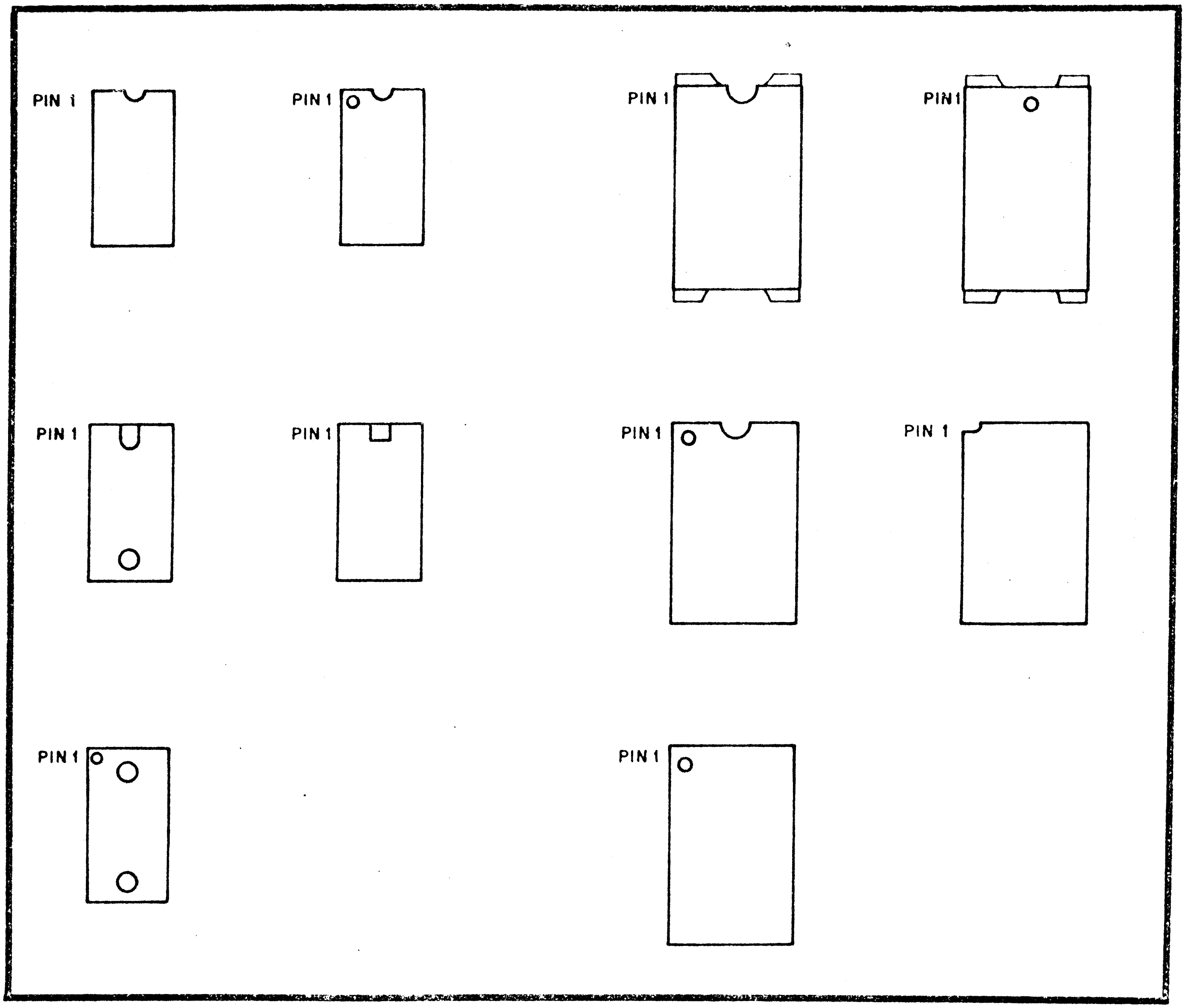
COMPONENT INSTALLATION METHODS

This section of you manual describes the proper procedures for installing various types of components in your kit.

Read these instructions over very carefully and refer back to them whenever necessary. Failure to properly install components may cause permanent damage to the component or the rest of the unit; it will definitely void your warranty.

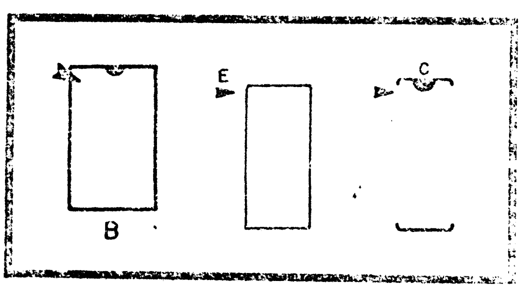
More specific instructions, or procedures of a less general nature, will be included within the assembly text itself.

Under no circumstances should you procede with an assembly step without fully understanding the procedures involved. A little patience at this stage will save a great deal of time and potential "headaches" later.



INTEGRATED CIRCUITS (IC's) CAN COME WITH ANY ONE OF, OR A COMBINATION OF, SEVERAL DIFFERENT MARKINGS. THESE MARKINGS ARE VERY IMPORTANT IN DETERMINING THE CORRECT ORIENTATION FOR THE IC's WHEN THEY ARE PLACED ON THE PRINTED CIRCUIT BOARDS. REFER TO THE ABOVE DRAWING TO LOCATE PIN 1 OF THE IC's, THEN USE THIS INFORMATION IN CONJUNCTION WITH THE INFORMATION BELOW TO PROPERLY ORIENT EACH IC FOR INSTALLATION.

WARNING: INCORRECTLY ORIENTED IC's MAY CAUSE PERMANENT DAMAGE!



THE DRAWING ON THE LEFT INDICATES VARIOUS METHODS USED TO SHOW THE POSITION OF IC's ON THE PRINTED CIRCUIT BOARDS. THESE ARE SILK-SCREENED DIRECTLY ON THE BOARD. THE ARROWHEAD INDICATES THE POSITION FOR PIN 1 WHEN THE IC IS INSTALLED.

IC Installation

All ICs must be oriented so that the notched end is toward the end with the arrowhead printed on the PC board. Pin 1 of the IC should correspond with the pad marked with the arrowhead. If the IC does not have a notch on one end, refer to the chart on the preceeding page for the identification of Pin 1.

To prepare ICs for installation:
All ICs are damaged easily and should be handled carefully — especially static-sensitive MOS ICs. Always try to hold the IC by the ends, touching the pins as little as possible.

When you remove the IC from its holder, CAREFULLY straighten any bent pins using needle-nose pliers. All pins should be evenly spaced and should be aligned in a straight line, perpendicular to the body of the IC itself.

1. Orient the IC so that Pin 1 coincides with the arrowhead on the PC board.

2. Align the pins on one side of the IC so that just the tips are inserted into the proper holes on the board.

3. Lower the other side of the IC into place. If the pins don't go into their holes right away, rock the IC back, exerting a little inward pressure, and try again. Be patient. The tip of a small screwdriver may be used to help guide the pins into place. When the tips of all the pins have been started into their holes, push the IC into the board the rest of the way.

4. Tape the IC into place on the board with a piece of masking tape.

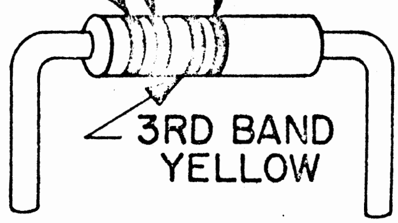
5. Turn the board over and solder each pin to the foil pattern on the back side of the board. Be sure to solder each pin and be careful not to leave any solder bridges.

6. Turn the board over again and remove the piece of masking tape.

Resistor Installation

Resistors have four (or possibly five) color-coded bands as represented in the chart below. The fourth band is gold or silver and indicates the tolerance. NOTE: In assembling a MITS kit, you need only be concerned with the three bands of color to the one side of the gold or silver (tolerance) band. These three bands denote the resistor's value in ohms. The first two bands correspond to the first two digits of the resistor's value and the third band represents a multiplier.

For example: a resistor with red, violet, yellow and silver bands has a value of 270,000 ohms and a tolerance of 10%. By looking at the chart below, you see that red is 2 and violet 7. By multiplying 27 by the yellow multiplier band (10,000), you find you have a 270,000 ohm (270K) resistor. The silver band denotes the 10% tolerance. Use this process to chose the correct resistor called for in the manual.



RESISTOR COLOR CODES

COLOR	BANDS 1&2	3rd BAND (Multiplier)
Black	0	1
Brown	1	10
Red	2	10^2
Orange	3	10^3
Yellow	4	10^4
Green	5	10^5
Blue	6	10^6
Violet	7	10^7
Gray	8	10^8
White	9	10^9

Use the following procedure to install the resistors onto the boards. Make sure the colored bands on each resistor match the colors called for in the list of Resistor Values and Color Codes given for each board.

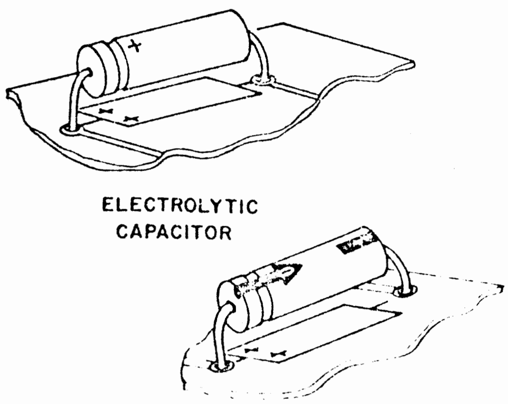
1. Using needle-nose pliers, bend the leads of the resistor at right angles to match their respective holes on the PC board.
2. Install the resistor into the correct holes on the silk-screened side of the PC board.
3. Holding the resistor in place with one hand, turn the board over and bend the two leads slightly outward.
4. Solder the leads to the foil pattern on the back side of the board; then clip off any excess lead lengths.

Capacitor Installation

A. Electrolytic and Tantalum Capacitors

<u>Polarity requirements must be noted on the electrolytic capacitors and the tantalum capacitor</u> before they are installed.

The electrolytic capacitors contained in your kit may have one or possibly two of three types of polarity markings. To determine the correct orientation, look for the following.



ELECTROLYTIC CAPACITOR

One type will have plus (+) signs on the positive end; another will have a band or a groove around the positive side in addition to the plus signs. The third type will have an arrow on it; in the tip of the arrow there is a negative (-) sign and the capacitor must be oriented so the arrow points to the negative polarity side.

The tantalum capacitor is metallic in appearance and smaller than the electrolytic capacitors. Its positive end has a plus sign on it or a red dot.

Refer to the chart included for each board for correct Capacitor Values and install the electrolytic capacitors and tantalum capacitors using the following procedure.

1. Bend the two leads of the capacitor at right angles to match their respective holes on the board. Insert the capacitor into the holes on the silk-screened side of the board. Be sure to align the positive polarity side with the "+" signs printed on the board.

2. Holding the capacitor in place, turn the board over and bend the two leads slightly outward. Solder the leads to the foil pattern and clip off any excess lead lengths.

B. Ceramic Disk Capacitors

Refer to the chart included for each board for correct Capacitor Values, and install the ceramic disk capacitors using the following procedure.

1. Choose the correct value capacitor and straighten the two leads as necessary to fit their respective holes on the PC board.

2. Insert the capacitor into the correct holes from the silk-screened side of the board. Push the capacitor down until the ceramic insulation almost touches the foil pattern.

3. Holding the capacitor in place, turn the board over and bend the two leads slightly outward.

4. Solder the two leads to the foil pattern on the back side of the board; then clip off any excess lead lengths.

Transistor Installation

To install transistors, use the following instructions.

NOTE: Always check the part number of each transistor before you install it. (See listing of Transistor Part Numbers for each board.) Some transistors look identical but differ in electrical characteristics, according to part number. If you have received substitute part numbers for the transistors in you kit, check the Transistor Identification Chart which follows these instructions to be sure you make the correct substitutions.

NOTE: Always make sure the transistor is oriented so that the emitter lead is installed in the hole on the PC board labeled with an "E." To determine which lead is the emitter lead, refer to the Transistor Identification Chart.

1. After the correct transistor has been selected and the leads have been properly oriented, insert the transistor into the holes on the silk-screened side of the board.

2. Holding the transistor in place, turn the board over and bend the three leads slightly outward.

3. Solder the leads to the foil pattern on the back side of the board; then clip off any excess lead lengths.

Diode Installation

NOTE: Diodes are marked with a band on one end indicating the cathode end. Each diode must be installed so that the end with the band is oriented towards the band printed on the PC board. Failure to orient the diodes correctly may result in permanent damage to your unit.

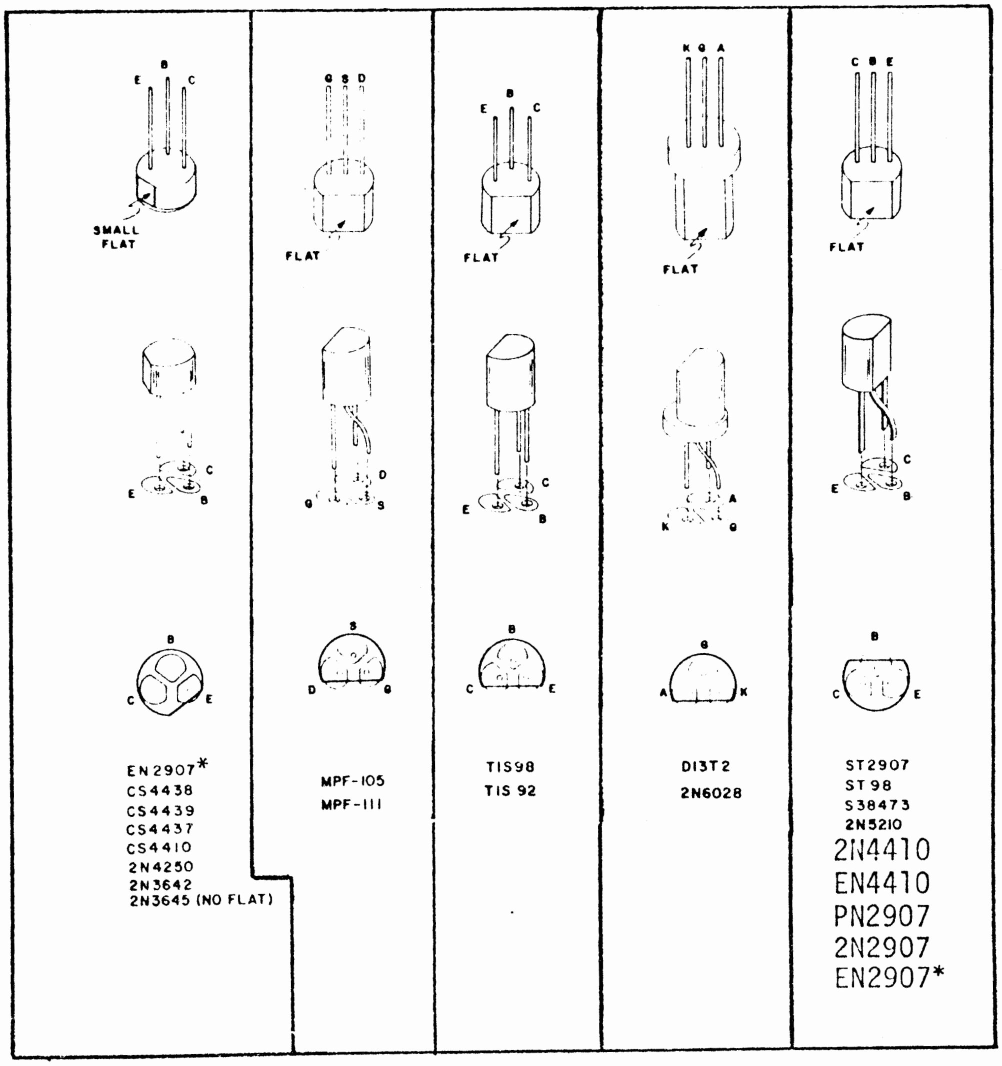
Use the following procedure to install diodes onto the board. Refer to the list of Diode Part Numbers included for each board to make sure you install the correct diode each time.

1. Bend the leads of the diode at right angles to match their respective holes on the board.

2. Insert the diode into the correct holes on the silk screen, making sure the cathode end is properly oriented. Turn the board over and bend the leads slightly outward.

3. Solder the two leads to the foil pattern on the back side of the board; then clip off any excess lead lengths.

TRANSISTOR IDENTIFICATION CHART



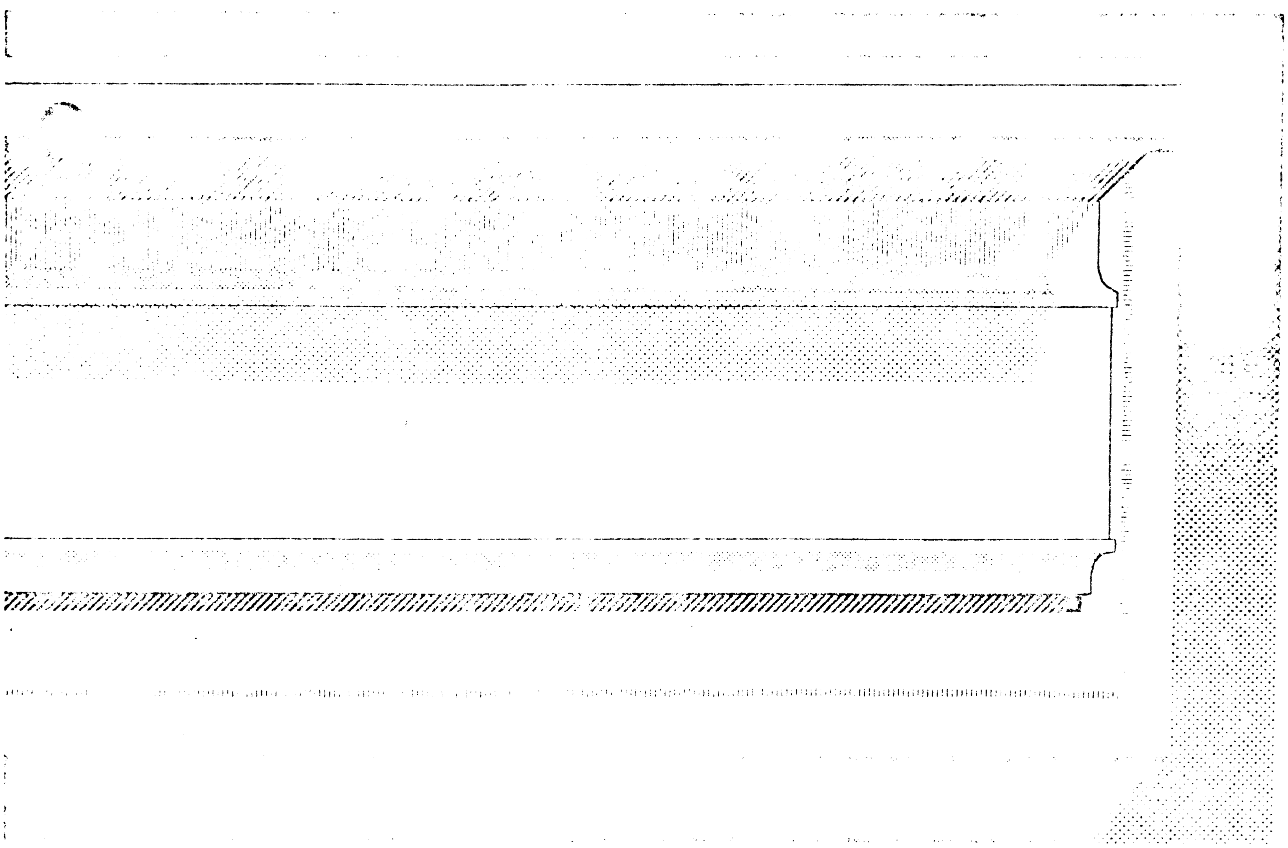
IN THE ILLUSTRATION ABOVE THE OUTLINE OF EACH TYPE OF TRANSISTOR IS SHOWN OVER THE PADS ON THE CIRCUIT BOARD WITH THE CORRECT DESIGNATION FOR EACH OF THE THREE LEADS. USE THIS INFORMATION TOGETHER WITH THE INFORMATION IN THE ASSEMBLY MANUAL FOR THE CORRECT ORIENTATION OF THE TRANSISTORS AS YOU INSTALL THEM.

THE FOLLOWING IS A LIST OF POSSIBLE SUBSTITUTIONS: IF ANY OTHERS ARE USED YOU WILL RISK DAMAGING YOUR UNIT:

2N4410 = EN4410 = CS4410 = CS4437, CS4438, TIS98, ST98, S38473 (NPN)
EN2907 = 2N2907 = PN2907 = ST2907, CS4439 (PNP)

WHEN MAKING SUBSTITUTIONS, REFER TO THE ILLUSTRATION TO DETERMINE THE CORRECT ORIENTATION FOR THE THREE LEADS.

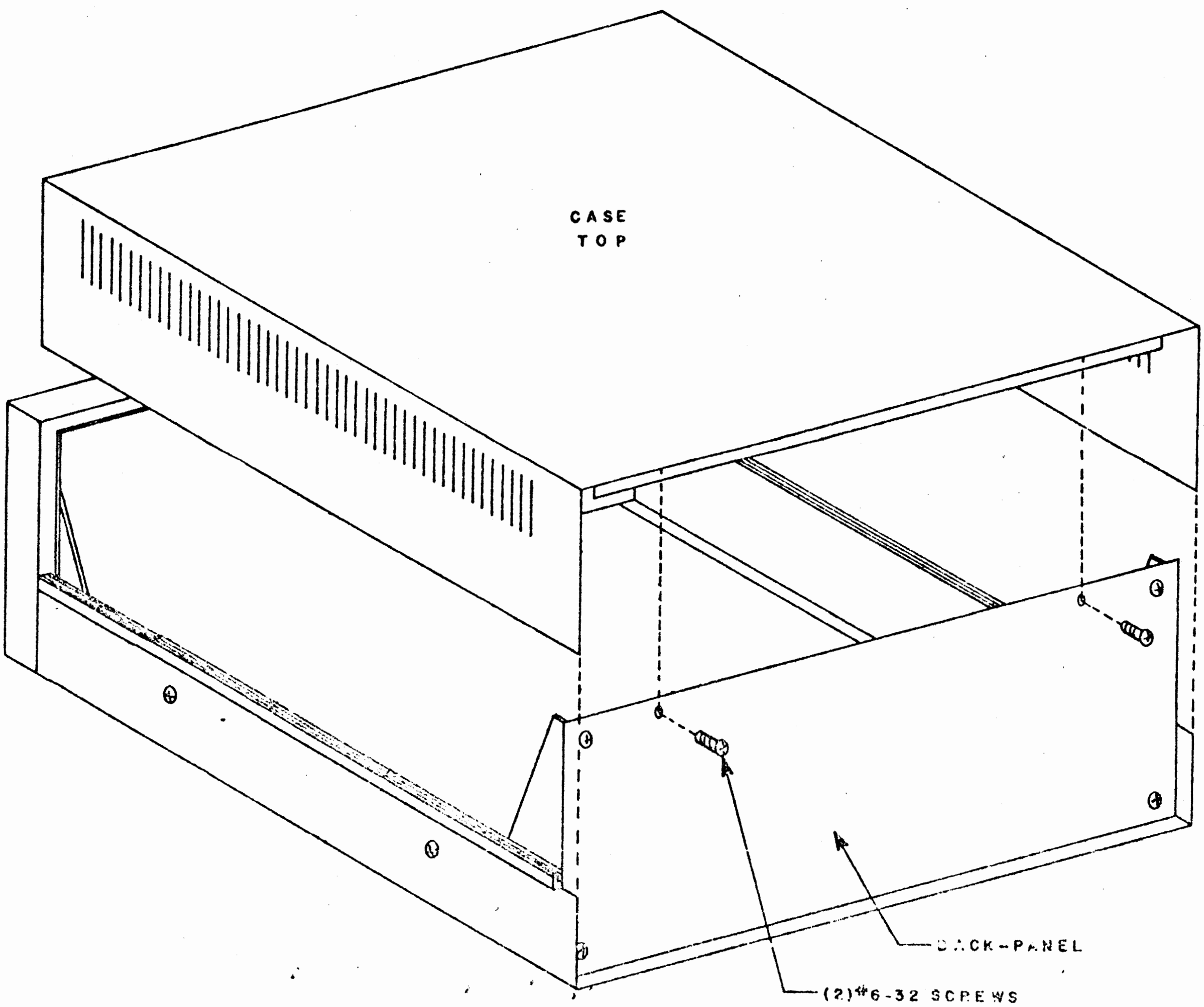
*Configuration of the leads on EN2907 may vary.



CASE TOP REMOVAL

Remove the top from the Disk Drive case by withdrawing the two screws indicated in the drawing below. Slide the case top backwards, lifting the back slightly, to remove it entirely from the chassis.

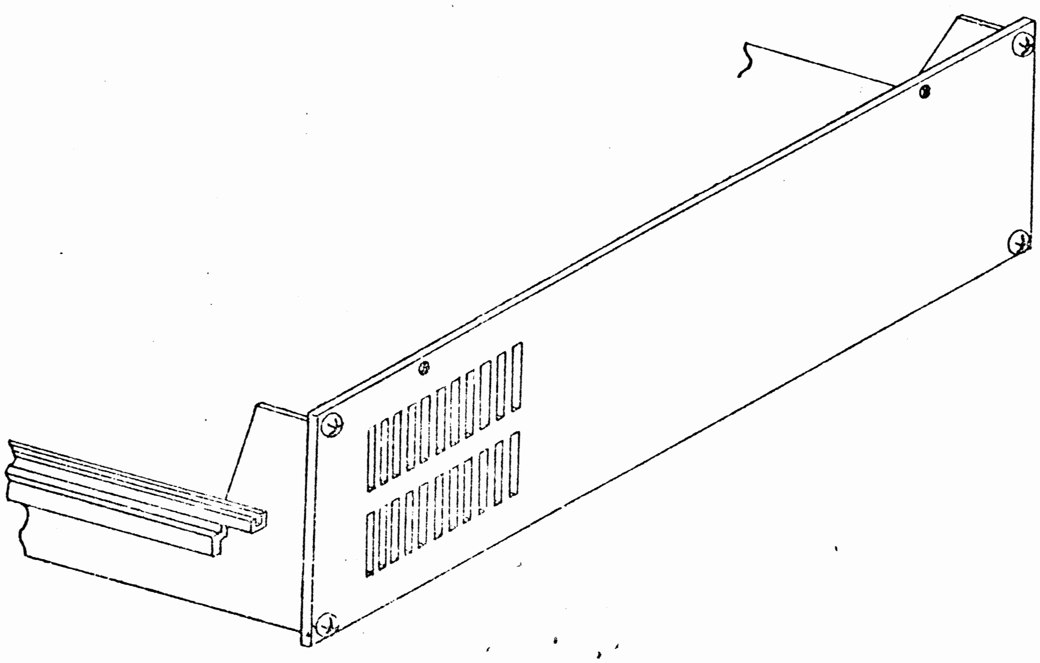
Also remove the 4 screws in the side of the case bottom, and remove the entire chassis assembly.



DISK DRIVE BACK PANEL ASSEMBLY

Remove the back panel from the case by withdrawing each of the four screws in the corners of the panel. These four screws are shown inserted in the drawing below.

Save these four screws for remounting the back panel later in the assembly procedure.

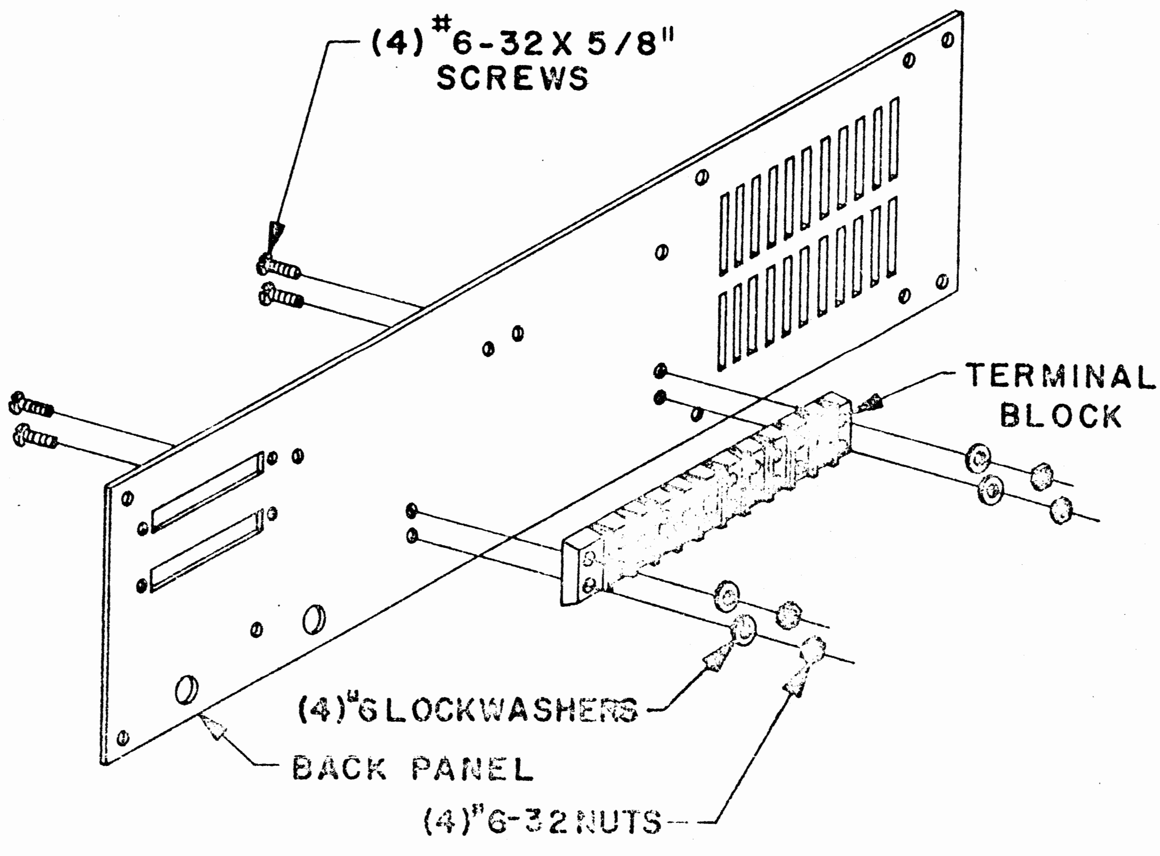


Terminal Block Installation

Mount the terminal block to the back panel as shown in the drawing below. Use the screw sizes and other hardware indicated in the drawing.

NOTE: Be sure that the back panel is oriented as shown; be careful not to mount the terminal block on the wrong side of the panel.

Tighten all four screws firmly into place.



Transformer Installation

There are two transformers included in this kit. The larger of the two will be refered to as T1, the smaller as T2.

Wire Preparation

Before mounting these transformers, the wires must be cut to the proper length and screw-mount crimp terminals attached to each of them. There are also three wires which will not be used at all, and will be cut off at the transformer coil.

Refering to the drawing on the opposite page, cut the wires on transformers T1 and T2 to the lengths indicated. The three unused wires should be cut off at the point where they enter the transformer coil itself.

Next, as indicated in the bottom of the drawing, strip exactly 1/2" of insulation from each of the eleven wires and bend the exposed portion in half to 1/4".

There are several screw-mount crimp terminals included with this kit. These have a slot in one end and an insulated portion on the other end (usually red) for attaching wires. One of these crimp terminals must be attached to each of the eleven transformer wires.

Insert one of the wires into one of the terminals as shown in the drawing. Push the wire in as far as it will go without distorting it or pushing it all the way through.

The wire should then be permanently connected to the terminal by either soldering it in place or crimping. To crimp the terminal use a crimping tool, if available, or else flatten the insulated portion of the terminal as tightly as possible using pliers.

Prepare each of the eleven transformer wires in the above manner.

Mounting

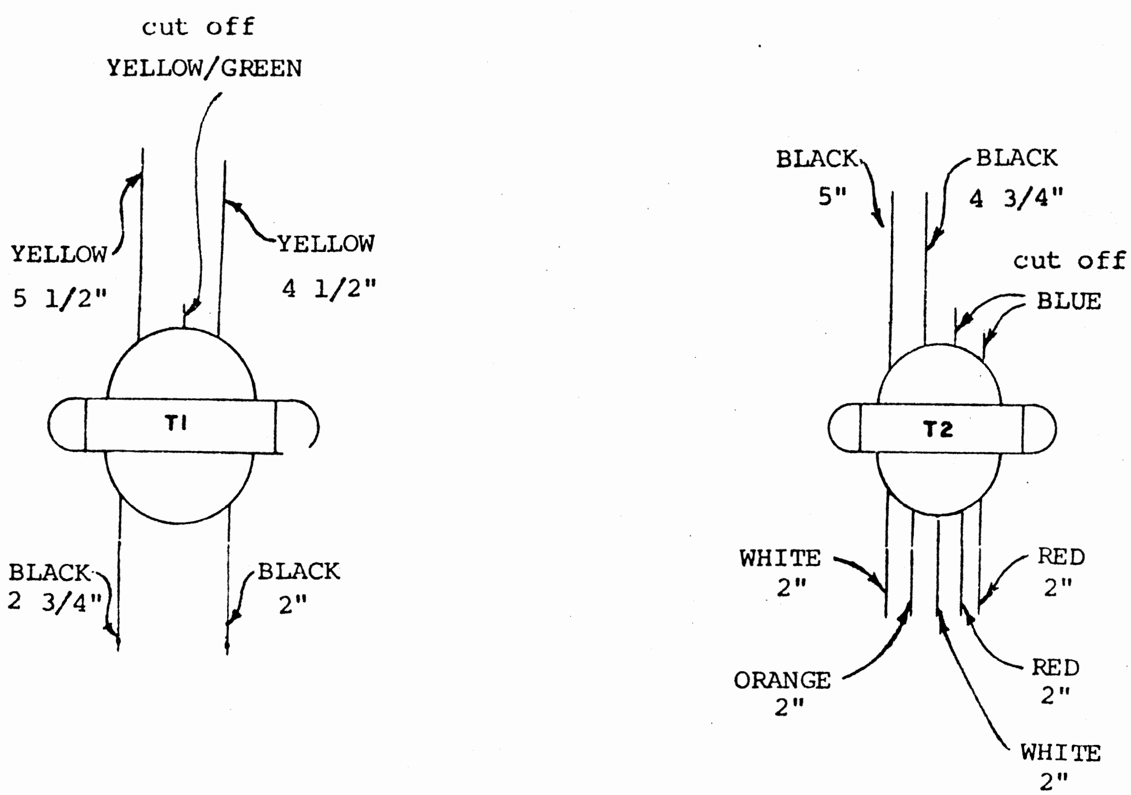
Refering to the drawings following the "Transformer Wire Preparation" drawing, mount transformers T1 & T2 to the back panel.

NOTE: For proper orientation, transformer T1 should have the two yellow wires towards the top of the panel (with reference to the drawings), and T2 should have the two black wires towards the top of the panel.

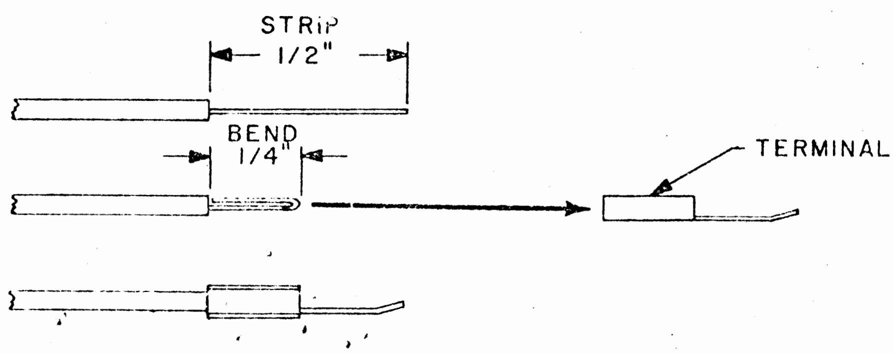
Be sure to install a terminal lug on transformer T1 as shown in the drawing. This is a solder type lug, and not the screw-mount type used for the transformer wires.

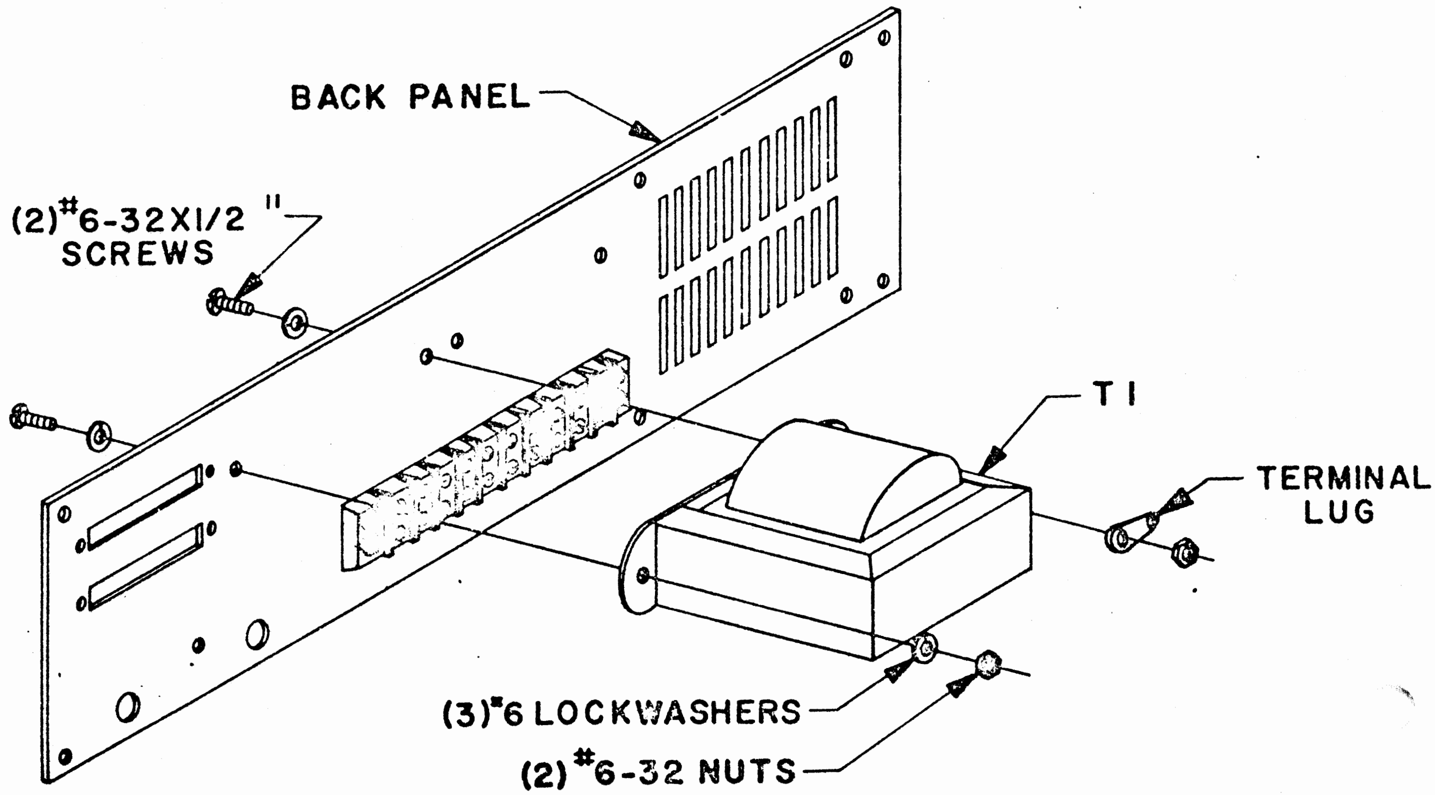
Use the hardware indicated in the drawings to mount the transformers and tighten the screws firmly into place.

NOTE: Save all wires that you cut off for later use.

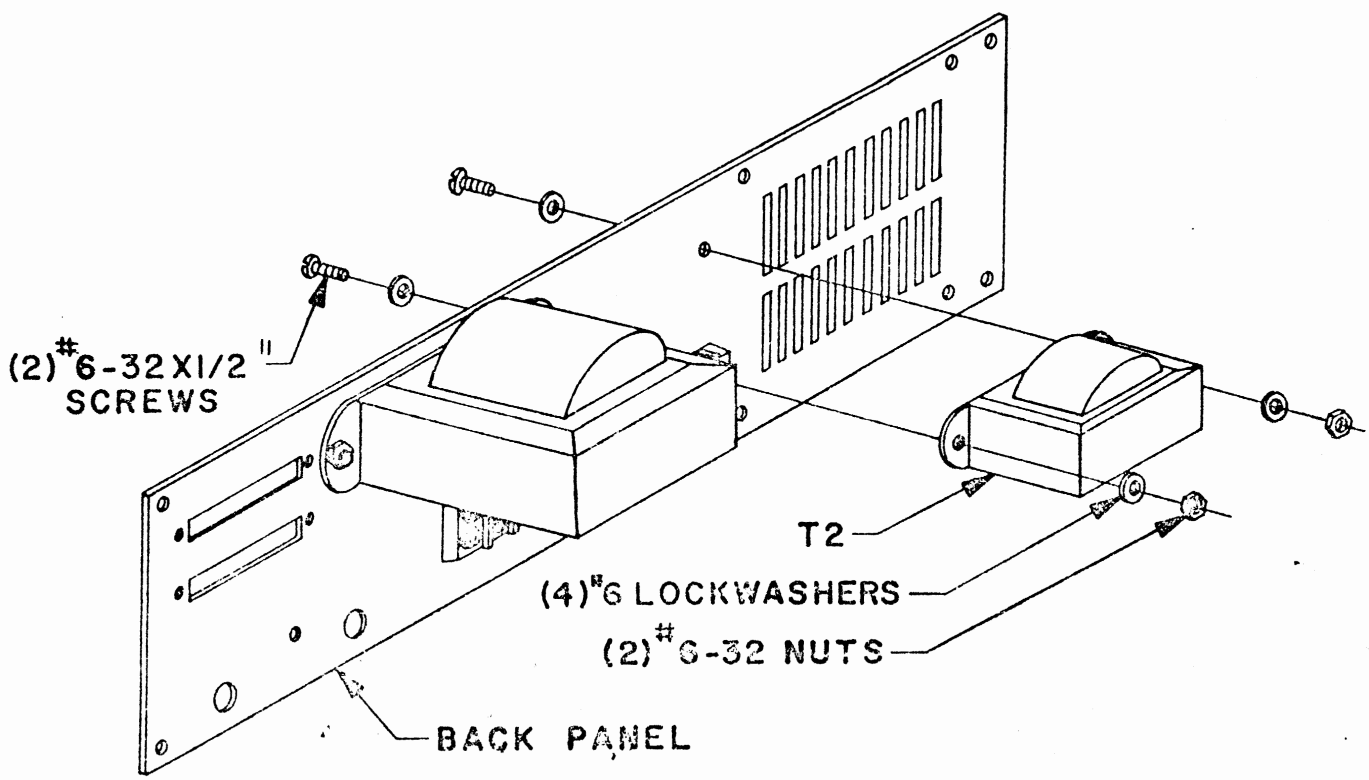


TRANSFORMER WIRE PREPARATION





BACK PANEL
(2)#6-32X1/2"
SCREWS
T1
TERMINAL
LUG
(3)#6 LOCKWASHERS
(2)#6-32 NUTS

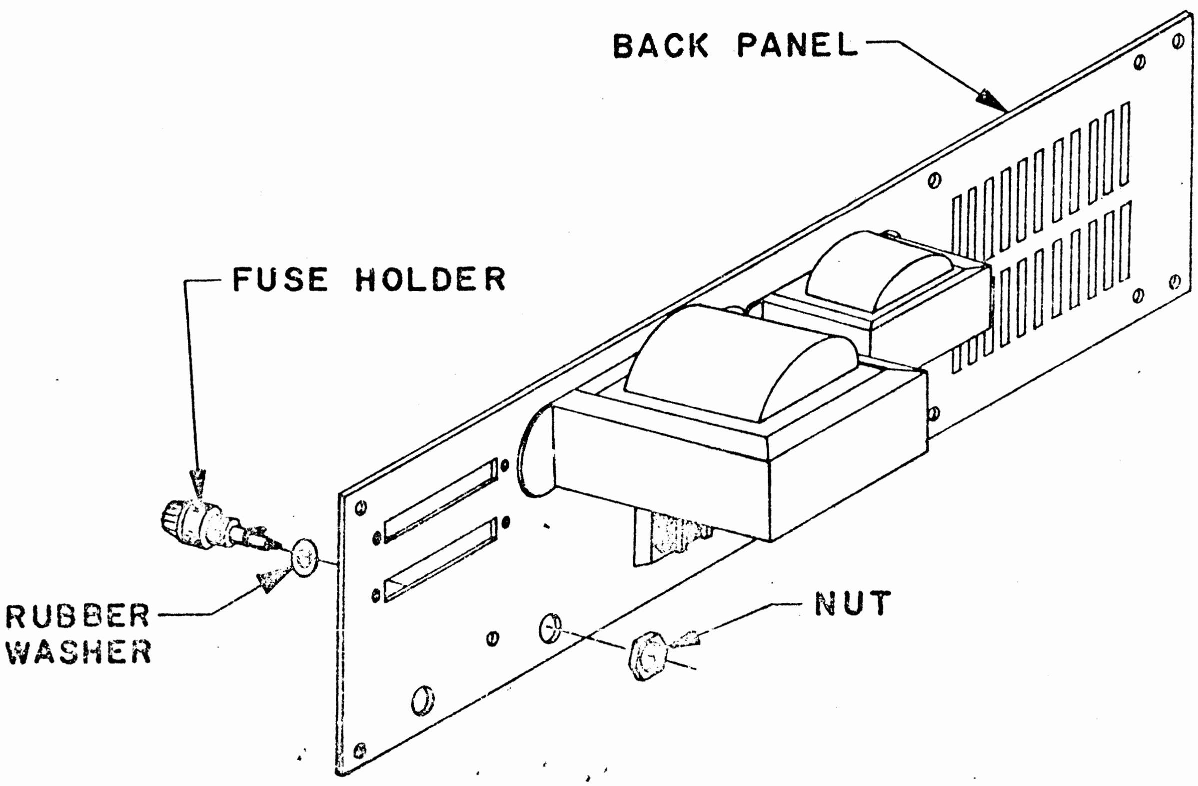


(2)#6-32X1/2"
SCREWS
T2
(4)#6 LOCKWASHERS
(2)#6-32 NUTS
BACK PANEL

Fuse Holder Installation

Refering to the drawing below, mount the fuse holder to the back panel using the rubber washer and nut provided. Tighten it firmly into place.

Remove the cap and place the fuse provided with your kit into the holder, then replace the cap.

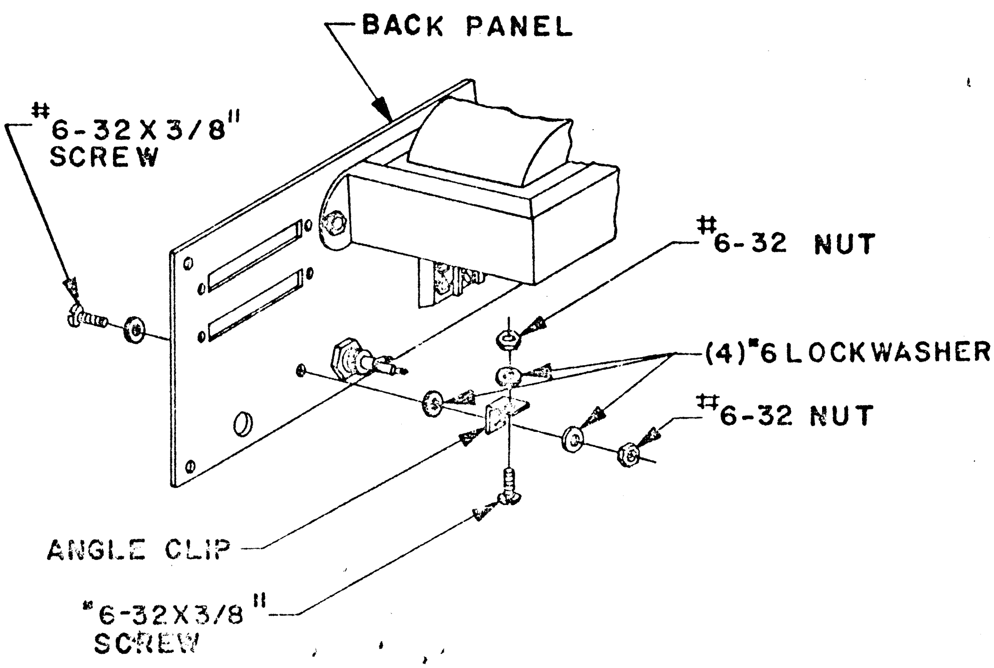


90° ANGLE CLIP INSTALLATION

The drawing below illustrates the hardware and orientation for mounting the 90° angle clip included with this kit.

NOTE: One side of the clip is slightly shorter than the other. The shorter side should be mounted against the back panel with the longer side extending at 90°.

Install the clip as shown below and tighten the screws firmly into place. Be sure that clip remains "square" with the panel when tightening the screws.



Fan Installation

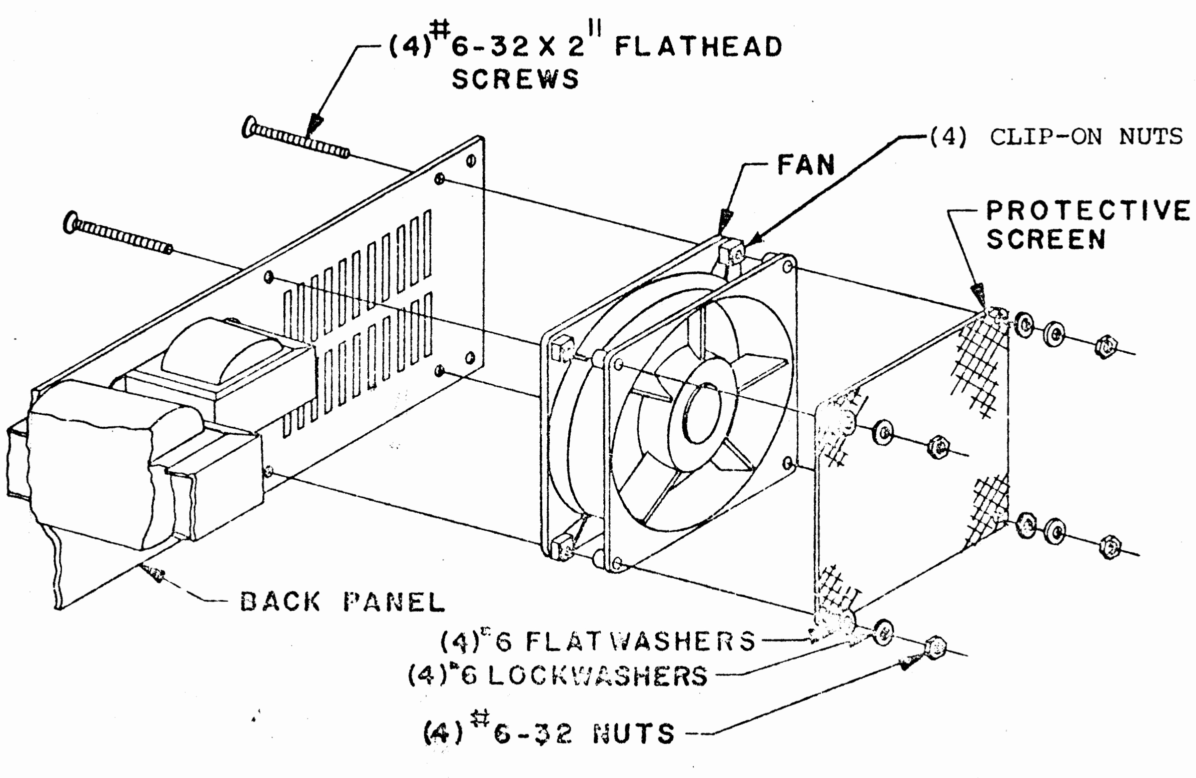
Before the cooling fan is installed onto the back panel, two lengths of wire must be prepared and connected to it.

There is some black wire included with the kit; cut two 6 1/2 inch lengths of this wire. Strip 1/2 inch of insulation from one end of each of the wires, and 1/4 inch of insulation from the other.

In the same manner as described on page   , attach a screw-mount crimp terminal to the 1/2" stripped end of each of the two wires. Tin the 1/4" stripped ends of the wires by applying a thin coat of solder.

There are two terminals on the fan in one of the corners. Solder the ends of the two wires opposite the crimp terminals to the terminals on the fan.

Refering to the drawing below, mount the fan and screen to the back panel using the hardware indicated. For proper orientation, the terminals with the two wires attached should be towards the bottom on the side nearest the terminal block. The arrow printed on the fan to indicate airflow should be facing towards the screen. The screen itself has a bump on one side in each of the four corners. The side with the bumps should be towards the fan.



Power Cord Installation

There is a 3-wire power cord included with this kit which must be prepared as follows before installation.

1) Strip 4" of the cord casing from the wires by cutting a circle 4" from the end and pulling off the black insulation. Be careful not to cut into the insulation on any of the wires inside.

2) The green wire inside should already be at the correct length of 4 inches. Cut the white wire to 3 1/2 inches, and the black wire to 1 1/4 inches. Strip 1/4 inch of insulation from the ends of each of the three wires.

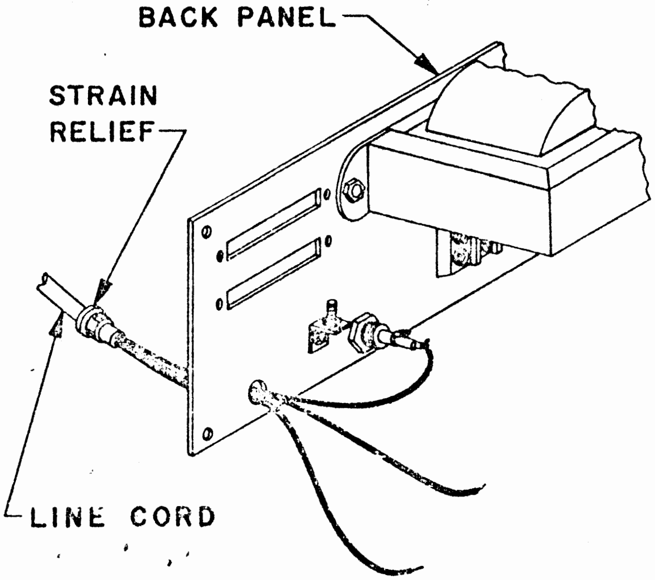
3) Tin the exposed 1/4" of the black wire by applying a thin coat of solder.

4) Solder or crimp screw-mount crimp terminals to the white and green wires.

Place the strain relief, included with the kit, over the power cord. Be sure that the larger diameter end of the relief is towards the male plug end of the cord.

Be sure that there is approximately three inches of the cord's black insulation case extending beyond the strain relief*, then snap it into place on the back panel as shown below.

* The black wire should reach to the center of the fuse holder when the cord & strain relief are in place.



Wire Preparation

Using the wire supplied with this kit, and the length of yellow/green wire cut from transformer T1, prepare the power supply interconnect wires according to the following instructions.

To avoid confusion, it would be best to prepare these wires one at a time.

The list on the right indicates the color of each wire, the length to which it should be cut, and a reference "tag".

Use the following steps to prepare each wire:

1) Cut the specified color wire to the length indicated.

2) Strip 1/2 inch of insulation from one end and 1/4 inch from the other.

3) Tin the wire exposed 1/4 inch by applying a thin coat of solder.

4) According to the instructions on page , connect a screw-mount crimp terminal to the 1/2 inch stripped end.

5) Approximately 5 inches from the 1/4 inch tinned end of the wire label it, using masking tape, with the reference tag indicated.

An additional length of BLACK wire should be cut to 22 1/2 inches and 1/4 inch of insulation stripped from each end. Tin both ends by applying a thin coat of solder. Label this wire "FUSE".

Interconnect Wires

COLOR	LENGTH		TAG
Yellow/ Green*	2	inches	3
Black	22 3/4	"	3
Black	17 3/4	"	9
Black	17 1/2	"	10
Black	25	"	1
White	18	"	6
White	17 3/4	"	8
Orange	17 3/4	"	7
Orange	18 1/2	"	4
Orange	18 1/4	"	5

*From transformer T1, This wire need not be labeled.

Back Panel Wiring

The disk back panel assembly may now be completed by connecting all of the wires to their appropriate locations.

(See drawing page 23)

Three solder connections are necessary and should be made first. These include the black power cord wire, the yellow/green wire and the black 22 1/2 inch wire labeled "FUSE".

1) Solder the 1/4 inch tinned end of the yellow/green wire to the solder lug on transformer T1.

2) Solder the black power cord wire to the center terminal on the fuse holder.

3) Solder one end of the black "FUSE" wire to the other fuse holder terminal.

The remaining connections will be made to the terminal block.

The drawing (P.23) shows the proper orientation and connections for all of the wires on the back panel. The "tags" on the wires you prepared earlier refer to the numbers shown on the terminal block.

WARNING: The power supply is a critical part of any electronic system. Check the wiring here several times to be sure you have it correct. Be sure that each of the wires is in the proper location and that all of the screws on the terminal block are tight.

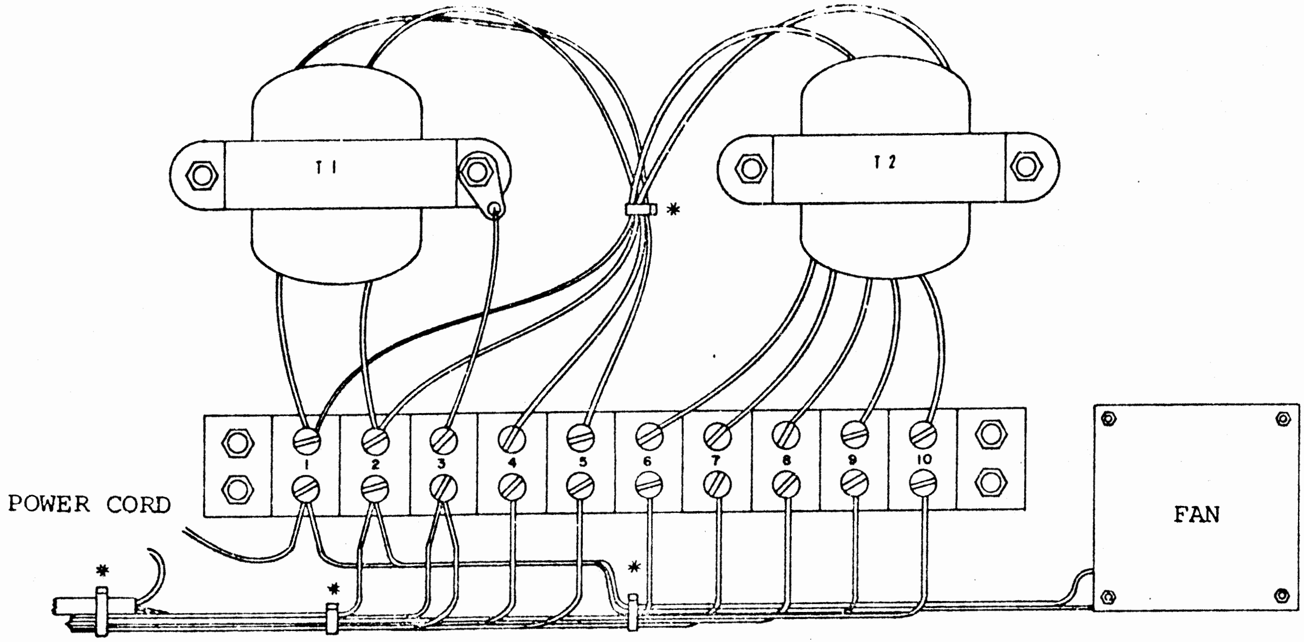
Use the drawing below for reference and connect all of the wires as indicated. Match the "tags" on the wires prepared earlier with the numbered positions on the terminal block. There should be a total of 25 crimp terminal connections made to the block.

NOTE: Where two terminals are to be connected to the same screw, place them "back to back". In this position they will fit flat together, and make a much more solid connection.

The ON-OFF SWitch may also be soldered in at this time. Use the free end of the black "FUSE" wire and the free end of the wire labeled "1" to connect to the switch terminals. There are three terminals on the switch. Use the center terminal and one to either side of it. (The switch position towards the side where the connections are made will be its OFF position.

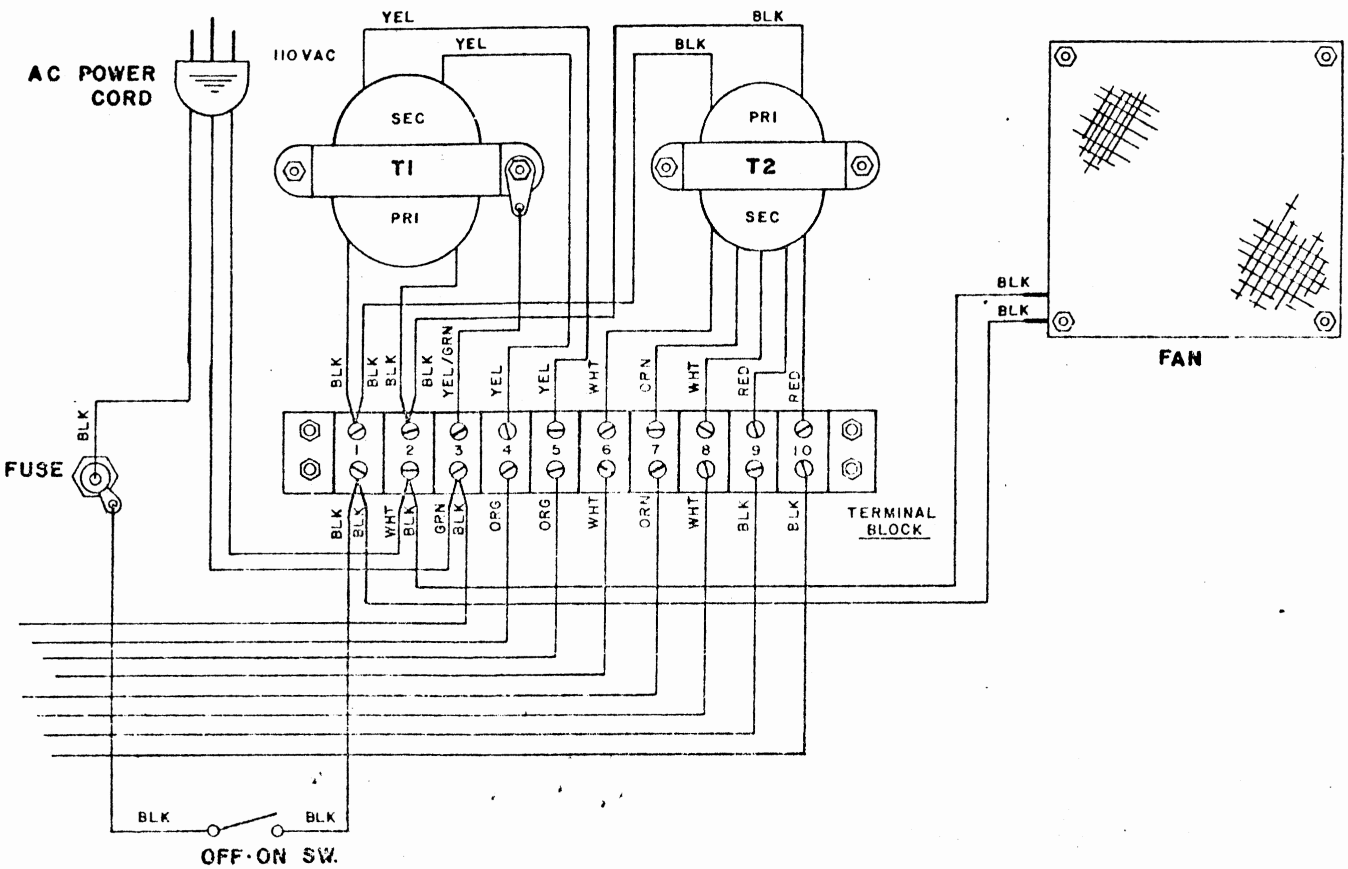
Install the 4 tie wraps in the positions shown in the top drawing on page 23.

WIRE ROUTING & TIE WRAPS



*TIE WRAPS (4)

BACK PANEL WIRING



DISK POWER SUPPLY BOARD ASSEMBLY

<u>NOTE</u>: Save all component leads clipped off during assembly until the entire unit is complete. Some of the leads will be used during the assembly process.

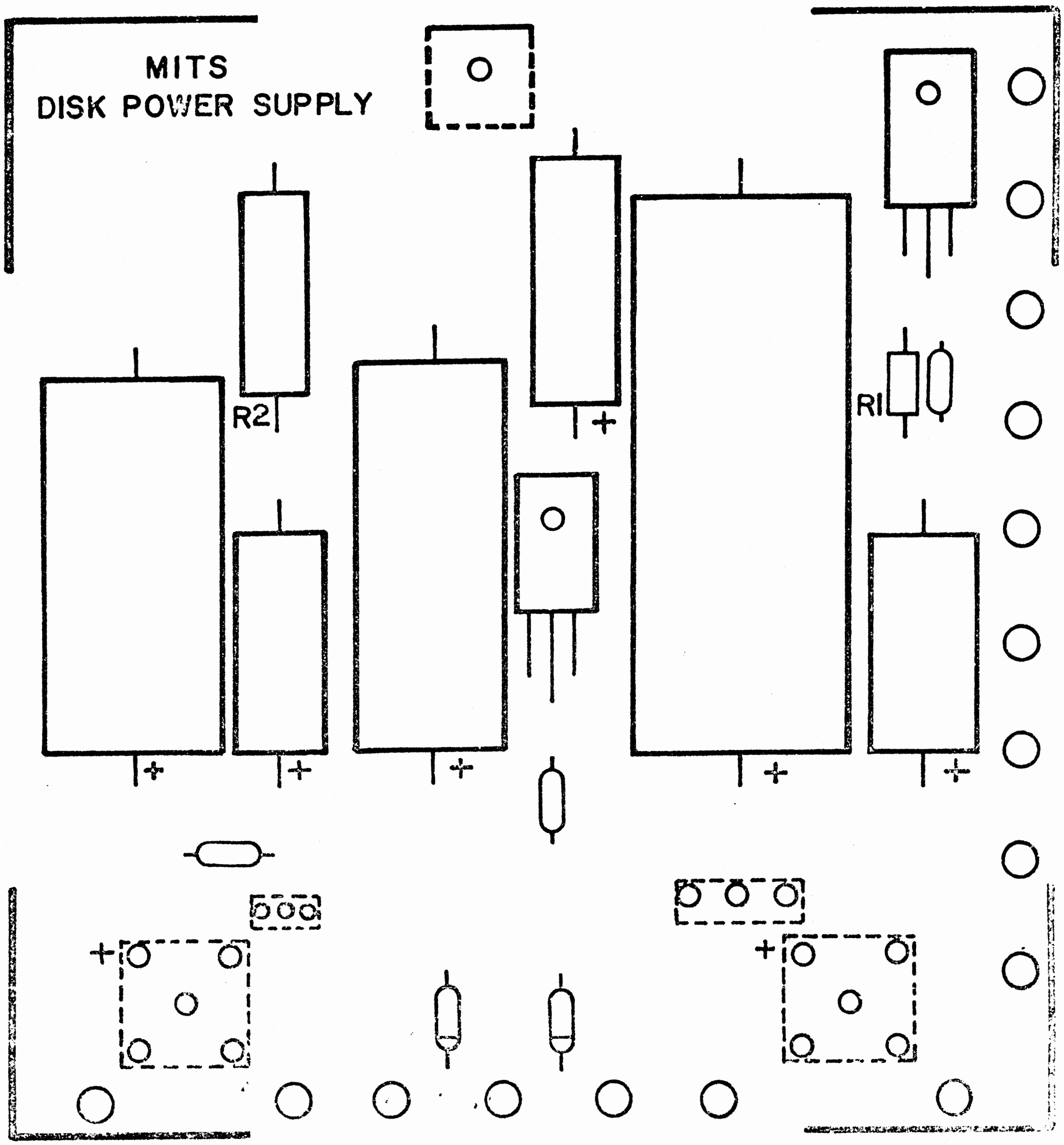
<u>RESISTOR INSTALLATION</u>

Install the following 2 resistors according to the instructions listed on page 5.

<u>RESISTOR VALUES AND COLOR CODES</u>

( ) R1 is 33 ohm (orange-orange-black) 1/2 W

( ) R2 is 7.5 ohm, 5 W (this may be color coded, violet-green-3rd band white or gold; or it may be a solid body color, with the value printed directly on the resistor itself.


MITS
DISK POWER SUPPLY
R2
R1

CAPACITOR INSTALLATION

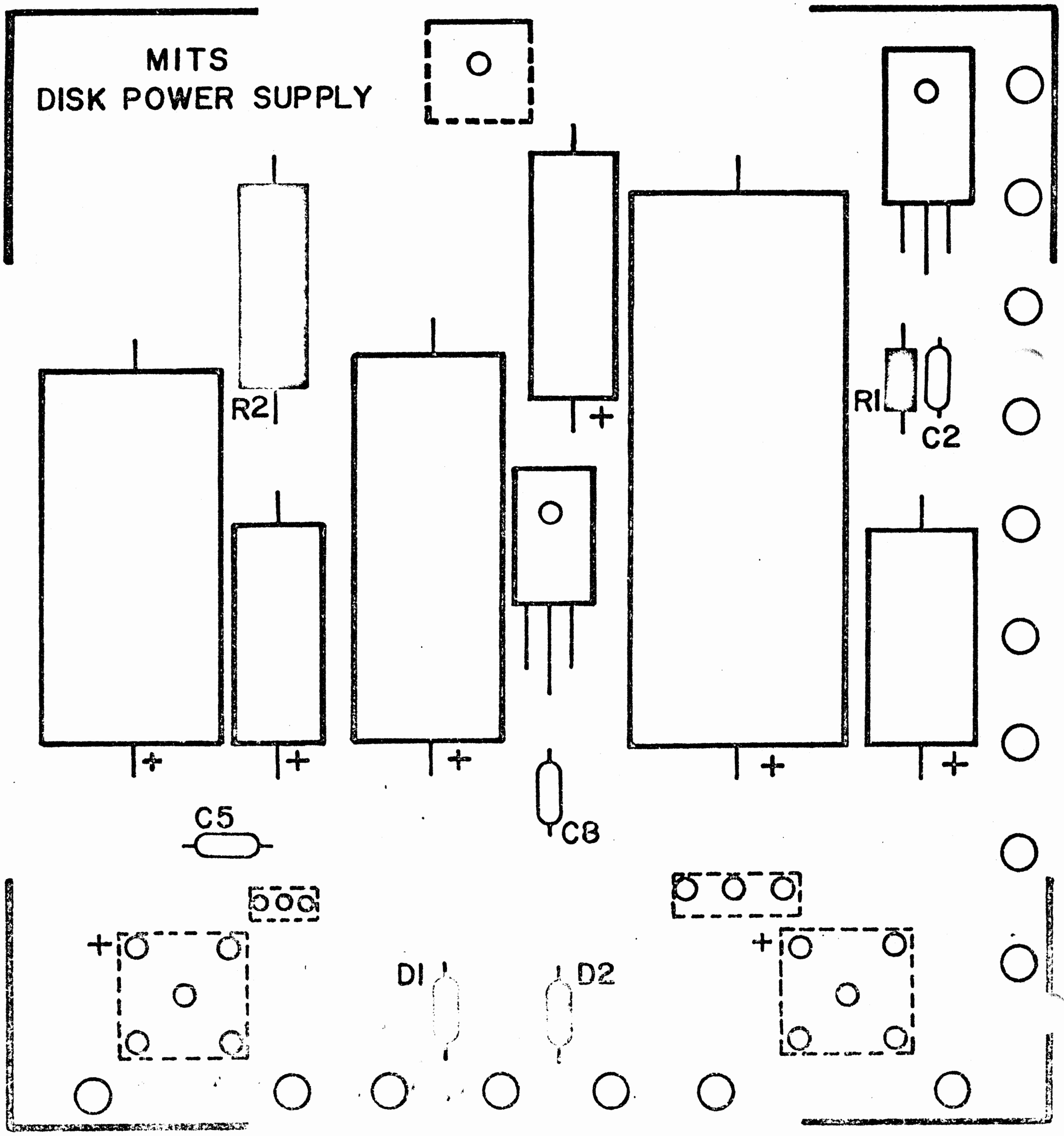
Install the following 3 ceramic disk capacitors according to the instructions on page 6 .

CAPACITOR VALUES

( ) C2 = .luf, 50V

( ) C5 = .luf, 50V

( ) C8 = .luf, 50V



CAPACITOR INSTALLATION

Install the following 6 electrolytic capacitors according to the instructions listed on page 6 .

CAPACITOR VALUES

( ) C1 = 2200uf, 50V

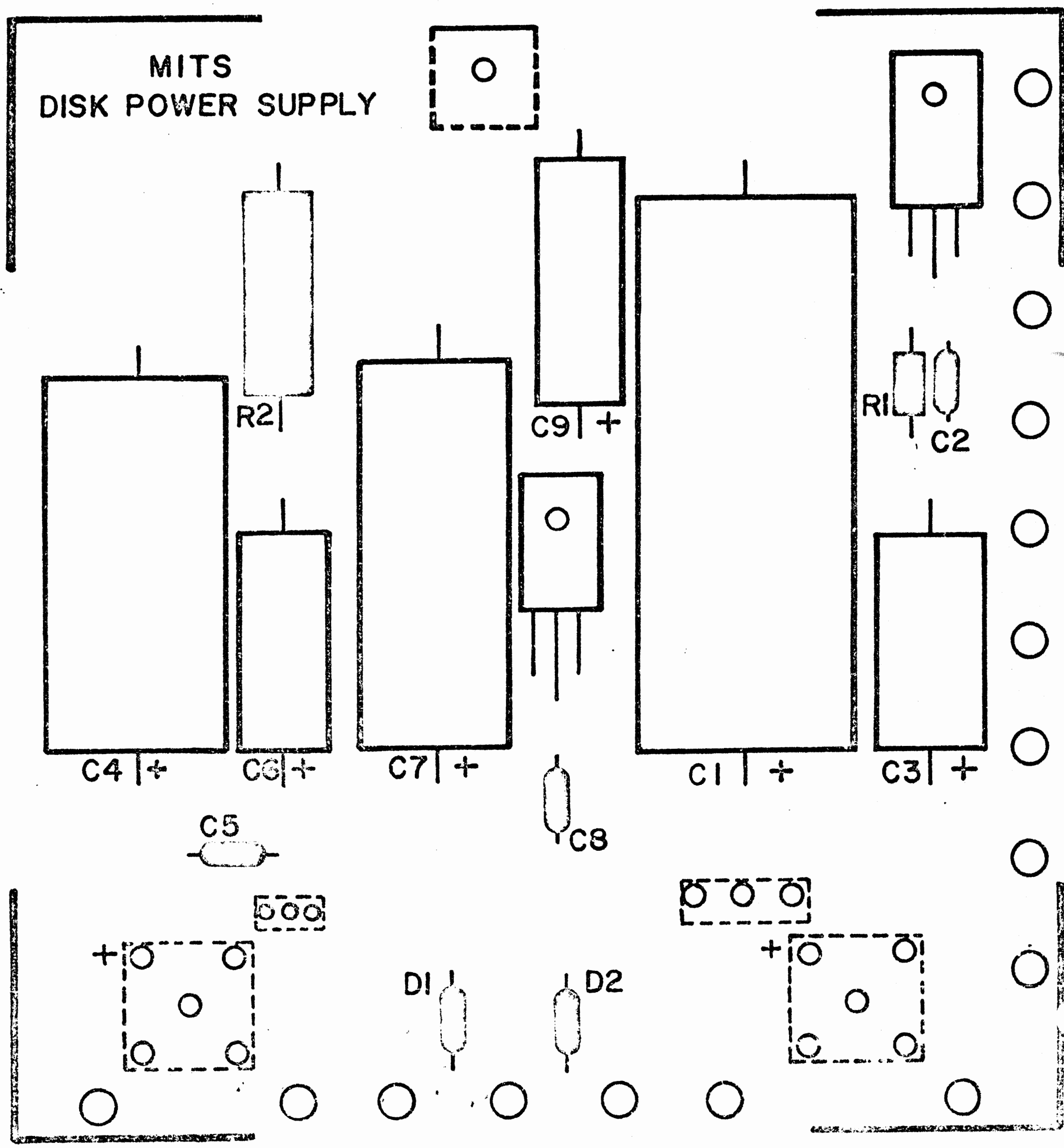
( ) C3 = 33uf, 50V

( ) C4 = 3300uf, 16V

( ) C6 = 33uf, 50V

( ) C7 = 1000uf, 25V

( ) C9 = 33uf, 50V

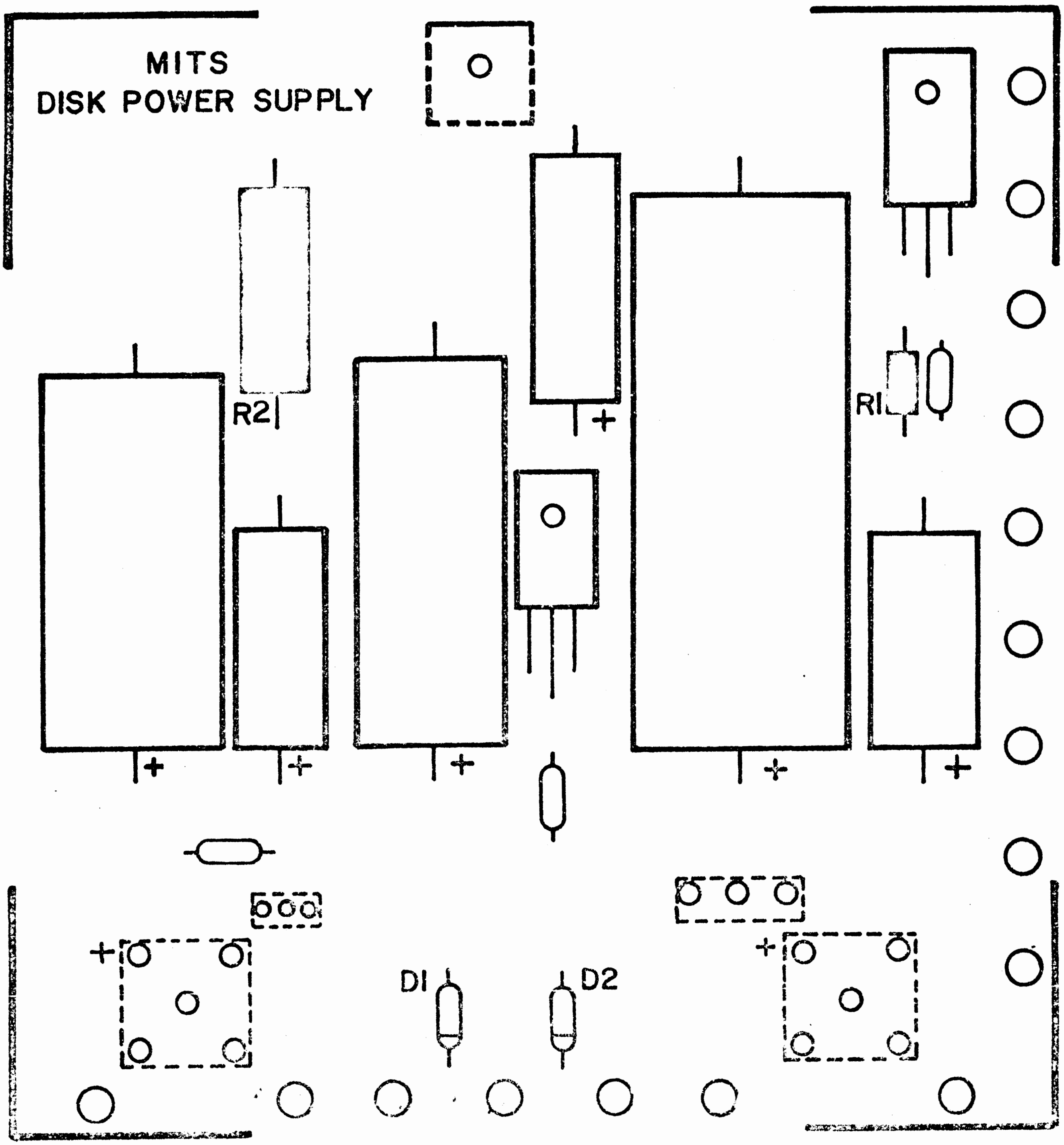

MITS
DISK POWER SUPPLY
R2
C9 +
R1
C2
C4 +
C6 +
C7 +
C1 +
C3 +
C5
C8
D1
D2
+
+

DIODE INSTALLATION

Install the following 2 diodes according to the instructions on page 7.

( ) D1 = 1N4004

( ) D2 = 1N4004



VOLTAGE REGULATOR INSTALLATION

There are 2 voltage regulators to be installed on the silk-screened side of the power supply board, X1 & X3.

These are to be installed according to the following procedure. (See drawing.)

(1) Set the regulator in place over the board so that the mounting hole in the regulator and the board align.

(2) Use a pencil to mark the point on each of the regulator's three leads directly over its corresponding hole in the board.

(3) Bend the three leads, using needle-nose pliers, at right angles from the printed side of the component.

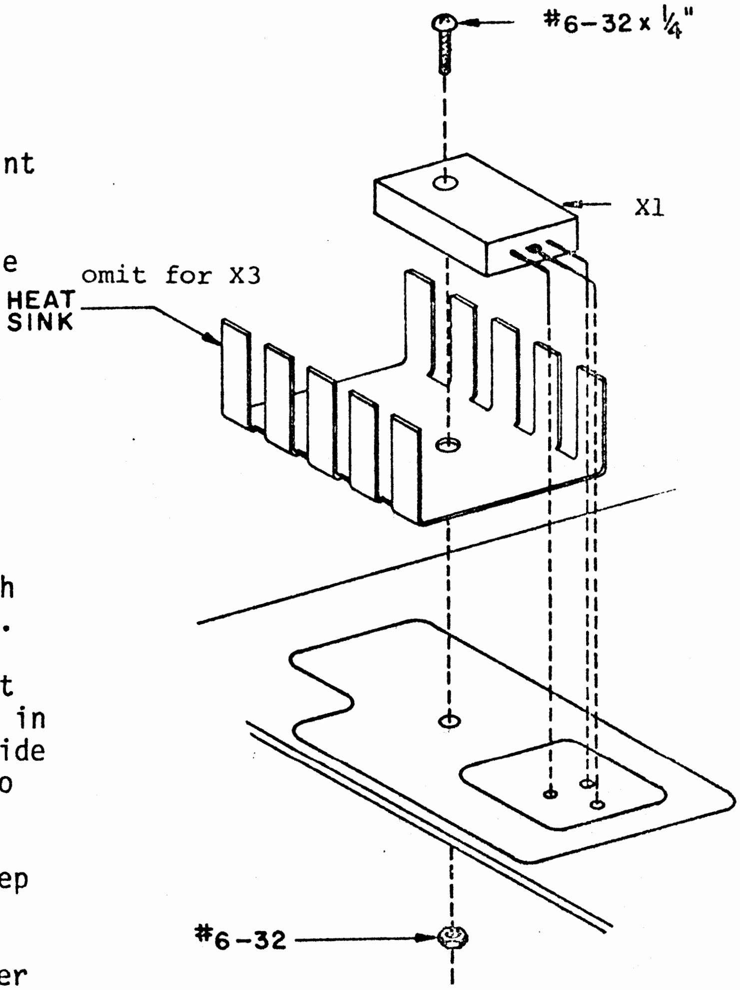
NOTE: Use heat-sink grease when installing this component. Apply the grease to all surfaces which come in contact with each other.

(4) Referring to the drawing, set the transistor and heat sink in place on the silk-screened side of the board. Secure them to the board using a #6-32 nut. Hold the transistor in place as you tighten the nut to keep from twisting the leads.

(5) Turn the board over and solder the three leads to the foil pattern on the back side of the board. Be sure not to leave any solder bridges.

(6) Clip off any excess lead lengths.

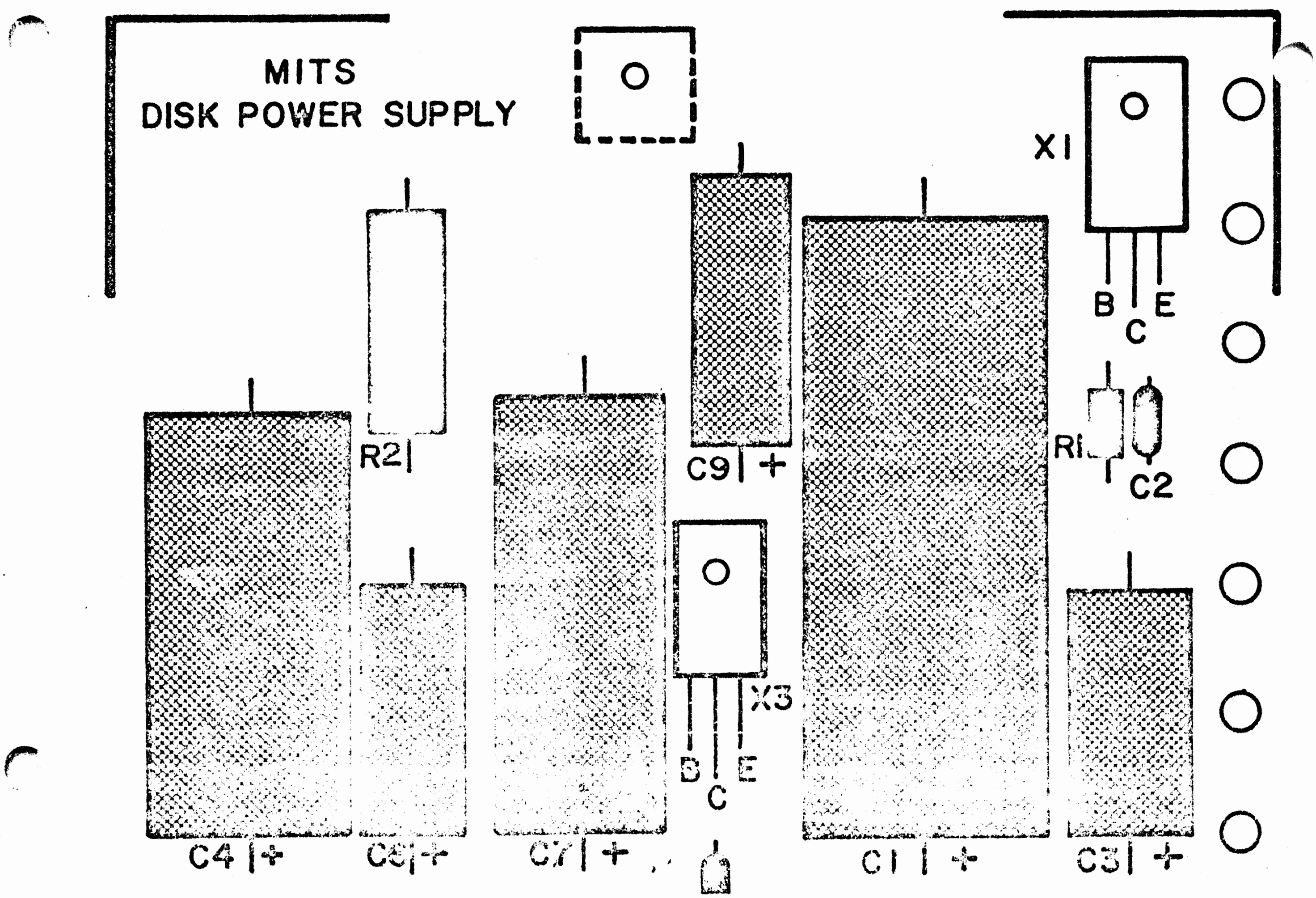
NOTE: For X1 the above procedure should be followed precisely. For X3 it is the same except that no heat-sink is to be installed.



VOLTAGE REGULATOR INSTALLATION

( ) X1 = 7805

( ) X3 = 7524



BRIDGE RECTIFIER INSTALLATION

There are two bridge rectifiers, BR1 & BR2, to be installed on the power supply board.

WARNING: Read the following instructions closely. Proper orientation of these two components is absolutely critical.

These two components are indicated on the silk-screen by broken lines. This is to indicate that they are to be mounted on the bottom (non-silk-screened) side of the board.

You will observe a "+" sign printed near one corner of the rectifier. The lead nearest this "+" sign is the positive lead of the rectifier. This lead must be inserted into the hole marked on the silk-screen with a "+" sign.

NOTE: There is also a "-" sign printed on the regulator. The lead nearest this sign is the negative lead of the rectifier, and should be diagonally opposite the "+" lead on the board.

BE ABSOLUTELY SURE THAT THE PROPER ORIENTATION IS USED WHEN INSTALLING THESE TWO COMPONENTS.

Install the rectifiers according to the following procedure:

(1) Insert the four leads of the BR1 rectifier into their respective holes from the non-silk-screened side of the board. Be sure the "+" lead of the rectifier is inserted in the hole labeled "+" on the silk-screened side of the board.

(2) Insert the BR2 rectifier in the same manner. Be sure both rectifiers are pushed all the way against the board.

(3) There is a 90° angle bracket included with your parts. Each of the two sides has two holes in it.

Using the side with the two holes the furthest apart, set the angle bracket over the two rectifiers. The holes in the bracket, the rectifiers, and the board should align.

Temporarily attach the bracket & rectifiers to the board through these holes using #6-32 & 5/8" screws and nuts.

(4) Check the orientation once more, then solder all four leads of each rectifier to the board on the silk-screened side.

(5) Clip off any excess lead lengths. Leave the angle bracket in place for the next procedure.

NOTE: Apply heat-sink compound to all mating surfaces.

BRIDGE RECTIFIER INSTALLATION

There are two bridge rectifiers, BR1 & BR2, to be installed on the power supply board.

WARNING: Read the following instructions closely. Proper orientation of these two components is absolutely critical.

These two components are indicated on the silk-screen by broken lines. This is to indicate that they are to be mounted on the bottom (non-silk-screened) side of the board.

You will observe a "+" sign printed near one corner of the rectifier. The lead nearest this "+" sign is the positive lead of the rectifier. This lead must be inserted into the hole marked on the silk-screen with a "+" sign.

NOTE: There is also a "-" sign printed on the regulator. The lead nearest this sign is the negative lead of the rectifier, and should be diagonally opposite the "+" lead on the board.

BE ABSOLUTELY SURE THAT THE PROPER ORIENTATION IS USED WHEN INSTALLING THESE TWO COMPONENTS.

Install the rectifiers according to the following procedure:

(1) Insert the four leads of the BR1 rectifier into their respective holes from the non-silk-screened side of the board. Be sure the "+" lead of the rectifier is inserted in the hole labeled "+" on the silk-screened side of the board.

(2) Insert the BR2 rectifier in the same manner. Be sure both rectifiers are pushed all the way against the board.

(3) There is a 90° angle bracket included with your parts. Each of the two sides has two holes in it.

Using the side with the two holes the furthest apart, set the angle bracket over the two rectifiers. The holes in the bracket, the rectifiers, and the board should align.

Temporarily attach the bracket & rectifiers to the board through these holes using #6-32 & 5/8" screws and nuts.

(4) Check the orientation once more, then solder all four leads of each rectifier to the board on the silk-screened side.

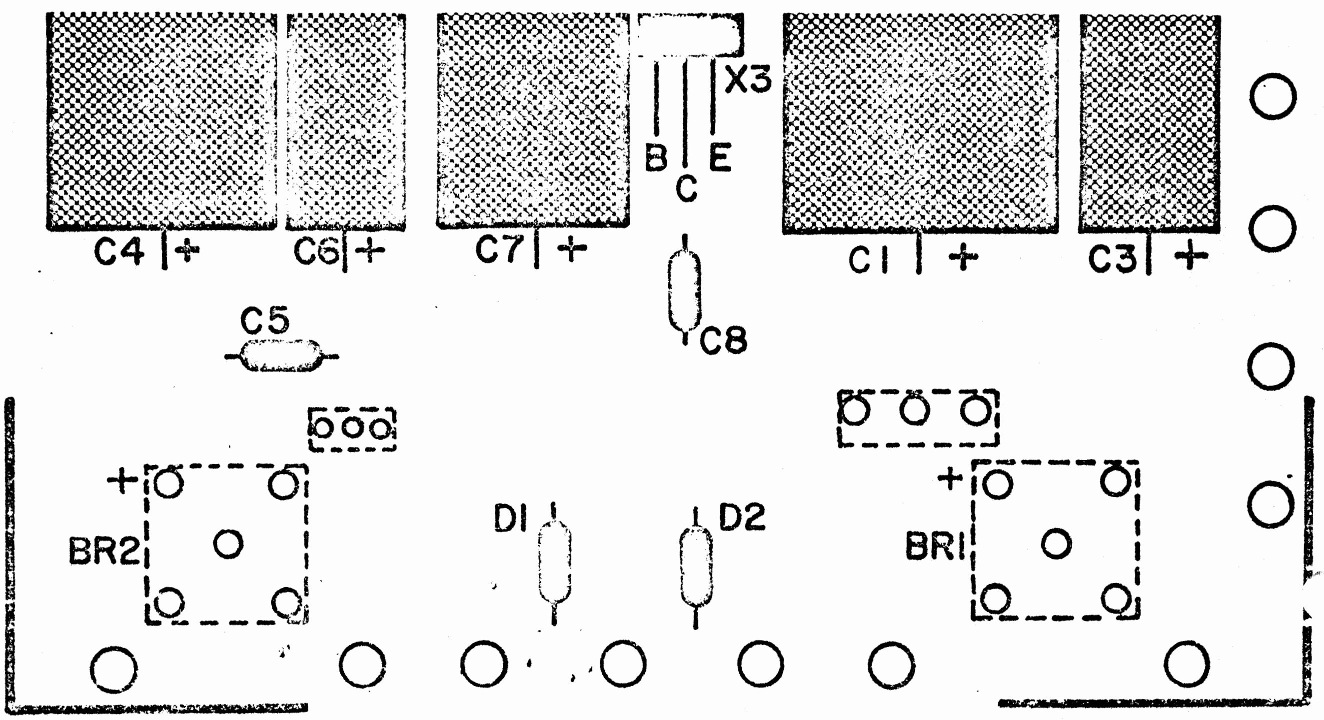
(5) Clip off any excess lead lengths. Leave the angle bracket in place for the next procedure.

NOTE: Apply heat-sink compound to all mating surfaces.

BRIDGE RECTIFIER INSTALLATION

( ) BR1 = VJ048

( ) BR2 = VJ048



DISK BUFFER BOARD
ASSEMBLY

IC INSTALLATION

Install the following 7 ICs onto the Disk Buffer Board according to the method described on page 4 .

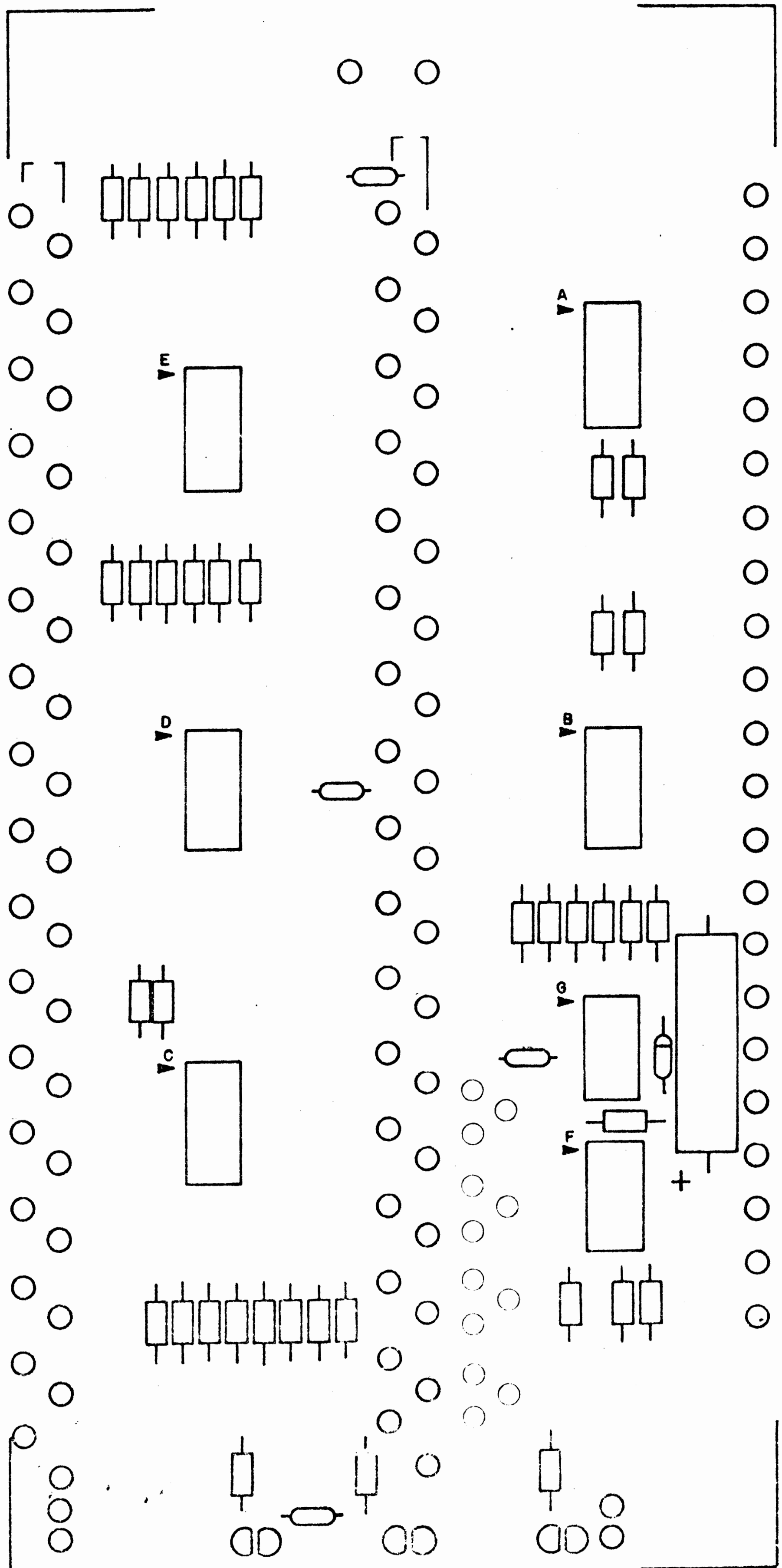
IC SILK-SCREEN DESIGNATIONS AND PART NUMBERS

( ) A, B, D, & E = 8T97

( ) C = 8T98

( ) F = 74L30

( ) G = 9601



RESISTOR INSTALLATION

Install the following 39 resistors according to the instructions listed on page 5 .

RESISTOR VALUES AND COLOR CODES

( ) R9, R7, R5 are 220 ohm
(red-red-brown) 1/2 W

( ) R10, R8, R6 are 330 ohm
(orange-orange-brown) 1/2 W

( ) R12, R14, R16 are 330 ohm
(orange-orange-brown) 1/2 W

( ) R11, R13, R15 are 220 ohm
(red-red-brown) 1/2 W

( ) R33 is 220 ohm
(red-red-brown) 1/2 W

( ) R34 is 330 ohm
(orange-orange-brown) 1/2 W

( ) R31, R29, R27, R25 are 220 ohm
(red-red-brown) 1/2 W

( ) R32, R30, R28, R26 are 330 ohm
(orange-orange-brown) 1/2 W

( ) R36, R35, R37 are 150 ohm
(brown-green-brown) 1/4 W

( ) R40 is 330 ohm
(orange-orange-brown) 1/2 W

( ) R39 is 220 ohm
(red-red-brown) 1/2 W

( ) R38 is 1K ohm
(brown-black-red) 1/2 W

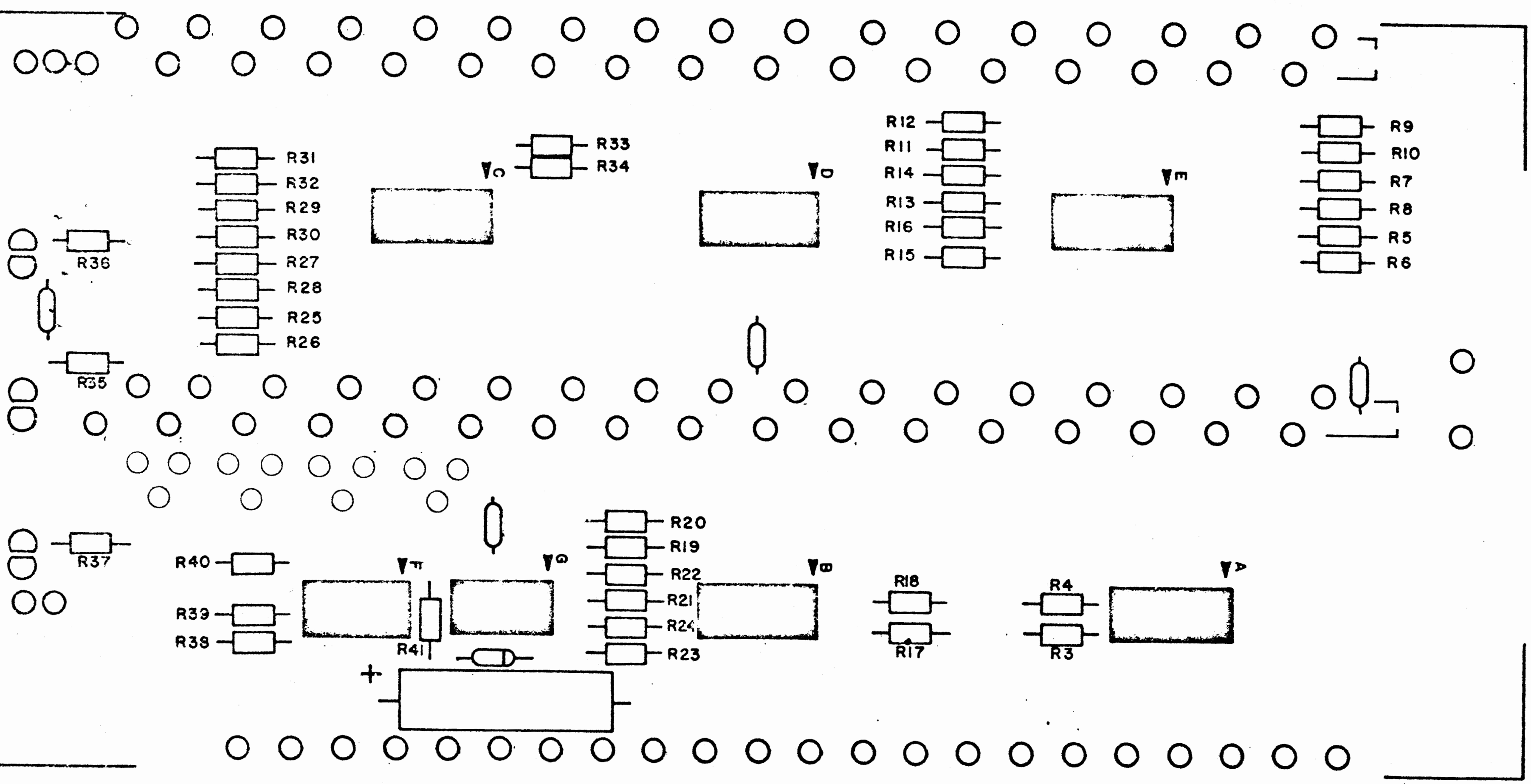
( ) R91 is 39K ohm
(orange-white-orange) 1/2 W

( ) R20, R22, R24 are 330 ohm
(orange-orange-brown) 1/2 W

( ) R19, R21, R23 are 220 ohm
(red-red-brown) 1/2 W

( ) R4 & R18 are 330 ohm
(orange-orange-brown) 1/2 W

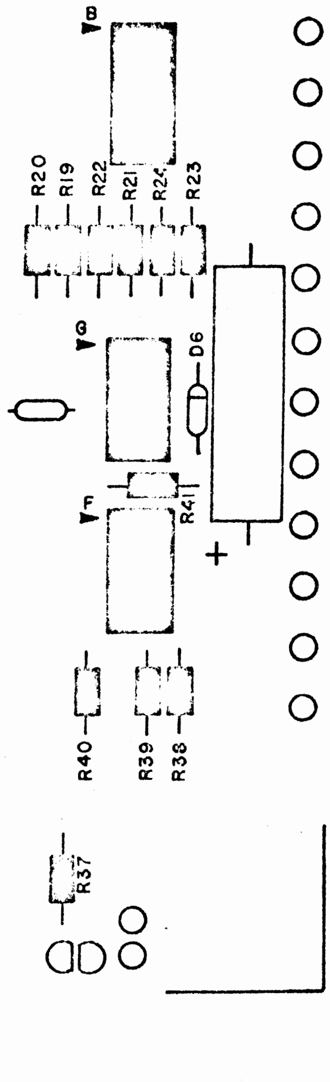
( ) Rd & R17 are 220 ohm
(red-red-brown) 1/2 W


R31
R32
R29
R30
R27
R28
R25
R26
R33
R34
R36
R35
R12
R11
R14
R13
R16
R15
R9
R10
R7
R8
R5
R6
R20
R19
R22
R21
R24
R23
R37
R40
R39
R38
R41
R18
R17
R4
R3
+

DIODE INSTALLATION

Install diode D6 according to the instructions on page 7 .

( ) D6 = 1N914



CAPACITOR INSTALLATION

Capacitor C14 is an electrolytic capacitor. Capacitors C10, C11, C12, and C13 are ceramic disk capacitors.

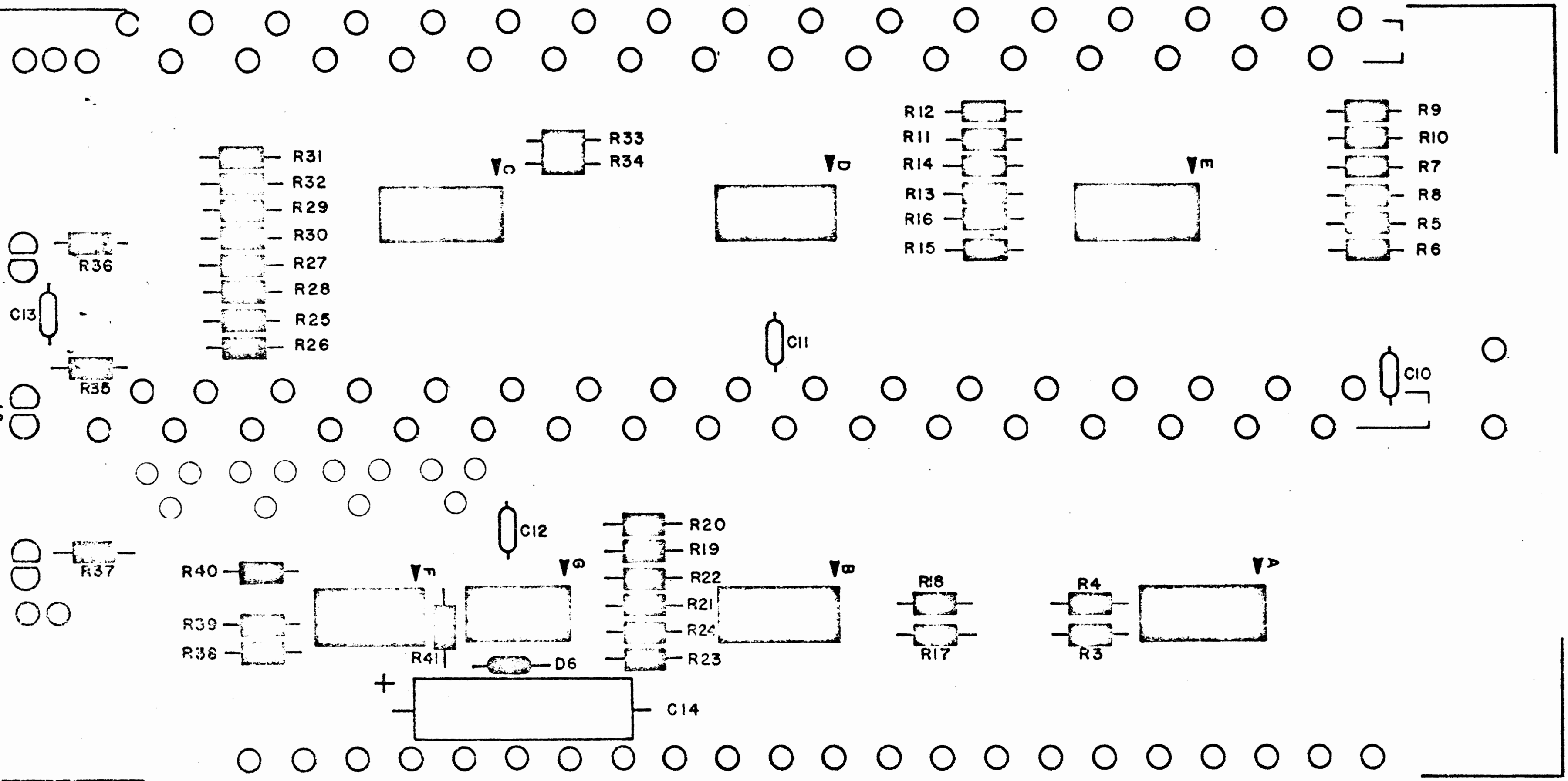
Install these components according to the instructions listed on page 6 .

CAPACITOR VALUES

(Different voltages may be substituted in some cases.)

( ) C14 = 500 uf, 25V electrolytic

( ) C10, C11, C12 & C13 are .1 uf, 12V ceramic disks.



R33
R34
R12
R11
R14
R13
R16
R15
R9
R10
R7
R8
R5
R6
R31
R32
R29
R30
R27
R28
R25
R26
C
D
M
R36
C13
R35
C11
C10
C12
R20
R19
R22
R21
R24
R23
R37
R40
R39
R38
R41
F
G
B
A
D6
+
C14
R18
R17
R4
R3

DISK DRIVE RIBBON CABLE ASSEMBLY

Ribbon Cable Preparation

There are three ribbon cable assemblies to be prepared for installation in the disk drive unit. A 12' length of 18-twisted pairs cable has been provided for this purpose.

First, cut the 12' length of cable into two 18-inch lengths and one 25-inch length. The remainder of the cable should be saved for later use.

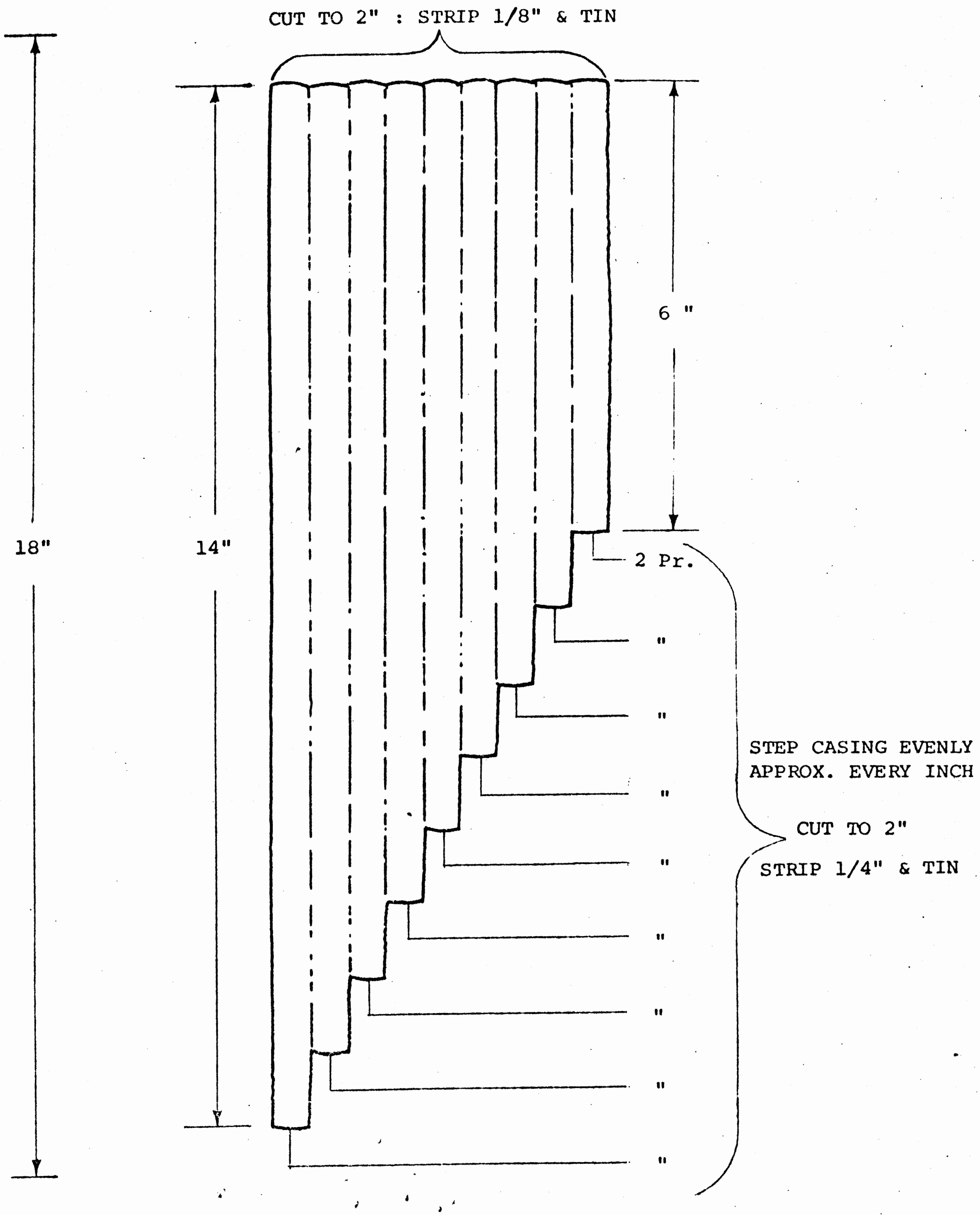
The following two pages contain diagrams for the proper lengths and arrangement for the three cable pieces you have just cut. The two 18" lengths will be prepared identically.

The cable sheath itself may be cut using scissors, and can be stripped by simply pulling it apart. You will note that the plastic sheath has "welds" approximately every inch between the twisted pairs. Try not to make any cuts on the welds themselves.

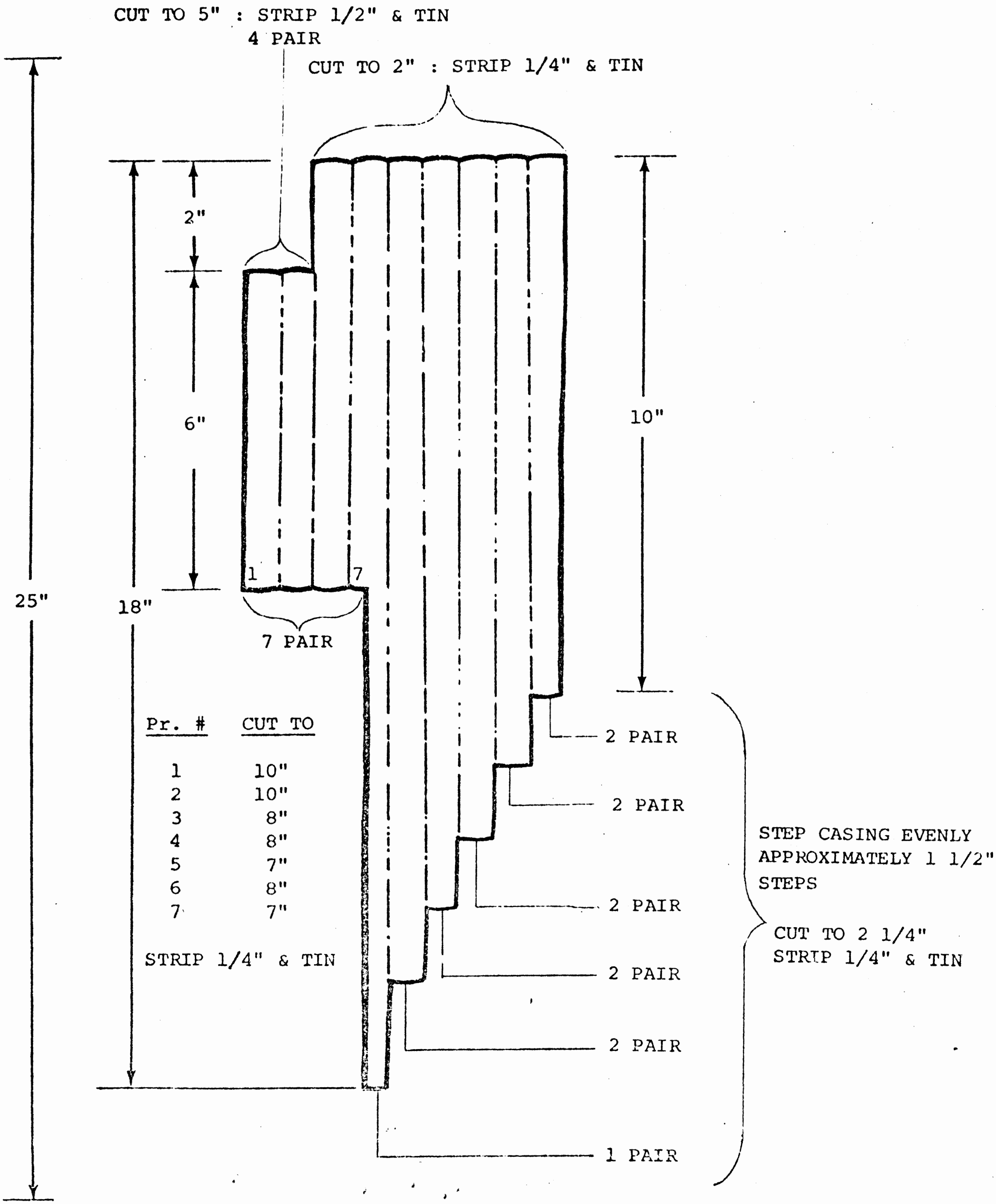
Each time a 1/4" of insulation is stripped from the wires themselves, the bare ends should be tinned by applying a thin coat of solder.

Study the diagrams on the next two pages and prepare the three cable assemblies as shown. Be careful to cut the wires precisely as indicated, and do not damage the wire insulation when cutting the cable sheath.

18 INCH RIBBON CABLES



25 INCH RIBBON CABLE



Pr. #	CUT TO
1	10"
2	10"
3	8"
4	8"
5	7"
6	8"
7	7"

STRIP 1/4" & TIN

There are several 37-pin connectors in this kit. One male connector and one female connector will be used now to connect onto one end of each of the two 18 inch lengths of ribbon cable that you have just prepared. The other end of the two cables will connect directly to the Disk Buffer board.

Connector Preparation

The two 37-pin connectors must first be prepared for attaching to the cables. It may be helpful to solidly mount the connectors to some steady object during this and the following procedures.

1) Place the connector in front of you with the hollow solder pins facing upwards.

2) Using your soldering iron, very carefully heat each pin one at a time and fill the hollow space with solder. The solder should not quite fill the pin and should have a slightly concave surface.

Prepare all 37 pins on one male and one female connector in this manner. Be sure not to leave any solder bridges between the pins, and be careful not to melt any of the nylon insulation around them.

WARNING

During the following procedure, and later steps involving ribbon cable, be sure that you fully understand all of the instructions before you begin. These points are the most likely areas for assembly errors to occur.

Cable Assembly

The following procedure should be used for assembling both of the 18 inch cables. In order to minimize the possibility of error, the cables will be attached to the 37-pin connectors and the Disk Buffer board during the same procedure. Read this entire procedure over carefully before beginning.

You will note that the pins on the 37-pin connectors are all numbered. Note also that the numbers on the male connector are the reverse of the female. The male connector will be wired to the rows of pads on the buffer board labeled "TO". The female connector will be wired to the rows of pads labeled "FROM". The numbers on the connector pins correspond directly with the numbers that label the pads on the buffer board.

The following pages contain drawings of both the 37-pin connectors, and the Disk Buffer board silk-screen. There is a space provided to "check-off" each of the twisted-pair wires as they are connected. Double arrows are also shown to indicate the connection points for each of the twisted-pairs.

Orient one of the 18 inch cables so that the "stepped" edge of the cable casing is along the rows of pads on the buffer board labeled "TO". The longest wires should be near the pads labeled "19 & 37" and the shortest wires near the pads labeled "1 & 20". Place the MALE 37-pin connector near the other end of the cable.

Begin with the shortest twisted-pair of wires, nearest the outside edge of the cable casing, on the buffer board end.

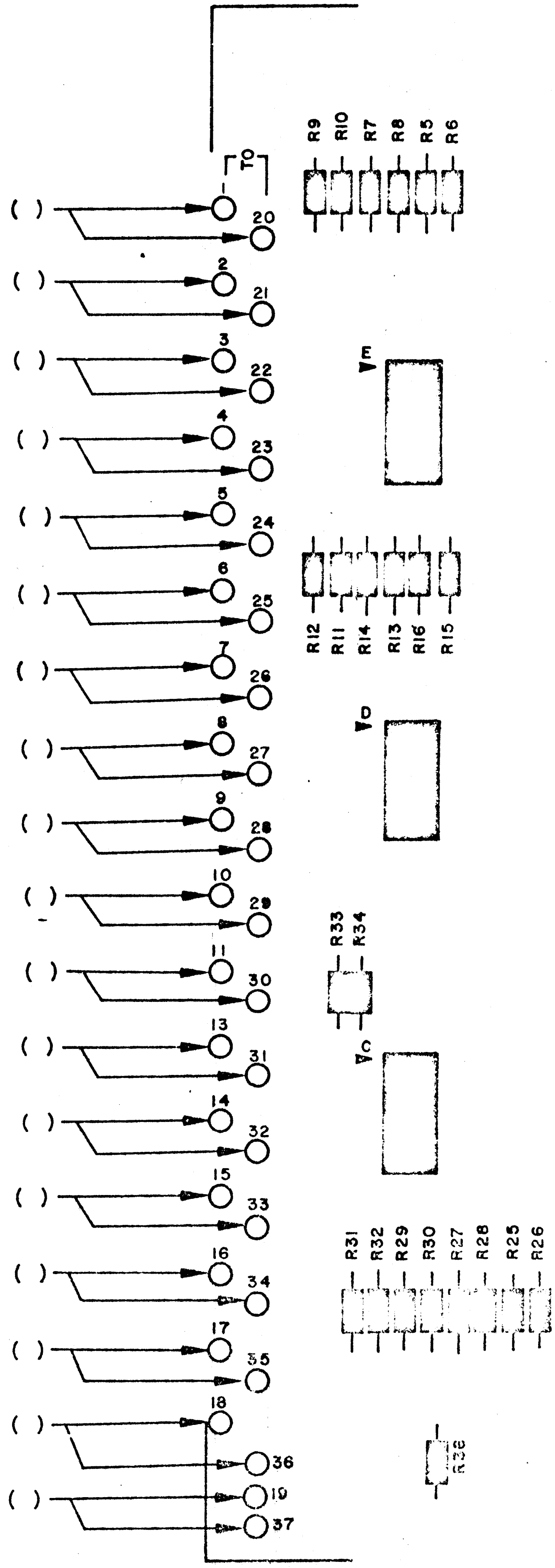
Separate the two wires slightly, then solder them into the two pads labeled "1 & 20" on the buffer board. Do this by inserting the wires from the silk-screened side of the board and soldering them on the back. Be careful not to push any of the wire insulation into the holes. Clip off any excess wire from the connections and then check-off the appropriate space on the silk-screen drawing.

The same twisted-pair of wires should now be connected to the pins numbered "1 & 20" on the 37-pin connector.

Observe the color of the wire now connected to the pad on the buffer board labeled "1". Be sure to connect this same wire to the pin numbered "1" on the connector. Do the same with pad "20" and pin "20".

Make the connections by re-melting the solder in the pins and inserting the wires up to their insulation. Remove the heat from the pins while still holding the wires in place until the solder cools. Check-off the appropriate space on the connector drawing.

Move to the next twisted-pair of wires in the ribbon cable and use the same procedure to connect pads "2 & 21" with pins "2 & 21". Continue in this manner, moving across the ribbon cable one pair at a time, until all 18 twisted-pairs are in place. Be sure that you do not connect any wires to pin "12" on the connector.

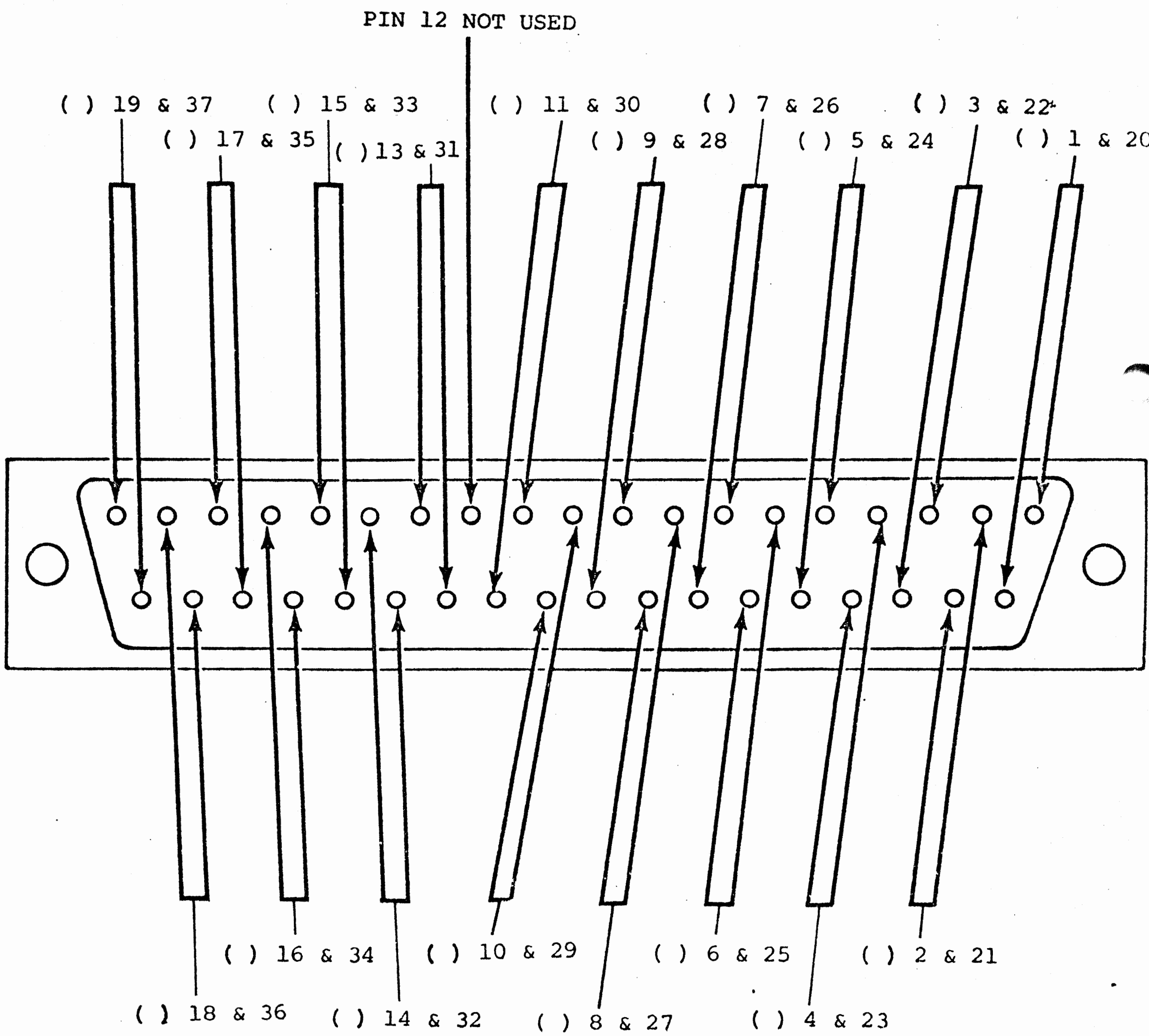


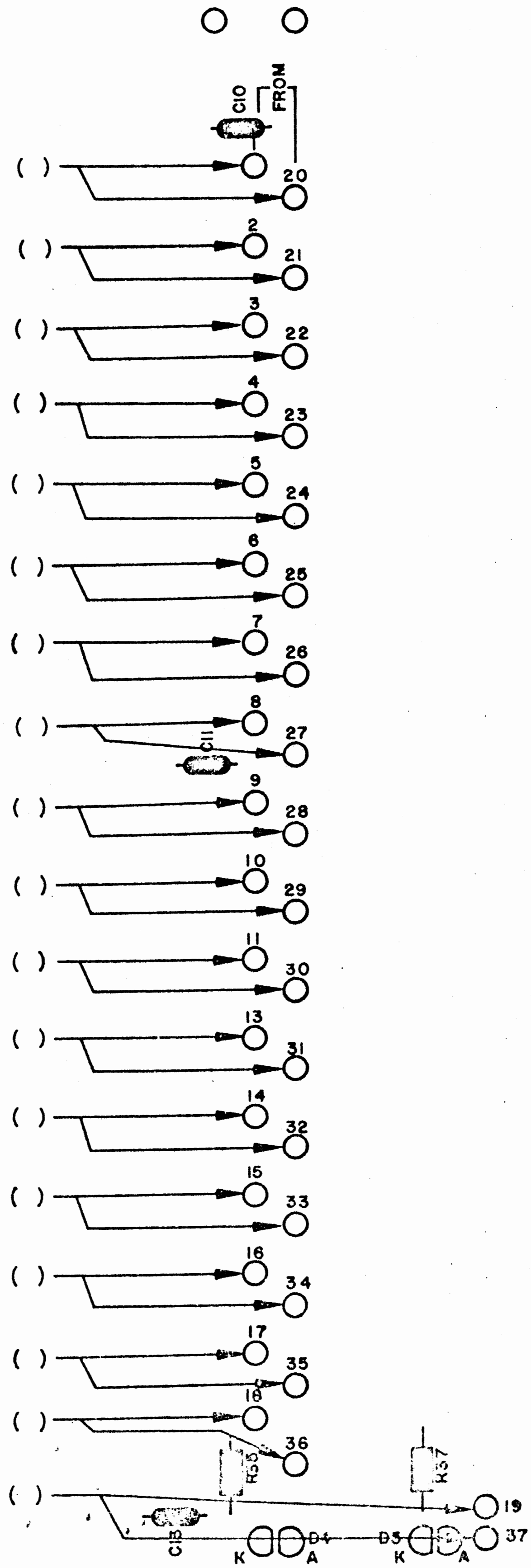
<u>NOTE</u>: Take your time and be careful while soldering the wires to the connectors. Do not melt any of the wire insulation or leave any solder bridges.

Check your work as you go along and be <u>sure</u> that 1 is connected to 1, 2 to 2, 3 to 3, etc., because corrections will be very difficult later.

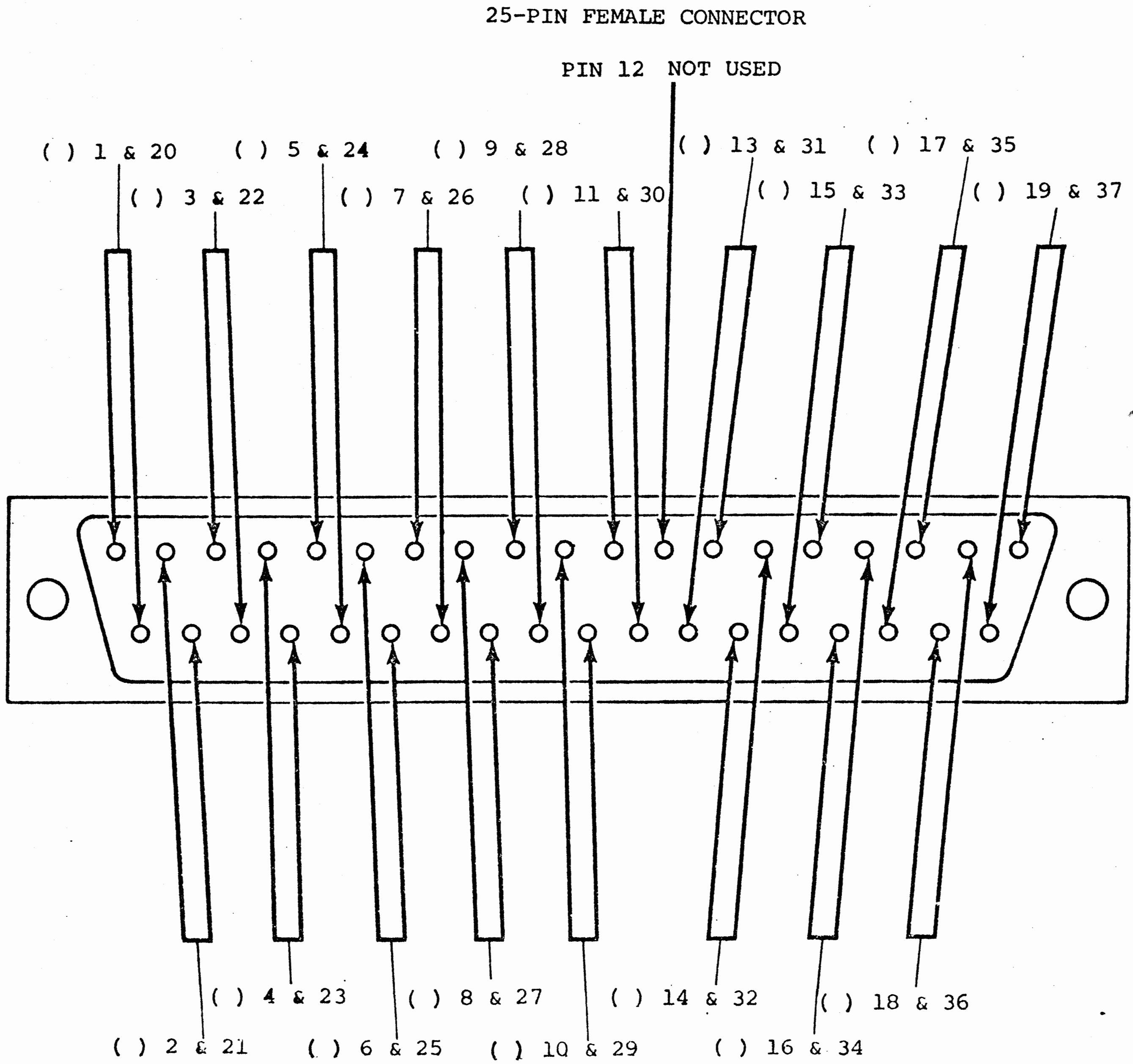
Use this procedure to assemble both of the 18 inch cables. Be sure that the MALE 37-pin connector goes to the pads labeled "TO" and the FEMALE connector to the pads labeled "FROM". Refer to the drawing on page to get a rough idea of how these and the next cable will appear when connected to the board.

25-PIN MALE CONNECTOR





FROM
C10
20
2
21
3
22
4
23
5
24
6
25
7
26
8
27
C11
9
28
10
29
11
30
13
31
14
32
15
33
16
34
17
35
18
36
R35
R37
19
C13
K
D4
A
D5
K
A
37


25-PIN FEMALE CONNECTOR
PIN 12 NOT USED
( ) 1 & 20
( ) 3 & 22
( ) 5 & 24
( ) 7 & 26
( ) 9 & 28
( ) 11 & 30
( ) 13 & 31
( ) 15 & 33
( ) 17 & 35
( ) 19 & 37
( ) 2 & 21
( ) 4 & 23
( ) 6 & 25
( ) 8 & 27
( ) 10 & 29
( ) 14 & 32
( ) 16 & 34
( ) 18 & 36

Due to its complexity, the 25 inch length of ribbon cable will be assembled in a slightly different manner.

The following two pages contain drawings of one end of the ribbon cable and the 44-pin edge connector included with this kit. These connections, on one end of the ribbon cable only, will be made first.

> NOTE: Be sure to observe that the orientation of the edge connector is not the same in all of the drawings. Use the pin designations themselves for any reference when making connections.

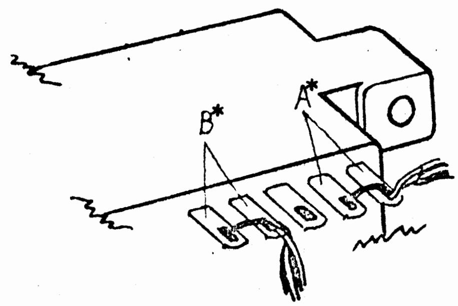
Orient the 25 inch ribbon cable as shown in the drawing on page 46 . The end that is shown at the top of this drawing will be attached to the 44-pin edge connector. The Connection Chart on the following page also refers to this drawing for the proper orientation. Twisted-pair #1 is the pair furthest to the right in the drawing, and pair #18 is furthest to the left. It is very important to begin numbering from the correct side when making the connections.

The Connection Chart on the following page indicates where on the edge connector each twisted-pair should be attached. The pin designations in the chart and in the drawings refer to those stamped into the plastic of the connector itself. Be sure that you connect the proper wires to the correct pins according to the designations stamped on the connector.

In most cases a single wire will connect to a single pin on the connector. Make these connections by first making a good mechanical connection, and then soldering the wire into place. Be careful not to leave any solder bridges, or to melt any insulation.

For twisted-pair #12, and pair #13, you will connect both wires of the pair to a single pin instead of each to a separate one.

For twisted-pairs #15 & #16, all four of the wires should first be twisted together and then all four attached to both of the pins A & B. Do the same for pairs #17 & #18 to connect them to pins D & E. Be sure that there is a solid electrical connection between both of the pins in each case. (see drawing below)



A*=pairs #15 & #16
B*=pairs #17 & #18

Be sure to check-off the appropriate space on the chart as you make each of the connections.

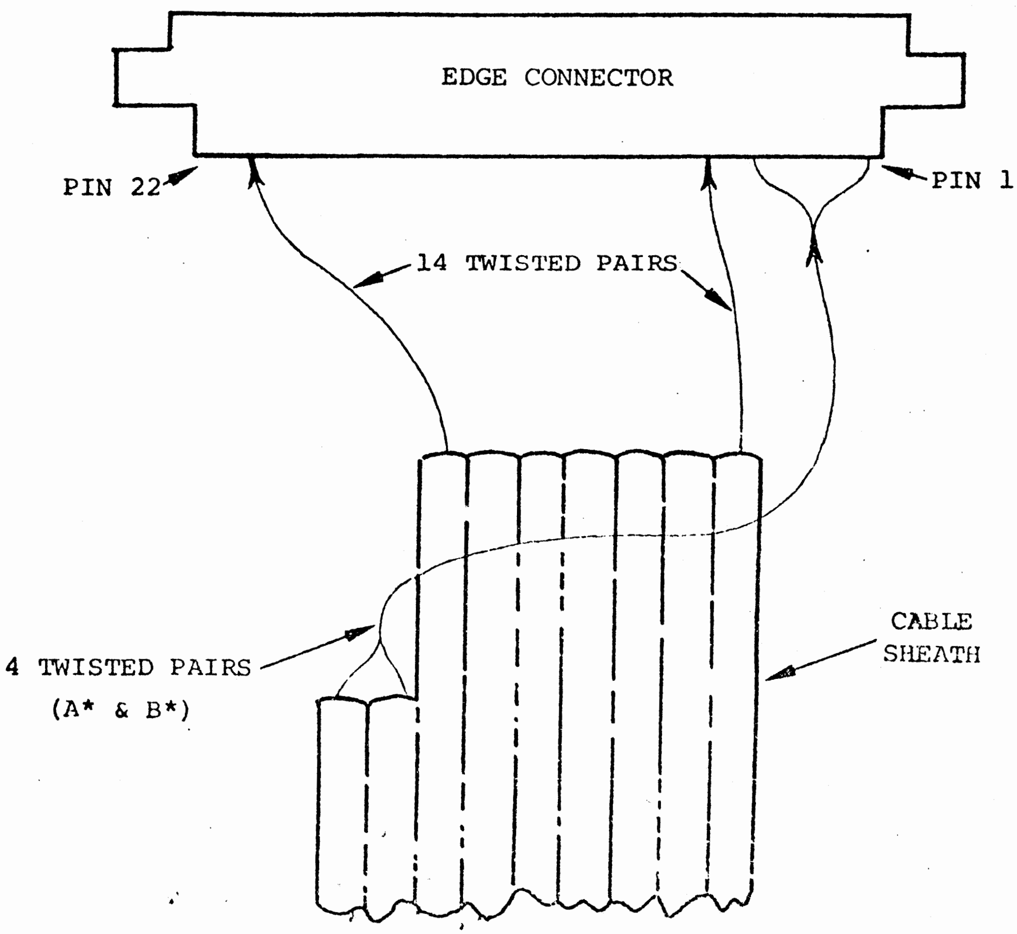
Use a small piece of ribbon cable wire to connect pin 18 to pin V on the edge connector.

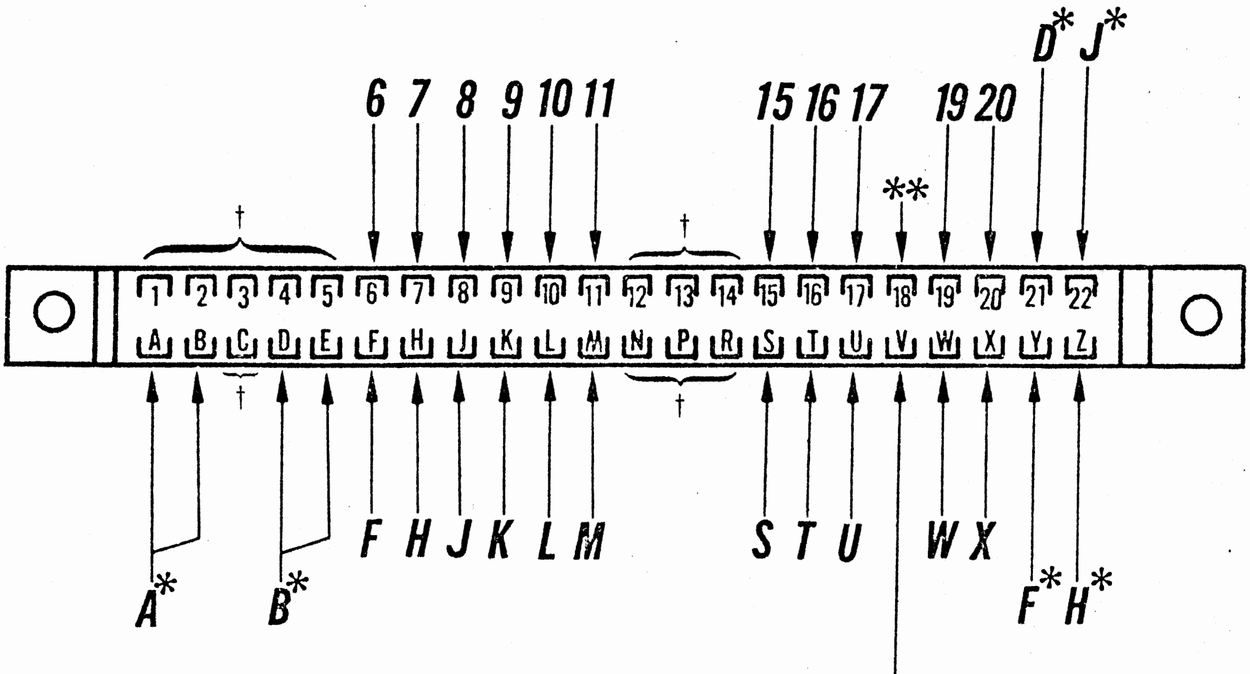
Insert the plastic key, packaged with the edge connector, into the slot between pins 5 & 6 as shown in the drawing on the bottom of page .

CONNECTION CHART

PAIR #	CONNECTOR PIN(S)	PAIR #	CONNECTOR PIN(S)
1	6 & F ( )	10	19 & W ( )
2	7 & H ( )	11	20 & X ( )
3	8 & J ( )	12	21 ( )
4	9 & K ( )	13	Y ( )
5	10 & L ( )	14	22 & Z ( )
6	11 & M ( )	15 }	A & B ( )
7	15 & S ( )	16 }	
8	16 & T ( )	17 }	D & E ( )
9	17 & U ( )	18 }	

( ) Jumper 18 to V

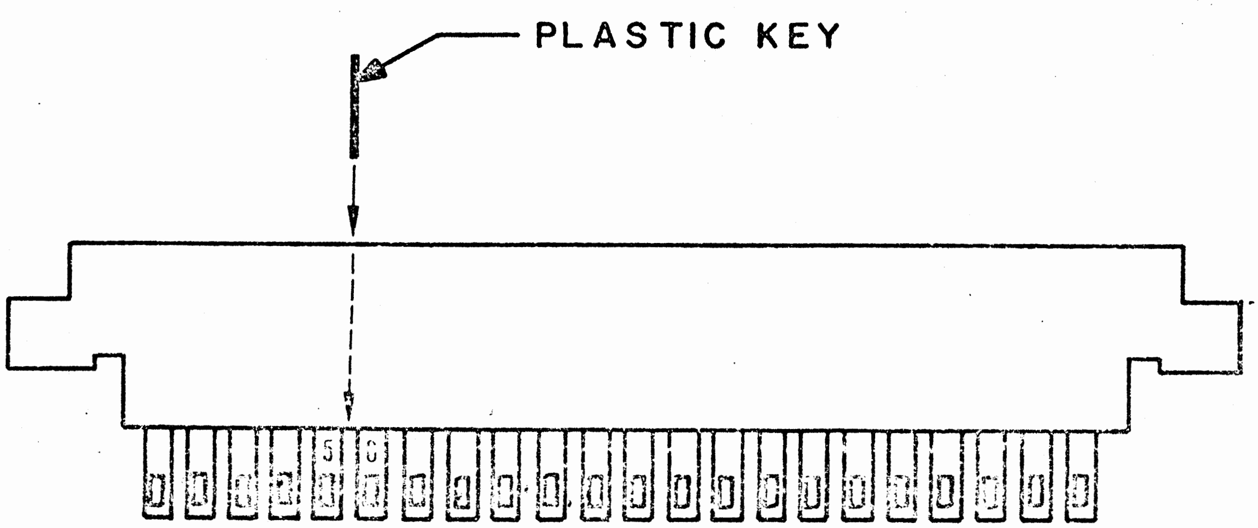




† - NOT USED

* - POWER SUPPLY BOARD

** - JUMPER WIRE (18 to V)



The other end of the ribbon cable will connect to both the Disk Buffer board and the Power Supply board.

When making these connections, the same numbering system will be used for the twisted-pairs as previously. That is, the pair furthest to the right in the drawing on page   will be refered to as pair #1.

Page   contains silk-screen drawings of both PC boards, with arrows to indicate the twisted-pair connections and a space to check-off each as it is completed.

The first eleven twisted-pairs will connect to the remaining row of pads on the Disk Buffer board. Make these connections in the same manner as the previous ribbon cable connections to this board.

Begin with pair #1 and connect one of its wires to pad 6 and the other to pad 7 on the board. Observe the color of the wires connected to the equivalent pins on the edge connector. Be sure you connect pin 6 to pad 6, F to F, etc., as when making the previous connections. Continue the connections through the first eleven of the twisted-pairs in this manner, checking-off the appropriate space as each is completed.

The next seven twisted-pairs will connect to the Power Supply board in nearly the same manner, except that all but two of the connections involve more than one of the wires.

The two wires of pair #12 should be twisted together and both connected to pad D. Pair #13 should connect to pad F in the same manner.

Twisted-pairs #15 & #16 should have all four wires (2 each) twisted together and connected to pad A. Pairs #17 & #18 should be connected to pad B in the same manner.

Only twisted-pair #14 should be separated and connected to pads J & H in the same manner as the first eleven pair.

Make all of the Power Supply board connections as described, checking-off the appropriate space as you complete each of them.

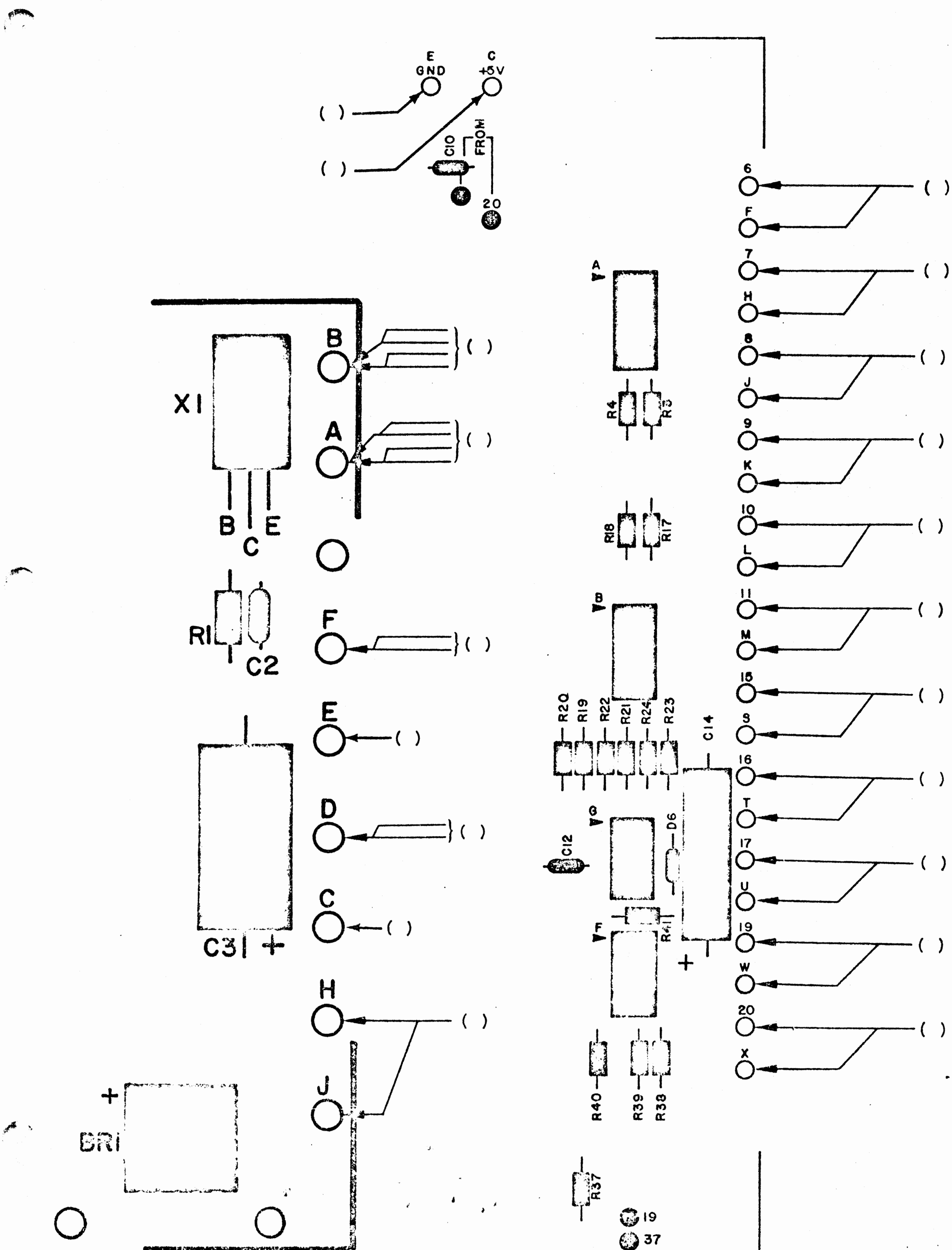
Starting approximately 1 inch from the cable casing, and moving along the Power Supply cable wires, attach a tie-wrap approximately every inch until 5 of them are used. Do these as necessary to make a neat, tight cable.

There are two other wires which should be installed at this time. Using the same wire that you used when making the connections to the terminal block, cut one 8 inch length of orange wire and one 8 inch length of black wire. Strip 1/4 inch of insulation from both ends on each of them and tin the exposed portion.

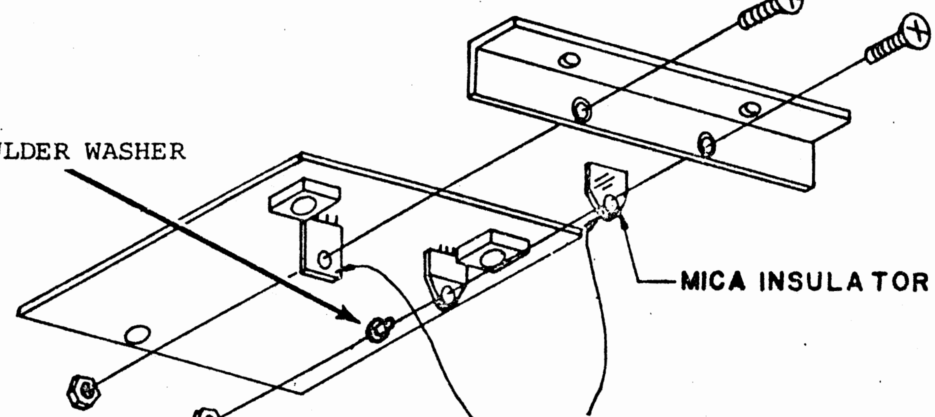
Connect the orange wire between pad C on the Power Supply board and pad C on the buffer board. +5V

Insert the wire from the silk-screened side of the board and solder it on the bottom.

Connect the black wire between E and E in the same manner. GND



E
GND
C
+5V
C10
FROM
20
X1
B
C
E
B
A
F
E
D
C
H
J
R1
C2
C3
BR1
A
B
G
F
R4
R3
R18
R17
R20
R19
R22
R21
R24
R23
C14
C12
D6
R41
R40
R39
R38
R37
19
37
6
F
7
H
8
J
9
K
10
L
11
M
15
S
16
T
17
U
19
W
20
X



VOLTAGE REGULATOR
INSTALLATION

The next two components will be mounted on the bottom side of the Power Supply board. These components will also be mounted to the 90° angle bracket, as with BR1 & BR2, in the two remaining holes.

When installing these components refer to the drawing above and orient them so that the markings on the components face away from the bracket.

Insert the two regulators from the bottom side of the board as shown. *Use heatsink compound between all mating surfaces. Be sure to place the mica insulating washer between Q1 and the bracket, and the shoulder washer between Q1 and the mounting nut.

Tighten the mounting screws firmly, being sure not to twist the component leads as you do so.

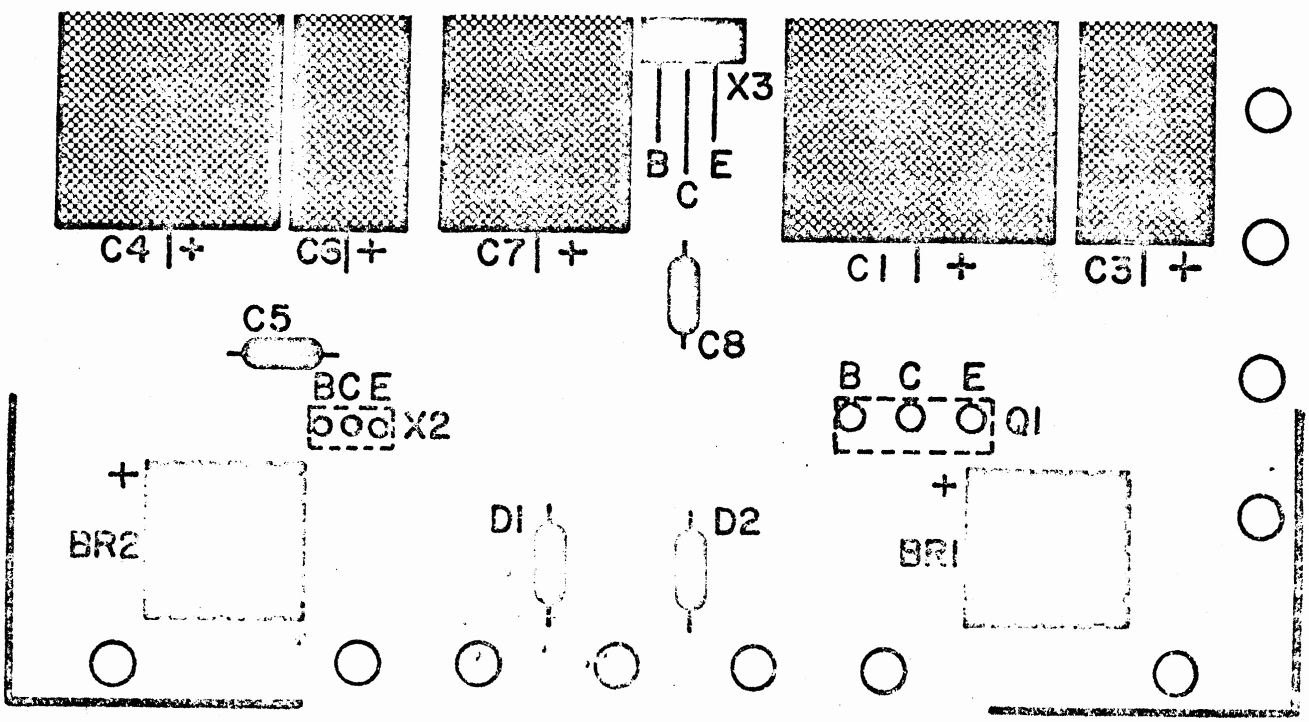
Solder all three leads of both components to the board on the silk-screened side.

Clip off the excess lead lengths; then remove the two screws used earlier to mount BR1 & BR2. The screws mounting X2 & Q1 should remain.

<u>VOLTAGE REGULATOR INSTALLATION</u>

( ) X2 = 7805

( ) Q1 - TIP 145 (w/Mica insulating washer and shoulder washer)



DISK CHASSIS ASSEMBLY

The next step in the assembly procedure is to prepare the chassis itself for mounting the boards and drive unit.

1) Refering to the drawing on the following page, mount the cross beam as shown using the existing screws now holding it in place. Note the number of holes for proper placement.

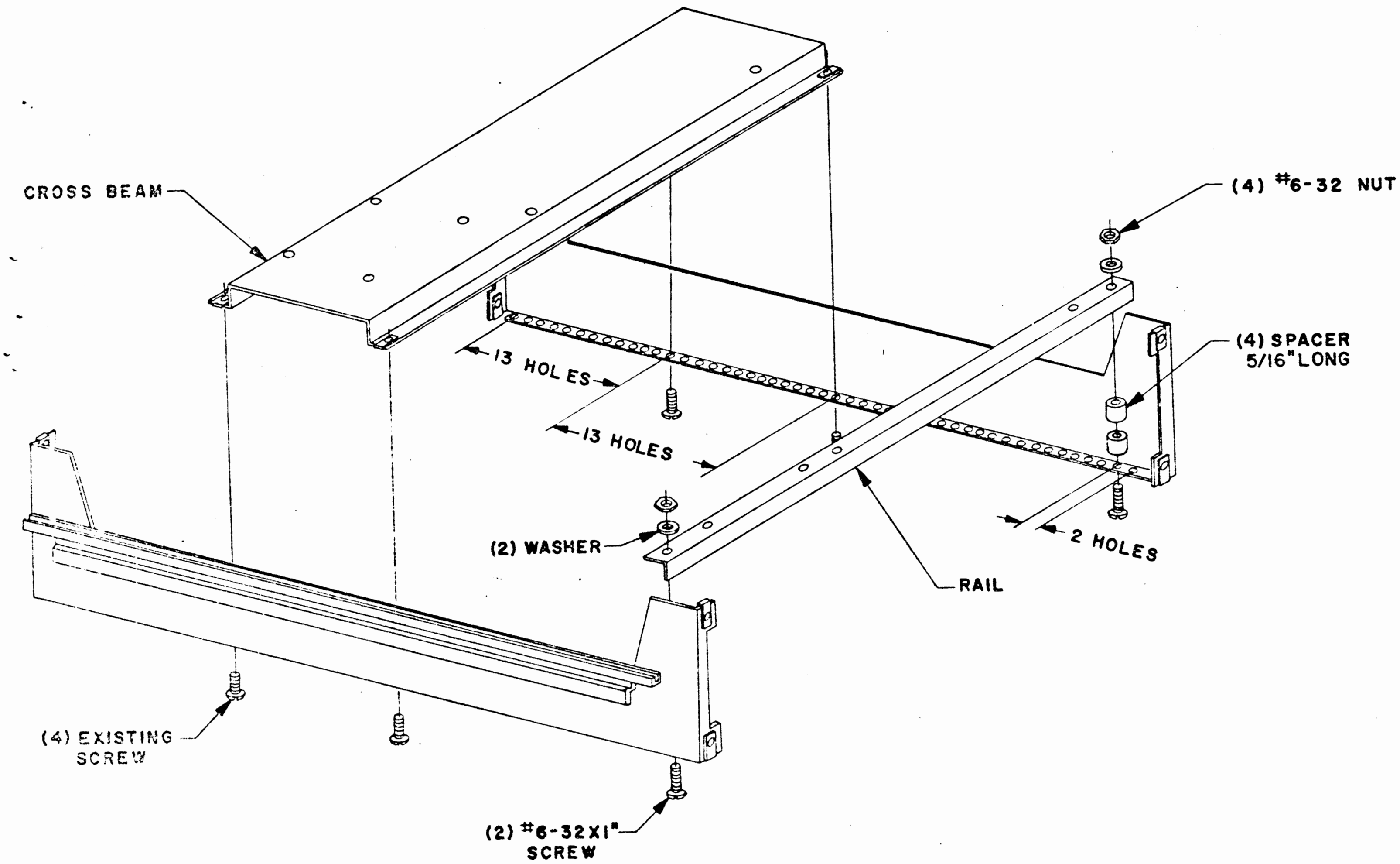
2) To make the following procedures as simple as possible, remove the front panels at this time. Save the screws used to mount the panel to the chassis.

3) Refering to the same drawing again, mount the rail as shown in the 2nd hole from the front. Be sure to include the 2 spacers as shown on each side.

There are 6 additional screws to be added to the chassis members, 4 on the beam and 2 on the rail.

4) Install two #6-32 x 3/4" screws onto the rail in the positions indicated on the same drawing. Insert them from the bottom and tighten them firmly using #6-32 lockwashers and nuts.

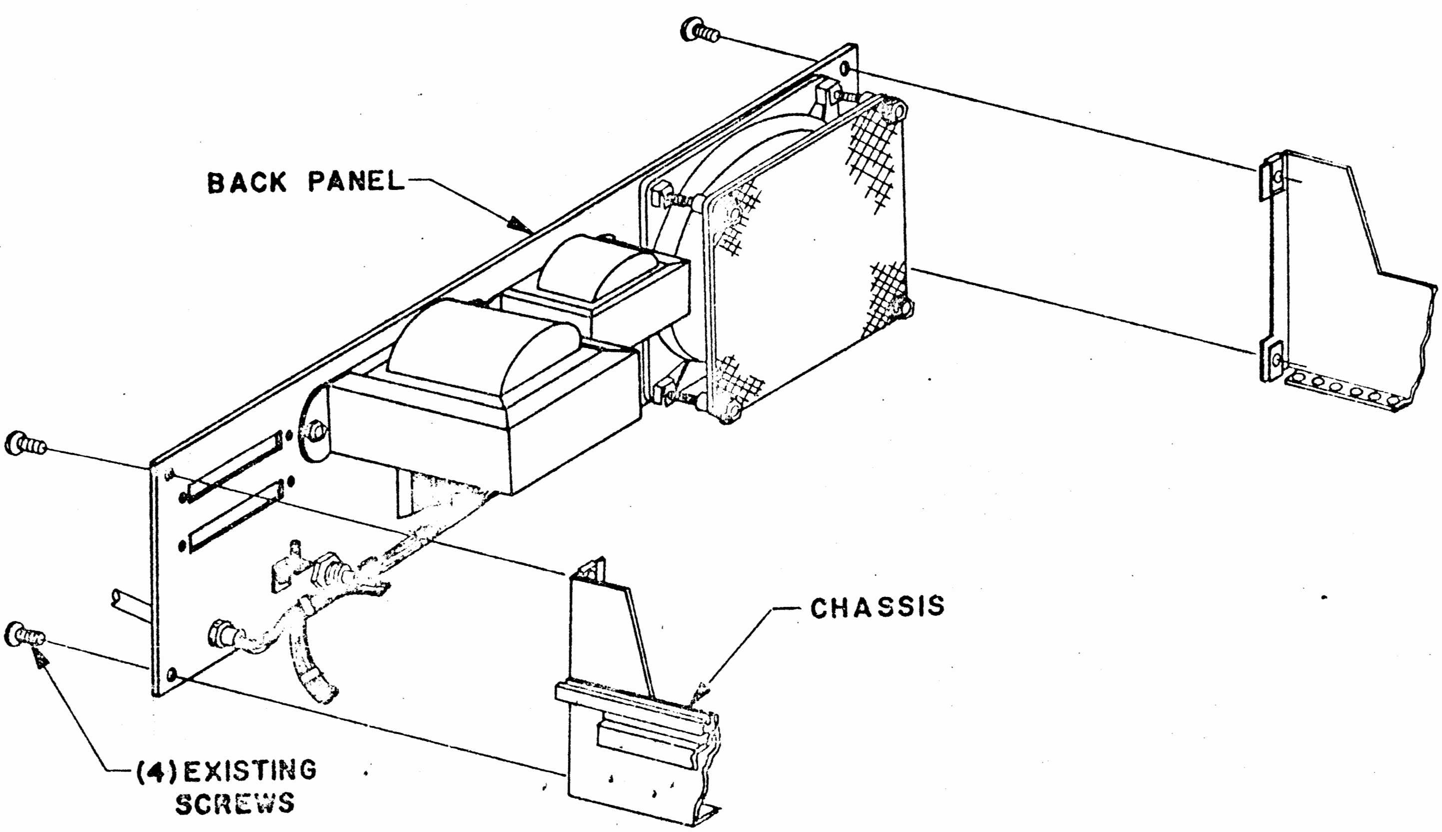
5) Install two 4-40 x 1" screws and two 6-32 x 1" screws on the cross beam as shown using the indicated hardware.


CROSS BEAM
(4) #6-32 NUT
(4) SPACER
5/16" LONG
13 HOLES
13 HOLES
2 HOLES
(2) WASHER
RAIL
(4) EXISTING
SCREW
(2) #6-32X1"
SCREW

BACK PANEL MOUNTING

Mount the back panel to the rear of the chassis as shown below using the same screws previously used to mount it.

Be careful not to catch any wires between the chassis and the panel.

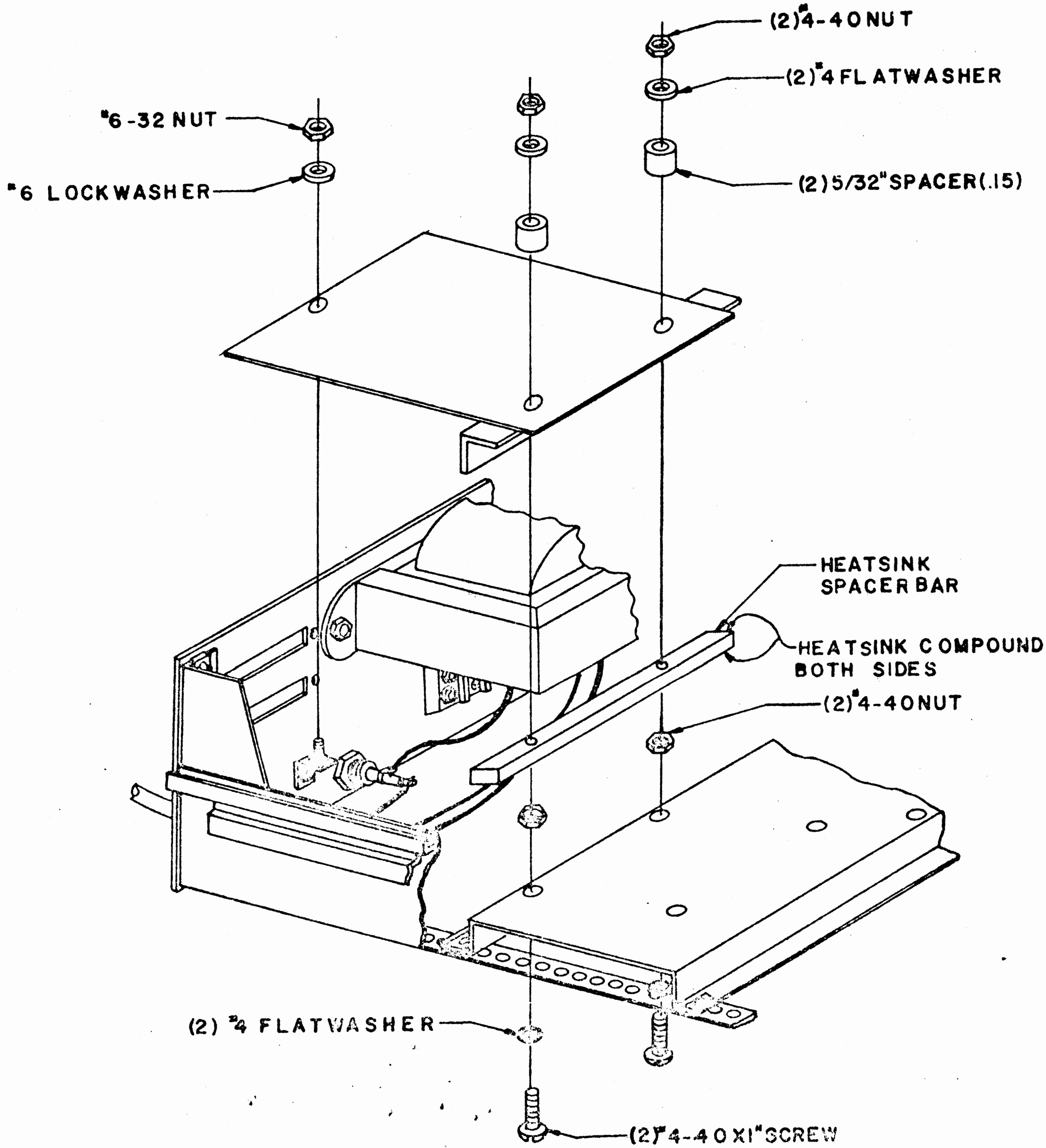


POWER SUPPLY BOARD MOUNTING

Refering to the drawing on the following page, mount the Power Supply board to the 90° angle clip and bracket as shown. Study the drawing carefully before beginning.

NOTE: The #4-40 screw shown are those installed earlier.

Be careful not to disturb the wire connects previously made between this board and the buffer board and cables.

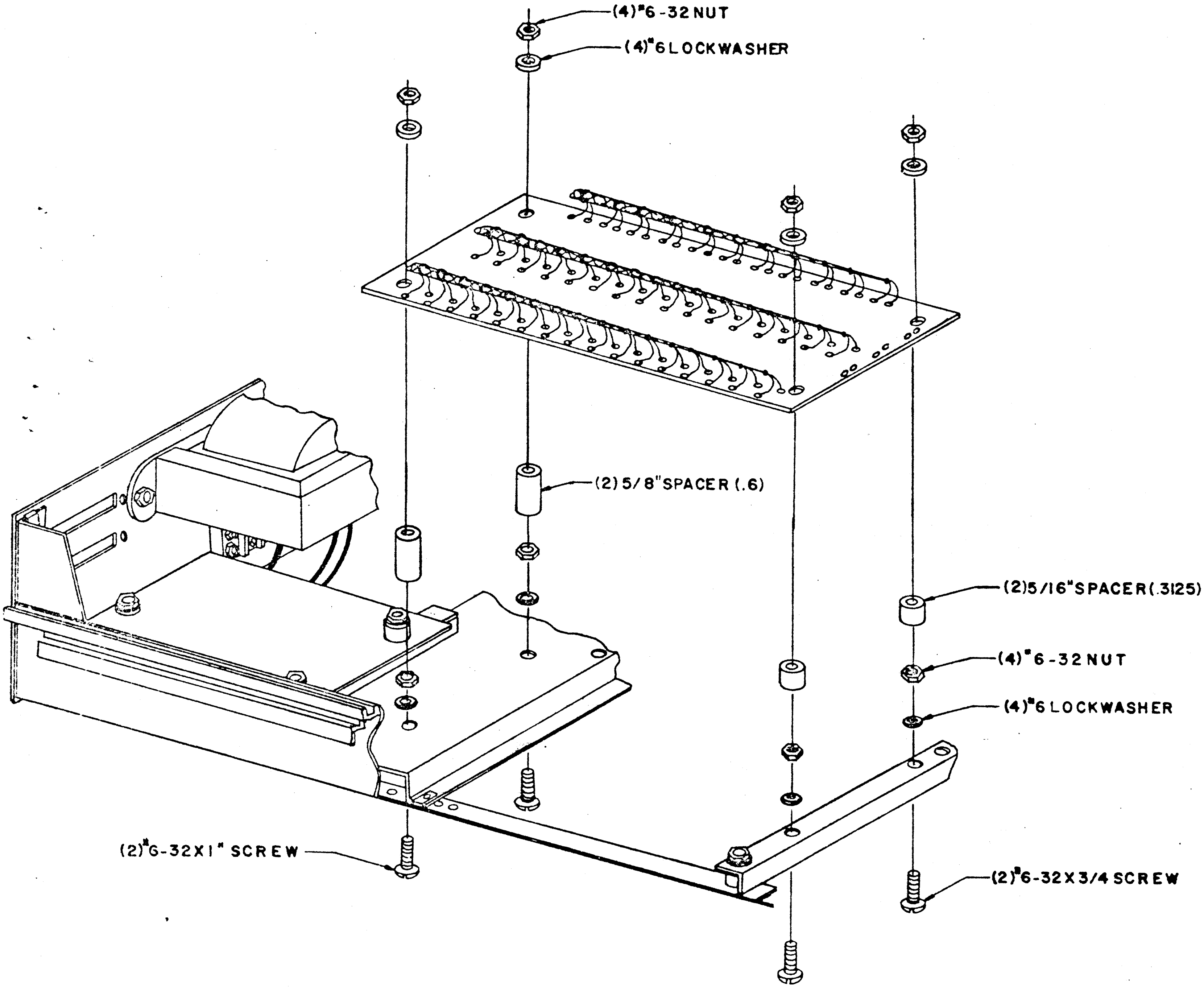

(2)#4-40 NUT
(2)#4 FLATWASHER
#6-32 NUT
#6 LOCKWASHER
(2) 5/32" SPACER(.15)
HEATSINK
SPACER BAR
HEATSINK COMPOUND
BOTH SIDES
(2)#4-40 NUT
(2) #4 FLATWASHER
(2) #4-40 X1" SCREW

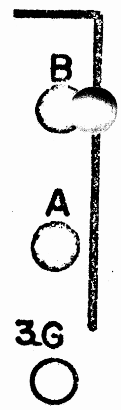
DISK BUFFER BOARD MOUNTING

Refering to the drawing on the following page, mount the Disk Buffer board as shown.

Again, study the drawing carefully before beginning. The screws shown have already been installed.

The connectors on the three cables should face towards the back panel.


(4)#6-32 NUT
(4)#6 LOCKWASHER
(2) 5/8"SPACER (.6)
(2)5/16"SPACER(.3125)
(4)#6-32 NUT
(4)#6 LOCKWASHER
(2)#6-32X1" SCREW
(2)#6-32X3/4 SCREW



POWER SUPPLY WIRING

Refering to the silk-screen drawing below, and the wiring diagram on the following page, connect the wires from the terminal block to the pads on the Power Supply board.

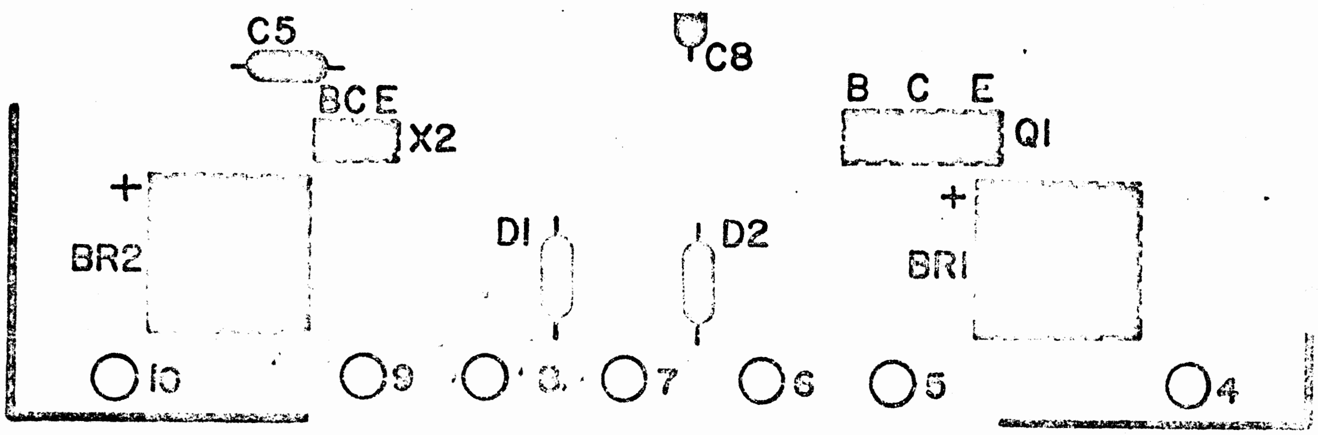
Use the following procedure:

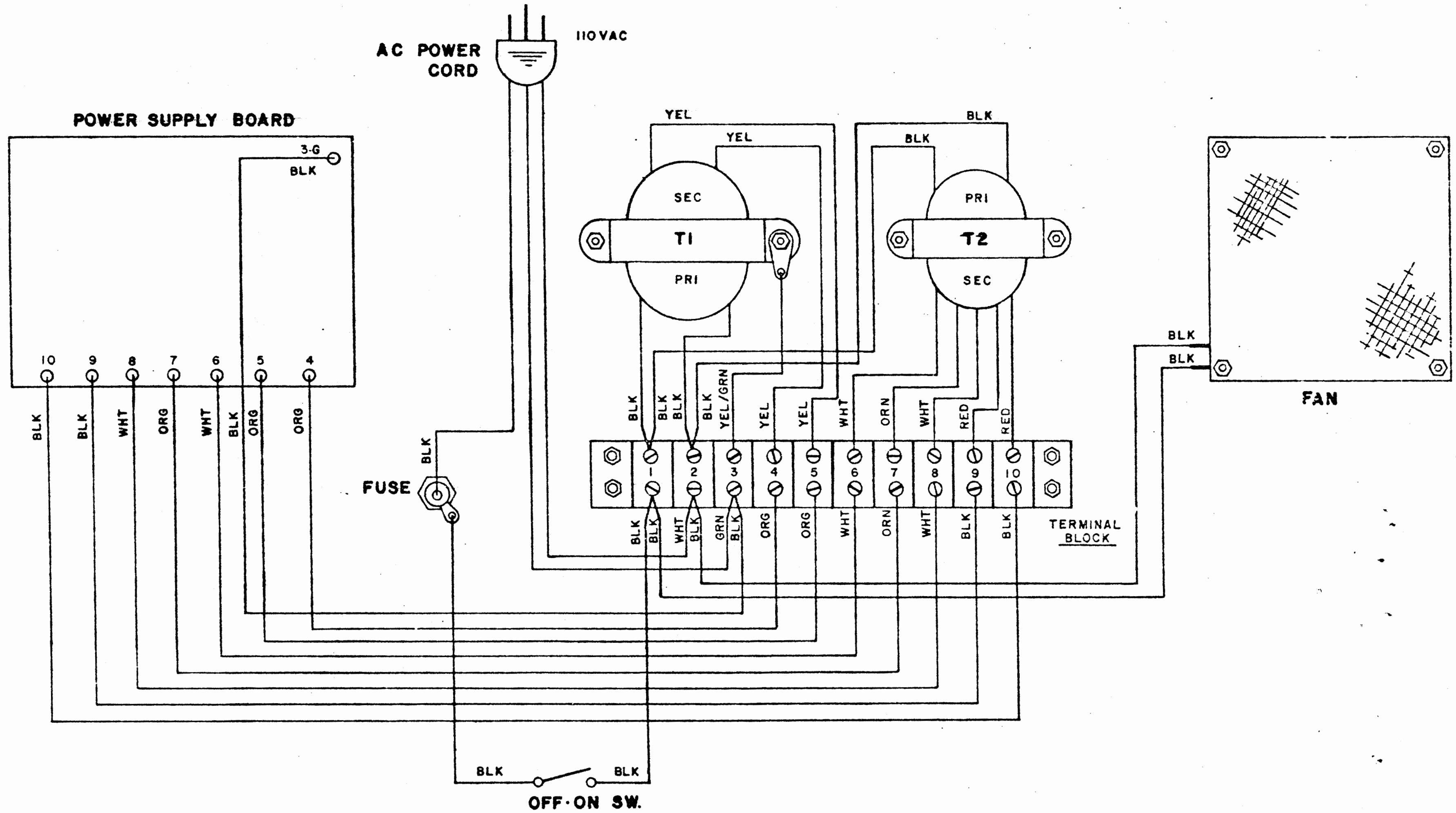
1) All of the wires should be connected to the pads on the board marked with the same designation as the tags placed on them earlier.

2) Insert all of the wires from the silk-screened side of the board, almost to the insulation. Add solder from the same side of the board except wire "3-G", and then continue applying heat while pushing the wires down as far as possible until the insulation just touches the solder. Be careful not to melt any insulation.

3) Turn the board over to solder wire "3-G" and then clip off all excess lead lengths.

Check this wiring over again carefully, and then remove the tags from the wires.





ALTAIR FLOPPY DISK DRIVE

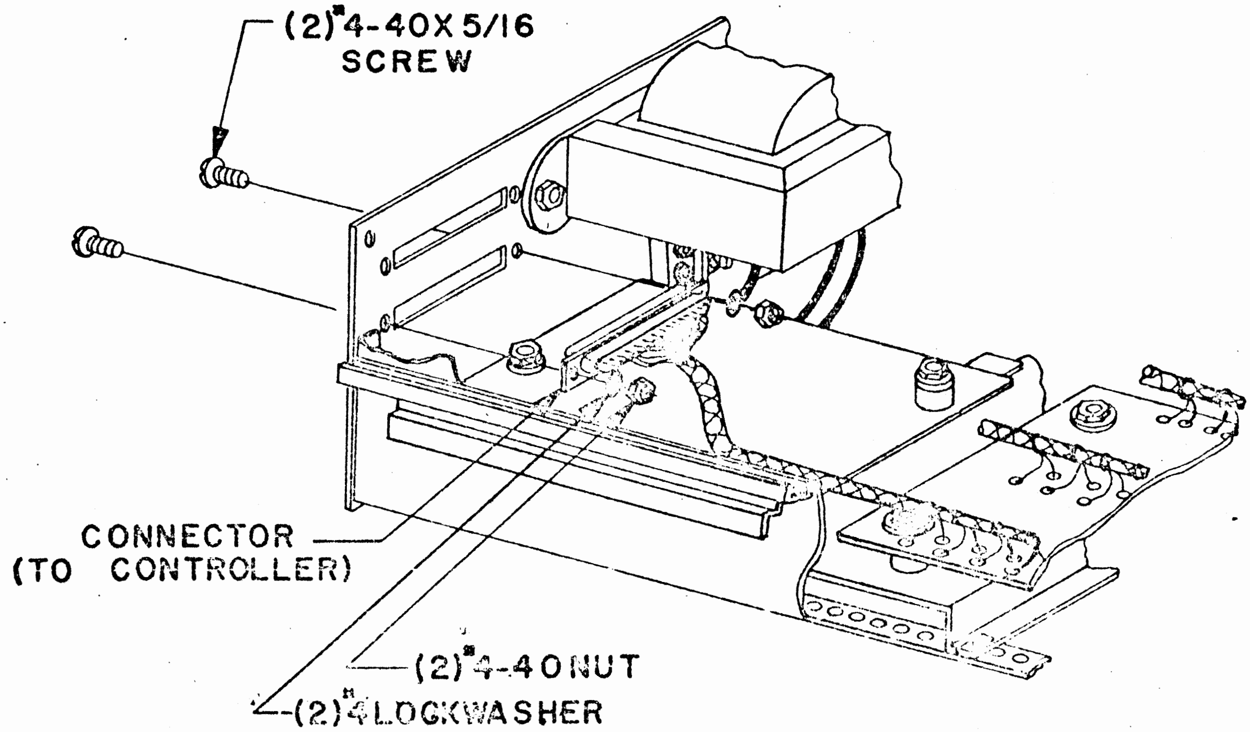
POWER SUPPLY WIRING DIAGRAM

CONNECTOR MOUNTING

Refering to the drawing below, mount the two 37-pin connectors to the back panel as shown.

Be sure to mount the male connector into the slot labeled "TO" and the female connector into the slot labeled "FROM".

On both connectors pin 1 should be towards the top.

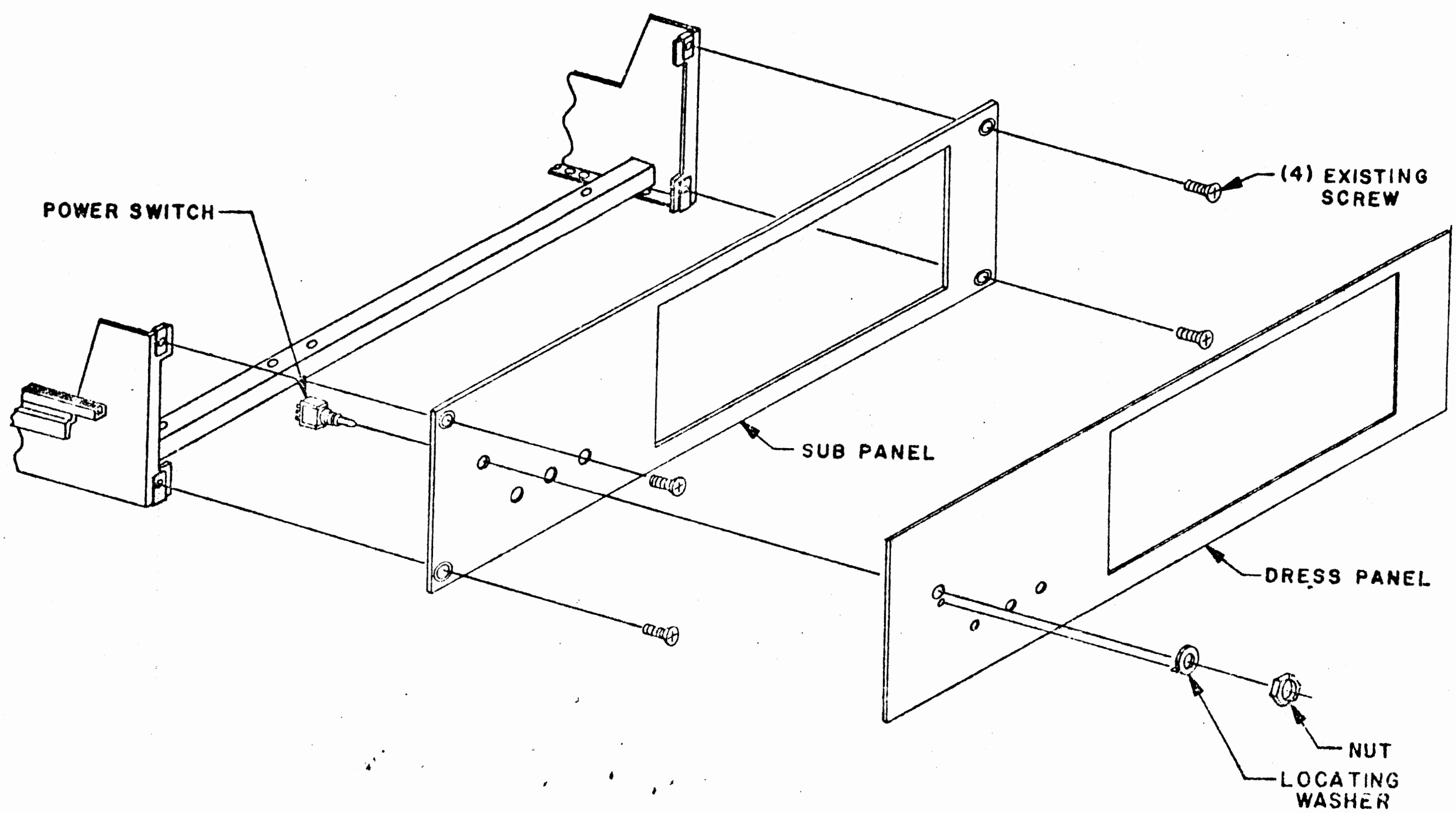


FRONT PANEL MOUNTING

The front sub panel and dress panel can now be re-installed. Use the same four screws previously used to mount the sub panel to re-mount it to the chassis as shown in the drawing below.

Note when setting the dress panel in place that it is a "floating" panel. Installing the power switch, as shown, at this time will temporarily hold it in place.

Be sure the lettering on the dress panel is facing outwards.



LED INSTALLATION

There are three RL-21 Light-Emitting-Diodes (LED's) to be installed on the Disk Buffer Board. These LED's have a cathode and anode lead on each of them which must be properly oriented for installation on the board. The diagram below shows you how to determine the cathode and anode leads of an RL-21. Hold the LED up to a light and you will be able to see inside. The larger of the two elements inside the plastic casing is the anode.

The silk-screen on the board itself has the cathode leads for the three LED's marked with a "K". The anode lead is marked with an "A". When you install these components, make sure that the cathode leads are in the pads marked "K" and the anode leads in the pads marked "A". Improper orientation when installing LED's may cause permanent damage to the component.

As is shown in the drawing on this page, these three components also require special spacing and bending of the leads in order to fit the unit properly.

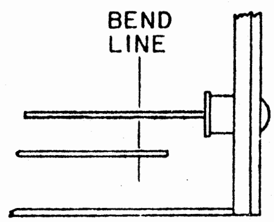
1) Set the LED's in place one at a time and bend as necessary to fit as shown in drawing [3].

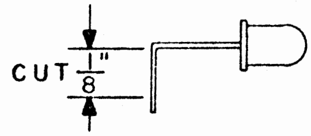
2) Cut the leads as shown in [2] and place the LED's on the board properly.

3) Solder them in place from the top side of the board. LED's are very heat sensitive, so use a minimum of heat for the shortest amount of time possible to make the connection.

When properly installed, the LED's should fit as shown in the drawing below.

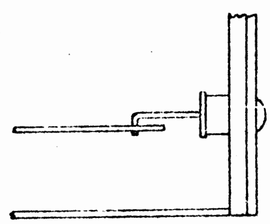
[1] SET THE LED IN PLACE AND MARK THE LEADS





[2] CUT THE EXCESS LEAD TO LEAVE 1/8 INCH

[3] SOLDER TO FIT IN PLACE AS SHOWN



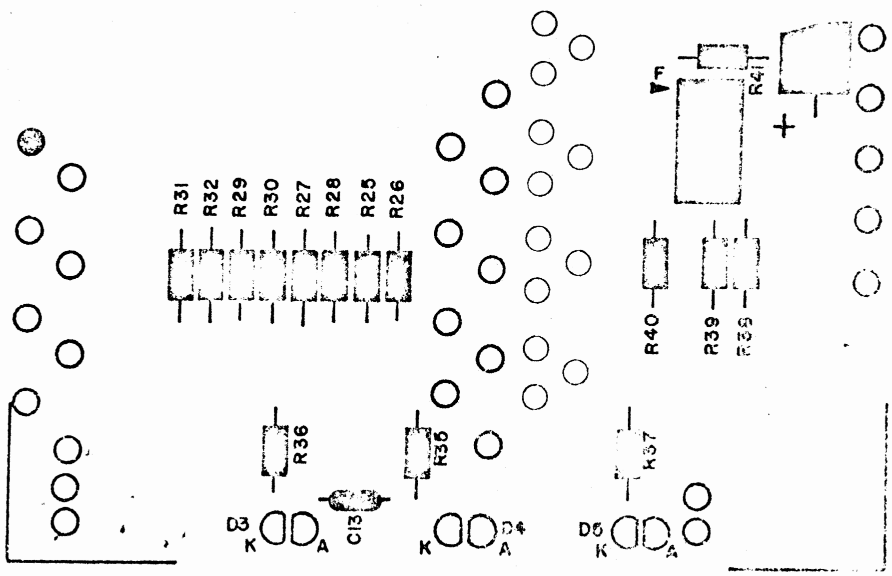
WARNING: RL-21 LED's are very sensitive to heat. Use a minimum application of heat with your iron when making these solder connections.

LED Installation

( ) D3 = RL-21 LED

( ) D4 = RL-21 LED

( ) D5 = RL-21 LED



DISK DRIVE UNIT INSTALLATION

The Disk Drive unit itself can now be installed into the chassis.

1) The first step in this process is to set the chassis on end, with the front panel facing upwards.

2) Remove the screws and rubber feet that were factory installed on the bottom of the drive unit.

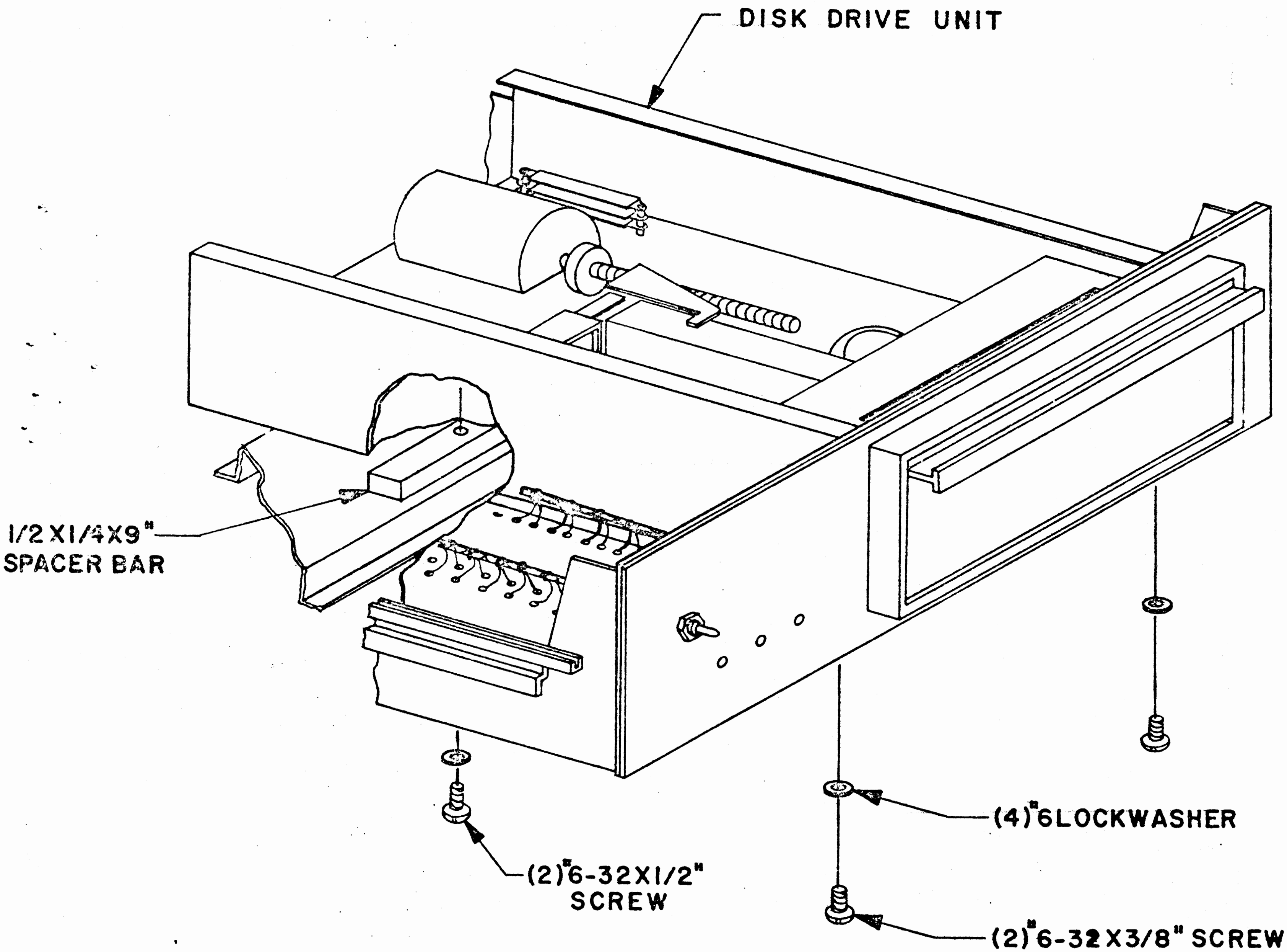
3) Being careful not to catch any of the wires or cables, slowly lower the drive unit into the chassis. Refer to the drawing on the following page for the proper orientation.

4) Refering again to the drawing on the following page, insert the two mounting screws and lockwashers on the front side of the drive unit. Do not tighten the screws down at this time.

5) Refering to the same drawing, install the spacer bar and mounting hardware for the rear end of the drive unit.

   Tighten all four mounting screws firmly.

6) The 44-pin edge connector should now be plugged into the rear of the drive unit. Line up the connector with the finger pads on the units PC board and align the plastic key between pins 5 & 6 with the slot in the board. Push the connector firmly into place.


DISK DRIVE UNIT
1/2 X1/4X9" SPACER BAR
(4) #6 LOCKWASHER
(2) #6-32X1/2" SCREW
(2) #6-32X3/8" SCREW

DISK DRIVE PRELIMINARY CHECK OUT

1. With no diskette in drive and the chassis unit not installed in cabinet, and no address jumpers installed, turn power on.

   A) Fan and disk drive motor should turn.

   B) Power indicator should light.

2. If voltmeter is available, measure:

   A) +24 volt supply at + end of C3 (with respect to chassis) on the power supply board.

   B) +5 volt supply at + end of C6 on the power supply board.

   C) -5 volt supply at point "J" of the power supply board.

   All voltages should be within 5% of rated output. If the disk drive motor does not start up, or the power indicator does not light, or the power supply voltages are wrong, consult the Theory of Operation and recheck wiring.

3. A) With a cliplead, ground to chassis wire #13 (Disk Enable) on the left edge of the buffer board (Pin 13 of "To Controller").

   The Disk Enable light should come on.

   B) Now open disk drive door. The drive motor should stop and Disk Enable light should turn off. Close the door and the motor should start up. 5-10 seconds later, the Disk Enable light should turn on (timing controlled by IC G).

   C) With another cup lead, test the mechanical disk functions by grounding (on the left edge of board)

      1. Wire #8 (Head Load)
         The Head Load solenoid should energize as long as #8 is grounded, and Head Load light should turn on.

      2. Wire #6 (Step In)
         The track stepping motor shaft should turn as point #6 is intermittantly grounded, simulating stepping pulses. The head carriage should move towards the front of the Disk Drive.

      3. Wire #7 (Step Out)
         The track stepping motor shaft should turn as Point #7 is intermittantly grounded, simulating stepping pulses. The head carriage should move towards the rear of the Disk Drive.

   This completes the preliminary check out of the Disk Drive.

   Remove the clip leads, and install the disk address jumpers as indicated on page 77 .

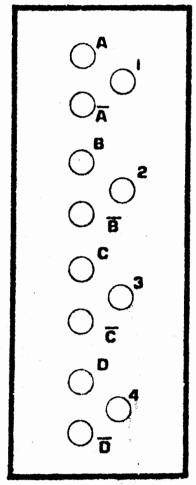
ADDRESS SELECTION

There are four jumper wires to be installed on the buffer board in order to select the I/0 address.

Use component leads saved earlier for this purpose. Install them from the silk-screened side of the board and solder them on either side.

To comply with MITS software, the board should be jumpered to address Ø unless it is a part of a multiple disk drive system.

Refering to the silk-screen drawing on the right, jumper as follows for address Ø:



PAD	TO	PAD
1	——	$\overline{A}$
2	——	$\overline{B}$
3	——	$\overline{C}$
4	——	$\overline{D}$

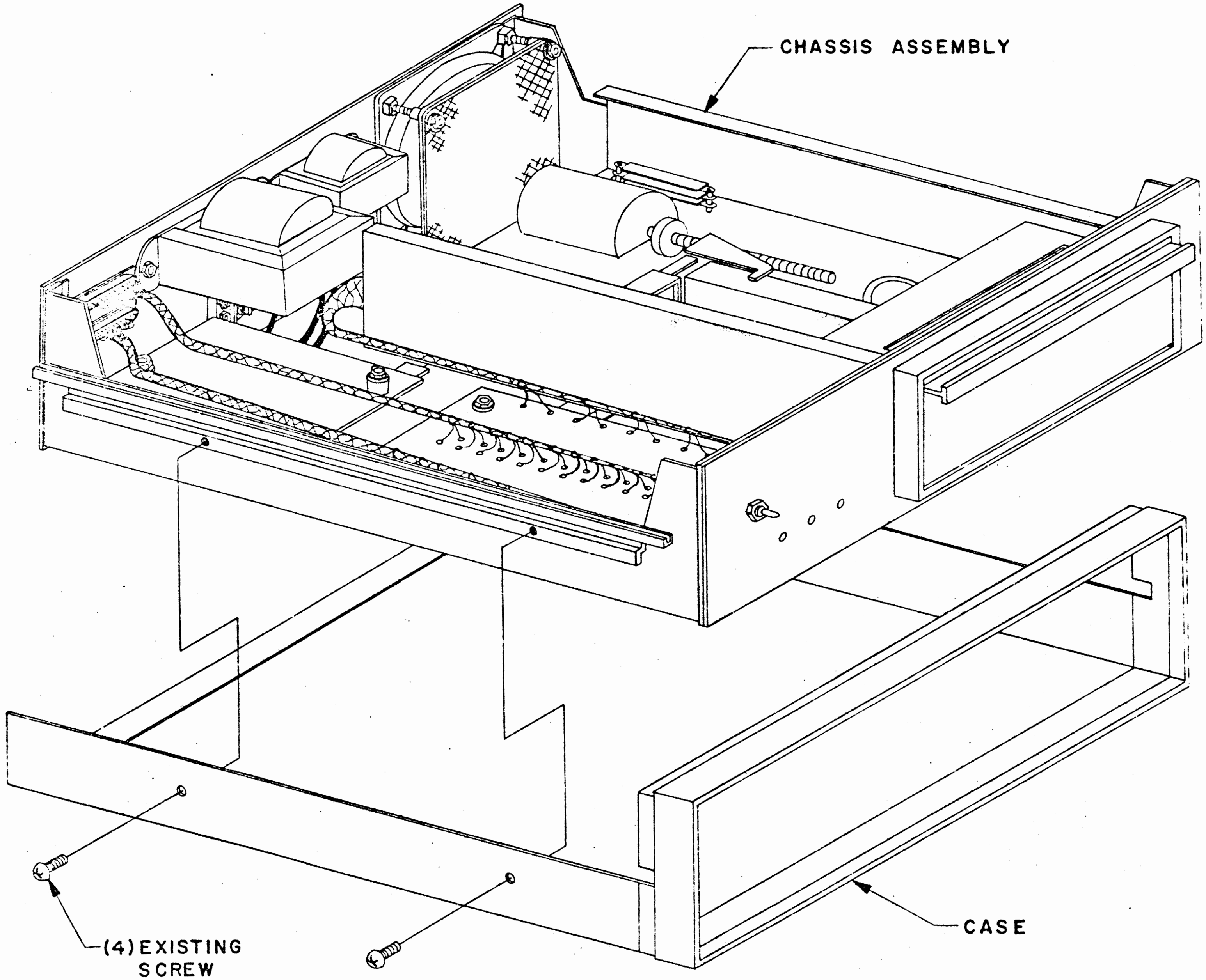
Consult the jumper chart in the Theory of Operation section if a different address is desired.

FINAL ASSEMBLY

The chassis assembly can now be installed into the outer case.

Refer to the drawing on the following page and mount the chassis as shown.

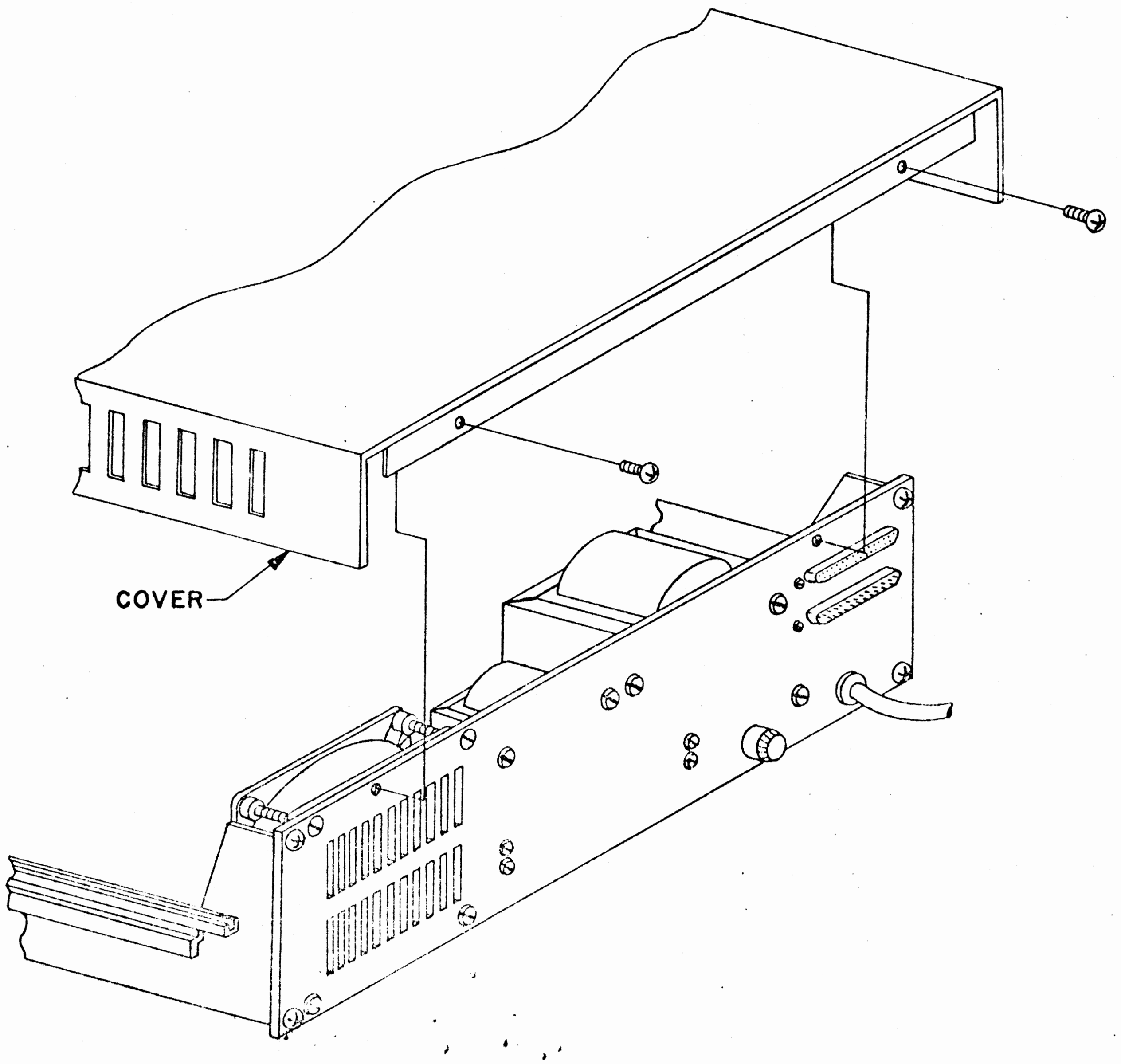
To insert it, start by setting it slightly towards the back of the case, and then slide it forward until the screw holes align. Tighten the four screws firmly.

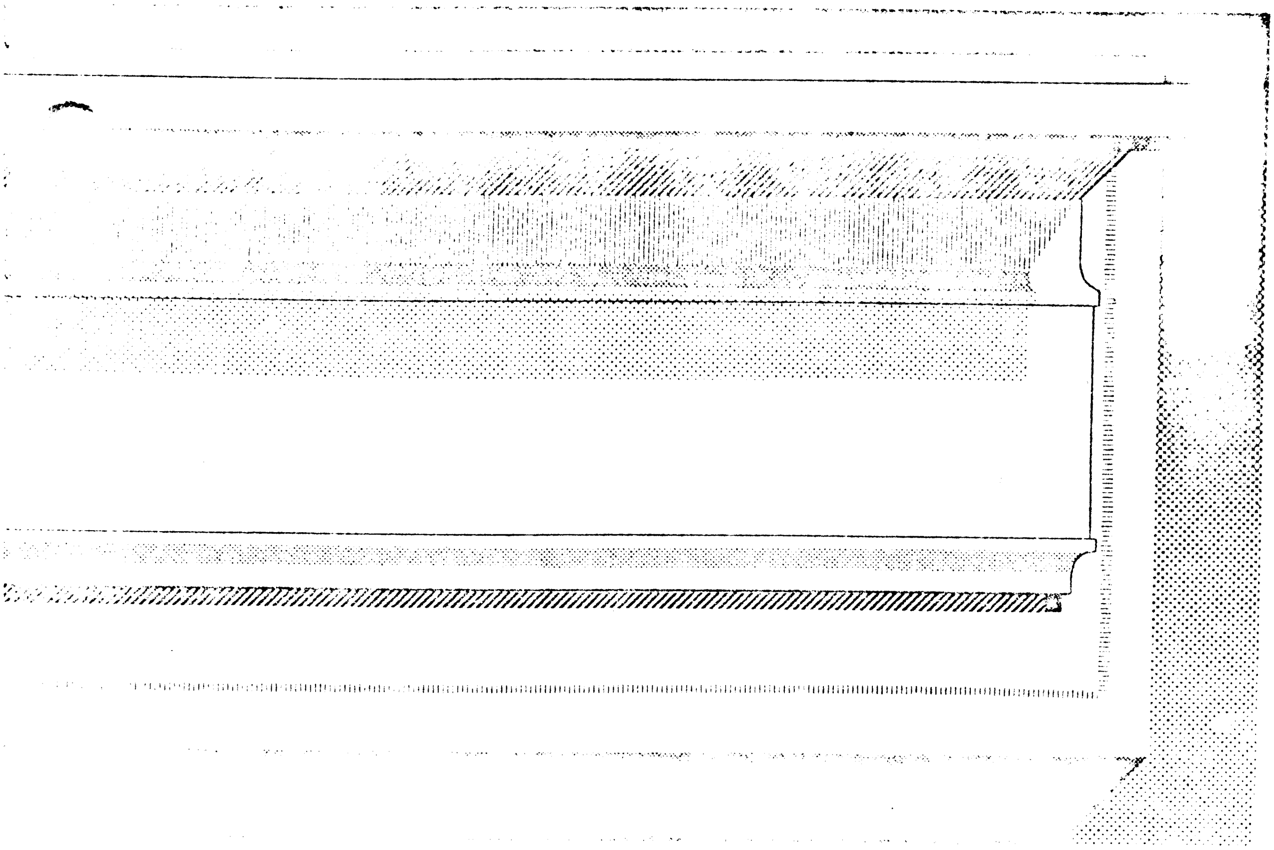

CHASSIS ASSEMBLY
CASE
(4) EXISTING SCREW

CASE TOP INSTALLATION

Re-install the case top onto the unit as shown below. Do not, however, use the same screws which held it originally.

Use #6-32 x 1/4" screws to secure the case top.





DISK CONTROLLER ASSEMBLY

The Disk Controller will now be assembled. This consists of two PC boards and interconnecting cables.

The Disk Controller mounts directly into the computer main-chassis and uses two slots.

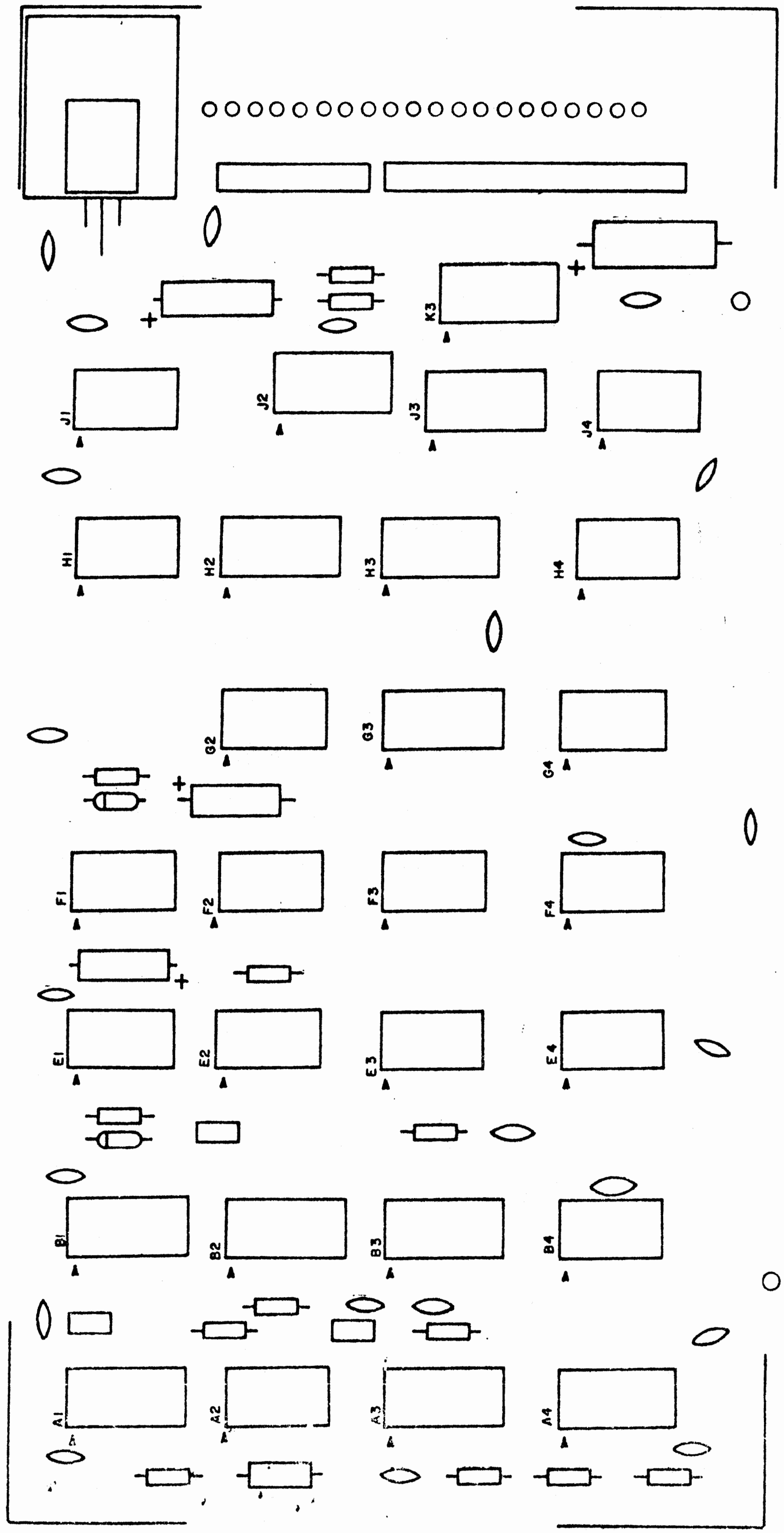
Controller Board #2 will be assembled first.

IC Installation

Install the following 28 ICs according to the instructions on page 4.

ICs

Silk Screen Designation	Number	Silk Screen Designation	Number
( ) A1	74123 ✓	( ) F3	74L02 ✓
( ) A2	74L73 ✓	( ) F4	74L02 ✓
( ) A3	93L16 ✓	( ) G2	74L04 ✓
( ) A4	93L16 ✓	( ) G3	74L75 ✓
( ) B1	74123 ✓	( ) G4	74L04 ✓
( ) B2	74123 ✓	( ) H1	74L02 ✓
( ) B3	74123 ✓	( ) H2	74166 ✓
( ) B4	74L04 ✓	( ) H3	74L75 ✓
( ) E1	74L00 ✓	( ) H4	74L04 ✓
( ) E2	74L73 ✓	( ) J1	74L02 ✓
( ) E3	74L00 ✓	( ) J2	8T98 ✓
( ) E4	74L10 ✓	( ) J3	74L75 ✓
( ) F1	74L02 ✓	( ) J4	74L74 ✓
( ) F2	74L73 ✓	( ) K3	8T97 ✓


K3
J1
J2
J3
J4
H1
H2
H3
H4
G2
G3
G4
F1
F2
F3
F4
E1
E2
E3
E4
B1
B2
B3
B4
A1
A2
A3
A4

Resistor Installation

Install the following 13 resistors according to the instructions on page 5 .

RESISTORS

( ) R1, Brown-Black-Orange, 1/4 or 1/2 W.

( ) R2, Brown-Black-Orange, 1/4 or 1/2 W.

( ) R3, Orange-White-Orange, 1/4 or 1/2 W.

( ) R4, Brown-Black-Orange, 1/4 or 1/2 W.

( ) R5, Brown-Green-Orange, 1/4 or 1/2 W.

( ) R6, Red-Red-Brown, 1/4 or 1/2 W.

( ) R7, Orange-Orange-Brown, 1/4 or 1/2 W.

( ) R8, Brown-Green-Orange, 1/4 or 1/2 W.

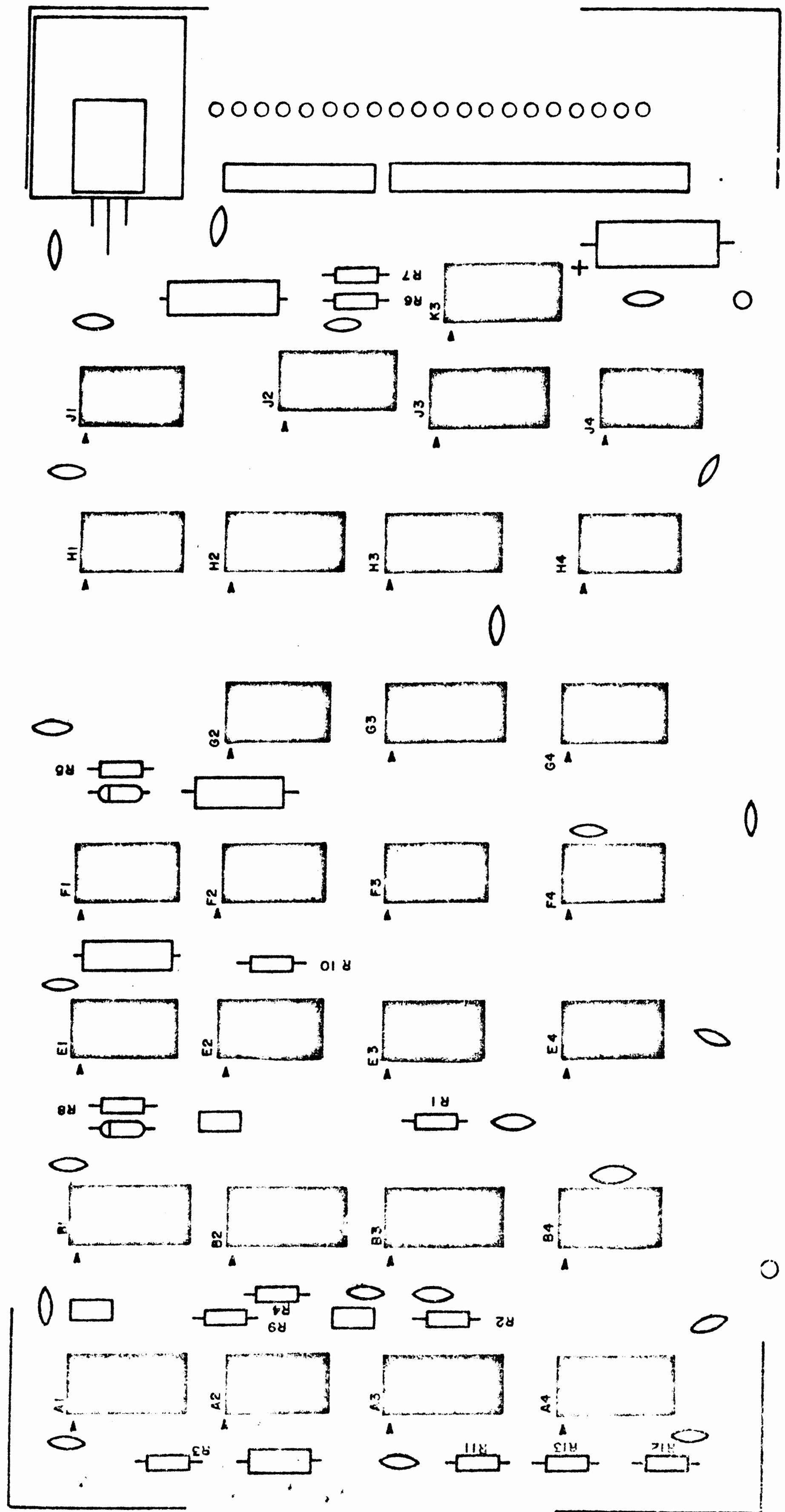
( ) R9, Blue-Gray-Red, 1/4 or 1/2 W.

( ) R10, Brown-Blue-Orange, 1/4 or 1/2 W.

( ) R11, Brown-Black-Red, 1/4 or 1/2 W.

( ) R12, Brown-Black-Red, 1/4 or 1/2 W.

( ) R13, Brown-Black-Red, 1/4 or 1/2 W.


R7
R6
K3
J1
J2
J3
J4
H1
H2
H3
H4
G2
G3
G4
R5
F1
F2
F3
F4
R 10
E1
E2
E3
E4
R8
R 1
B2
B3
B4
R4
R9
R2
A1
A2
A3
A4
R3
R11
R13
R12

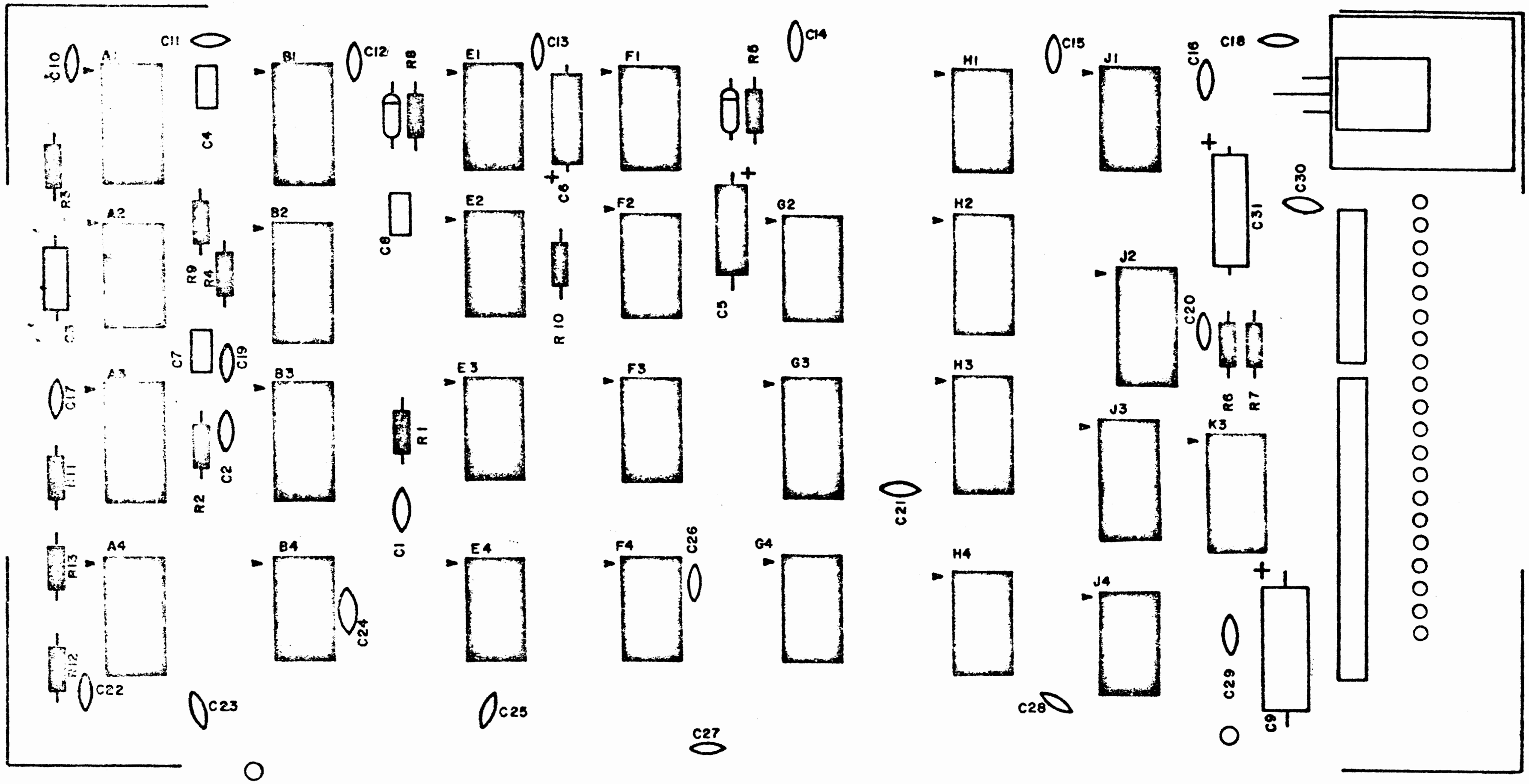
Capacitor Installation

Install the following 31 capacitors according to the instructions on page 6 . Note that all capacitors are installed in the same manner, except for electrolytic capacitors.

CAPACITORS

( ) C1, .001 uf
( ) C2, .001 uf
( ) C3, 1.0 uf
( ) C4, .22 uf
( ) C5, electrolytic, 4.7 uf
( ) C6, electrolytic, 10 uf
( ) C7, .1 uf
( ) C8, .1 uf
( ) C9, electrolytic, 35 uf
( ) C10, .1 uf
( ) C11, .1 uf
( ) C12, .1 uf
( ) C13, .1 uf
( ) C14, .1 uf
( ) C15, .1 uf
( ) C16, .1 uf
( ) C17, .1 uf
( ) C18, .1 uf
( ) C19, .1 uf
( ) C20, .1 uf
( ) C21, .1 uf
( ) C22, .1 uf
( ) C23, .1 uf
( ) C24, .1 uf
( ) C25, .1 uf
( ) C26, .1 uf
( ) C27, .1 uf
( ) C28, .1 uf
( ) C29, .1 uf
( ) C30, .1 uf
( ) C31, electrolytic, 35 uf

68


A1
A2
A3
A4
B1
B2
B3
B4
E1
E2
E3
E4
F1
F2
F3
F4
G2
G3
G4
H1
H2
H3
H4
J1
J2
J3
J4
K3
C1
C2
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
C14
C15
C16
C17
C18
C19
C20
C21
C22
C23
C24
C25
C26
C27
C28
C29
C30
C31
R1
R2
R3
R4
R6
R7
R8
R9
R10
R11
R12
R13

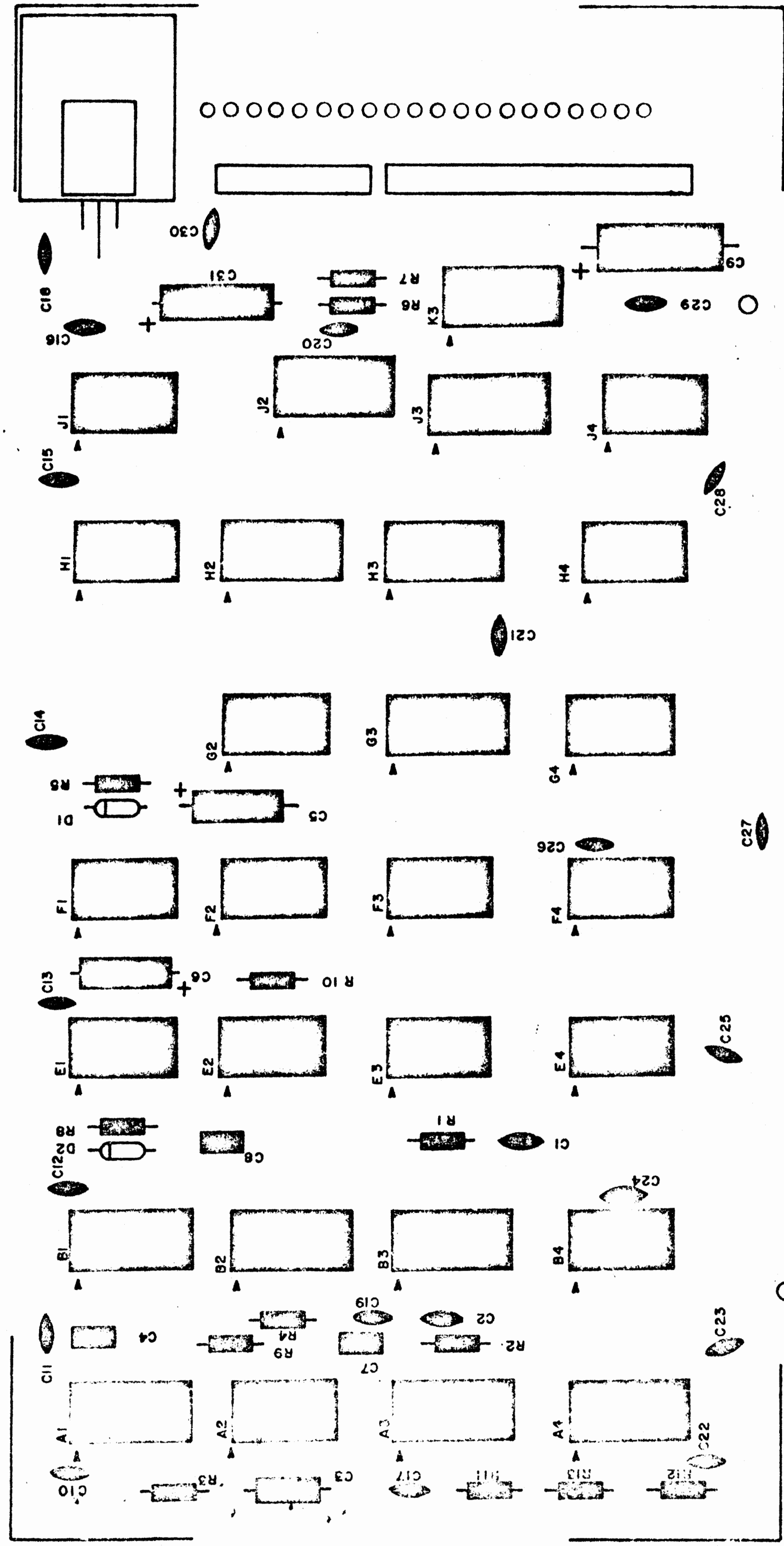
Diode Installation

Install the following two diodes according to the instructions on page 7 .

DIODES

( ) D1, 1N914

( ) D2, 1N914



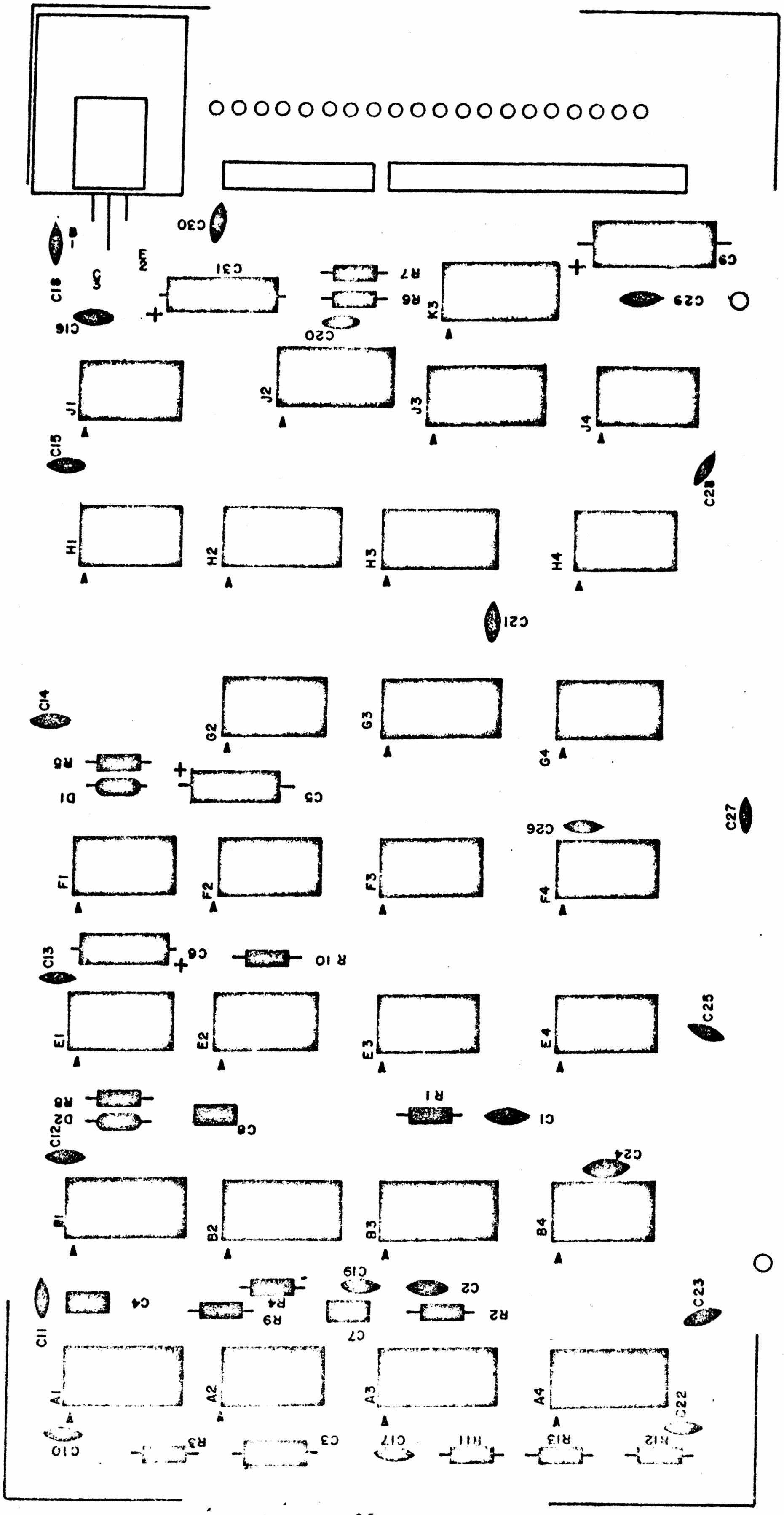
C30
C18
C31
R7
R6
K3
C9
C29
C16
C20
J1
J2
J3
J4
C15
C28
H1
H2
H3
H4
C21
C14
G2
G3
G4
R5
D1
C5
C27
C26
F1
F2
F3
F4
C13
C6
R 10
C25
E1
E2
E3
E4
R8
D2
C8
R 1
C1
C12
C24
B1
B2
B3
B4
C19
C2
C4
R4
R9
C7
R2
C23
C11
A1
A2
A3
A4
C22
C10
R3
C3
C17
R11
R13
R12

Voltage Regulator Installation

Install the voltage regulator according to the instructions on page 32 .

VOLTAGE REGULATOR

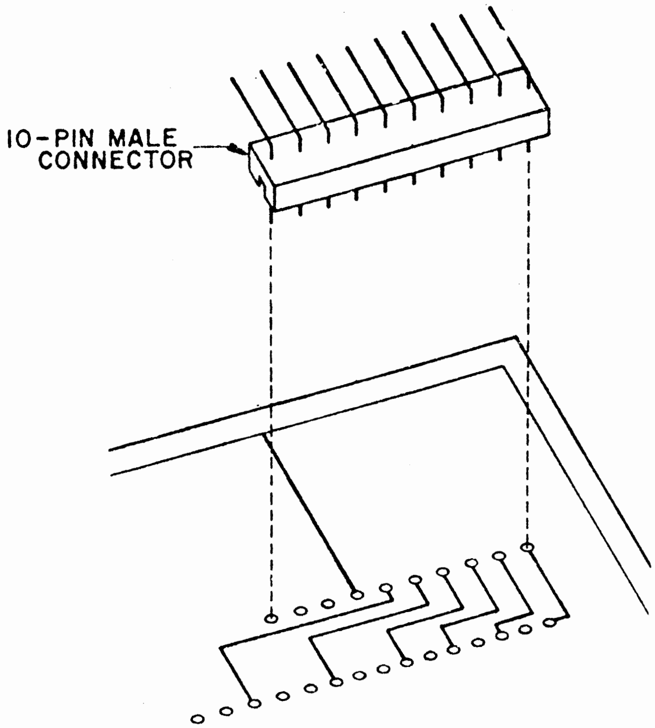
( ) 7805


C30
C18
C31
R7
R6
K3
C9
C29
C16
C20
J1
J2
J3
J4
C15
C28
H1
H2
H3
H4
C21
C14
G2
G3
G4
R5
D1
C5
C26
C27
F1
F2
F3
F4
C13
C6
R10
C25
E1
E2
E3
E4
R8
D2
C8
R1
C1
C12
C24
B1
B2
B3
B4
C19
C2
C11
C4
R9
R4
C7
R2
C23
A1
A2
A3
A4
C22
C10
R3
C3
C17
R11
R13
R12

Connector Installation

There are two "boxes" marked on the silk-screen. These are to indicate the positions for a 10-pin and a 20-pin male connector.

The drawing below illustrates the installation of a typical connector of this type.



Refering to the drawing, install the two male connectors onto the silk-screened side of the board. The long 90° bent pins should point towards the right side of the board. The 10-pin connector goes between "CC1" & "CC10"; while the 20-pin connector goes between "CD1" & "CD20".

Two pins should now be cut off. These are the 2nd pin from the top on the 10-pin connector, and the 4th from the top on the 20-pin connector. Cut them off right at the plastic body of the connector. (These pins are both labeled "KEY" on the silk-screen.)

There is a row of 20 pads along the right edge of the board labeled CB1 through CB20.

Remove 10 twisted-pairs of wire from an 8 inch length of ribbon cable. Leave the two wires in each pair twisted together. Strip 1/4 inch of insulation from both ends of all of the wires and tin the exposed portions.

Beginning with the bottom pad on the board, connect one of the twisted-pairs to pads CB1 & CB2. Continue up the row of pads, connecting a twisted-pair to each two pads as you go along.

NOTE: The twisted-pairs each have one wire the same color in each of them (usually black or white). Make the connection to pad CB1 with this wire on the 1st pair, and use this wire for the 1st connection on each of the following pairs as you go up the row of 20 pads.

Insert all of the wires from the silk-screened side of the board and solder them of the bottom side. Clip off any excess lead lengths.

Cut the free ends of all 20 wires so that only 1/8 inch of tinned wire is exposed beyond the insulation.

A 20-pin female connector will now be attached to the free ends of the 20 wires.

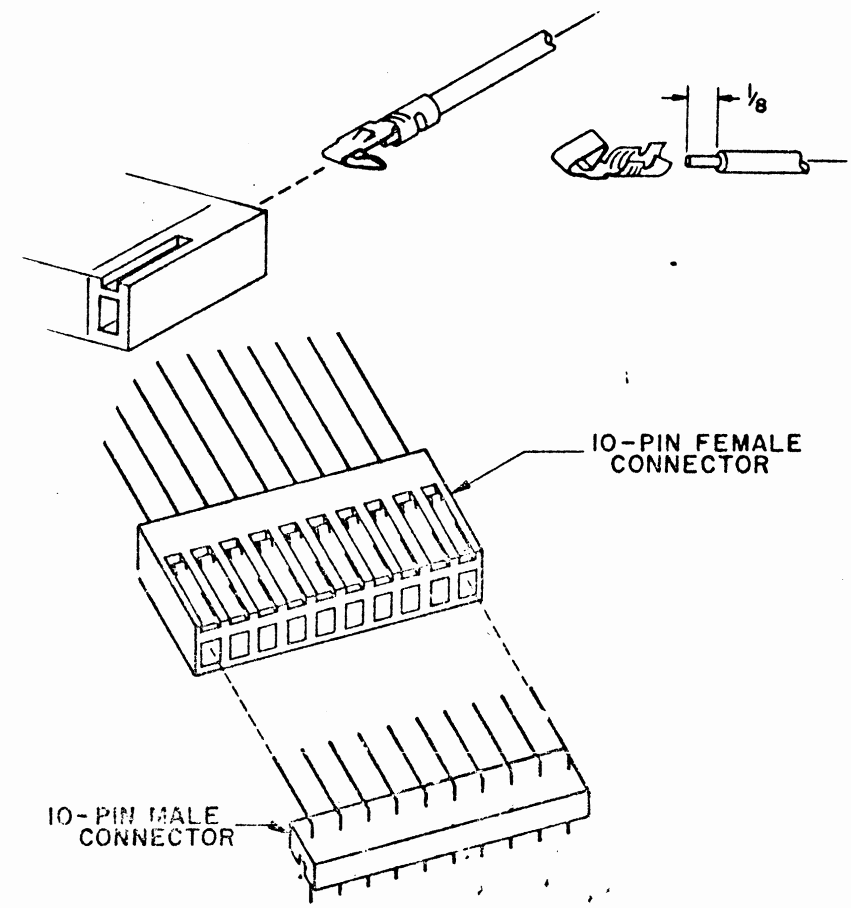
First, connector pins must be attached to the ends of all of the wires. The drawings below illustrate a typical connector of this type, and the method for attaching and inserting the pins.

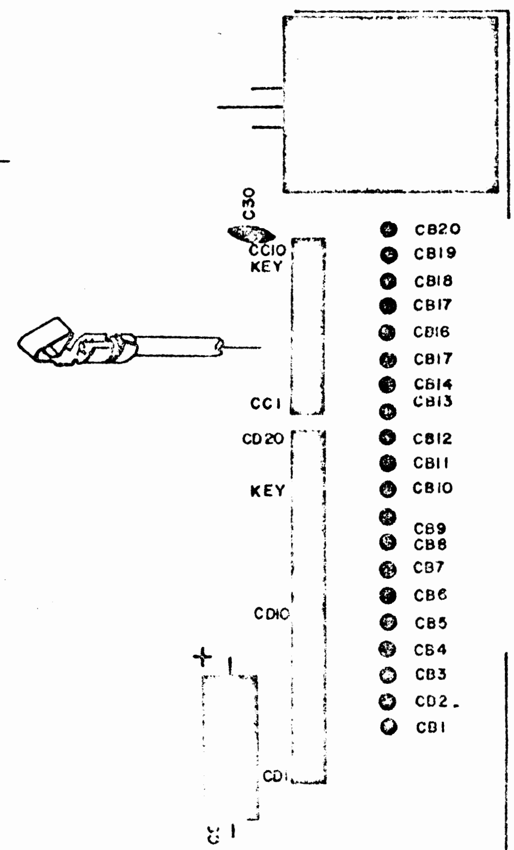
Connect a pin to each of the wires* as shown, and solder them carefully into place. Do not use too much solder or the pins will not fit into the connector properly.

NOTE: Two of the wires, both labeled CB17 on the PC board (see silk-screen), should be attached to a single pin.

Pins 1 & 20 are marked on the plastic body of the female connector. Refering to the silk-screen, insert the pins into the connector so that pad CB1 goes to pin 1, CB2 to pin 2, CB3 to pin 3, etc., being sure not to insert any wires into pin 15 on the connector. A plastic key should be inserted into pin 15 of the female connector, inserting it from the opposite side as the wires.

Place a tie-wrap approximately in the center between the connector and the board to hold the wires together. Place another tie-wrap around the wires and also through the holes in the PC board just to the right of the 20 pads.





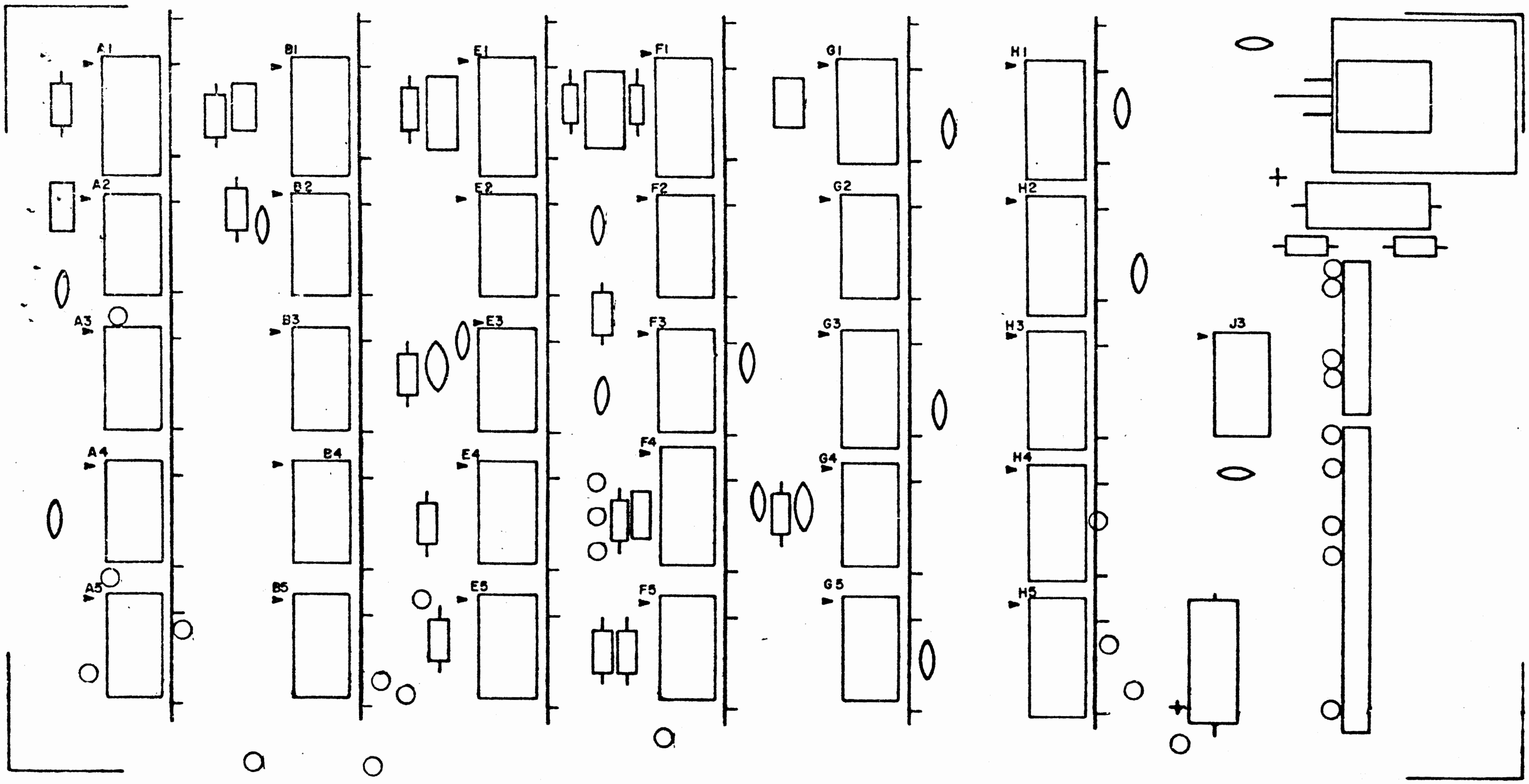
Controller Board #1 Assembly

IC Installation

Install the following 31 ICs according to the instructions on page 4 .

ICs

Silk Screen Designation	Number	Silk Screen Designation	Number
( ) A1	74123	( ) F2	74L73
( ) A2	74L02	( ) F3	74L73
( ) A3	74L20	( ) F4	74123
( ) A4	74L10	( ) F5	74L30
( ) A5	74L10	( ) G1	74164
( ) B1	93L16	( ) G2	74L00
( ) B2	74L74	( ) G3	74L75
( ) B3	74L73	( ) G4	7493
( ) B4	74L11	( ) G5	74L04
( ) B5	74L04	( ) H1	74L75
( ) E1	74123	( ) H2	8T97
( ) E2	74L00	( ) H3	8T97
( ) E3	74L73	( ) H4	8T97
( ) E4	74L04	( ) H5	8T97
( ) E5	74L00	( ) J3	74L04
( ) F1	74123		


A1
B1
E1
F1
G1
H1
A2
B2
E2
F2
G2
H2
A3
B3
E3
F3
G3
H3
J3
A4
B4
E4
F4
G4
H4
A5
B5
E5
F5
G5
H5

Resistor Installation

Install the following 16 resistors according to the instructions on page 5 .

RESISTORS

( ) R1, Orange-Orange-Brown, 1/4 or 1/2 W.

( ) R2, Red-Red-Brown, 1/4 or 1/2 W.

( ) R3, Brown-Black-Orange, 1/4 or 1/2 W.

( ) R4, Red-Black-Orange, 1/4 or 1/2 W.

( ) R5, Brown-Black-Orange, 1/4 or 1/2 W.

( ) R6, Red-Black-Orange, 1/4 or 1/2 W.

( ) R7, Green-Blue-Red, 1/4 or 1/2 W.

( ) R8, Brown-Black-Orange, 1/4 or 1/2 W.

( ) R9, Orange-Orange-Brown, 1/4 or 1/2 W.

( ) R10, Red-Red-Brown, 1/4 or 1/2 W.

( ) R11, Brown-Black-Orange, 1/4 or 1/2 W.

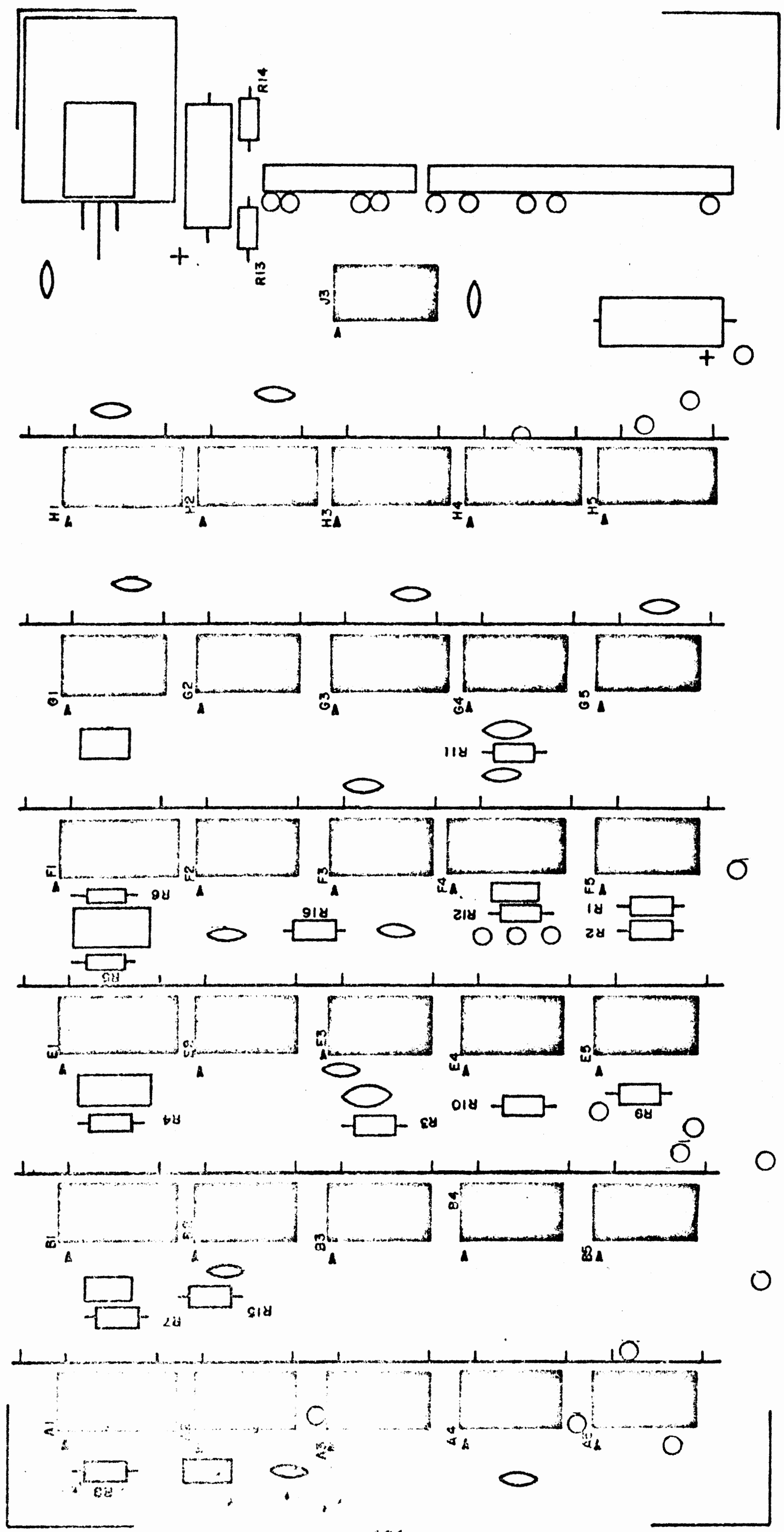
( ) R12, Red-Black-Orange, 1/4 or 1/2 W.

( ) R13, Red-Red-Brown, 1/4 or 1/2 W.

( ) R14, Orange-Orange-Brown, 1/4 or 1/2 W.

( ) R15, Brown-Black-Red, 1/4 or 1/2 W.

( ) R16, Brown-Black-Red, 1/4 or 1/2 W.


R14
R13
J3
H1
H2
H3
H4
H5
G1
G2
G3
G4
G5
R11
F1
F2
F3
F4
F5
R6
R16
R12
R1
R2
R5
E1
E3
E4
E5
R4
R3
R10
R9
B1
B3
B4
B5
R7
R15
A1
A3
A4
A5
R8

Capacitor Installation

Install the following 25 capacitors according to the instructions on page 6 . Note that all capacitors are installed in the same manner, except for electrolytic capacitors.

CAPACITORS

( ) C1, .1 uf

( ) C2, .68 uf

( ) C3, .047 uf

( ) C4, .68 uf

( ) C5, 430 pf

( ) C6, 910 pf

( ) C7, electrolytic, 33 uf

( ) C8, .01 uf

( ) C9, .047 uf

( ) C10, .1 uf

( ) C11, .1 uf

( ) C12, .1 uf

( ) C13, .1 uf

( ) C14, .1 uf

( ) C15, .1 uf

( ) C16, .1 uf

( ) C17, .1 uf

( ) C18, .1 uf

( ) C19, .1 uf

( ) C20, .1 uf

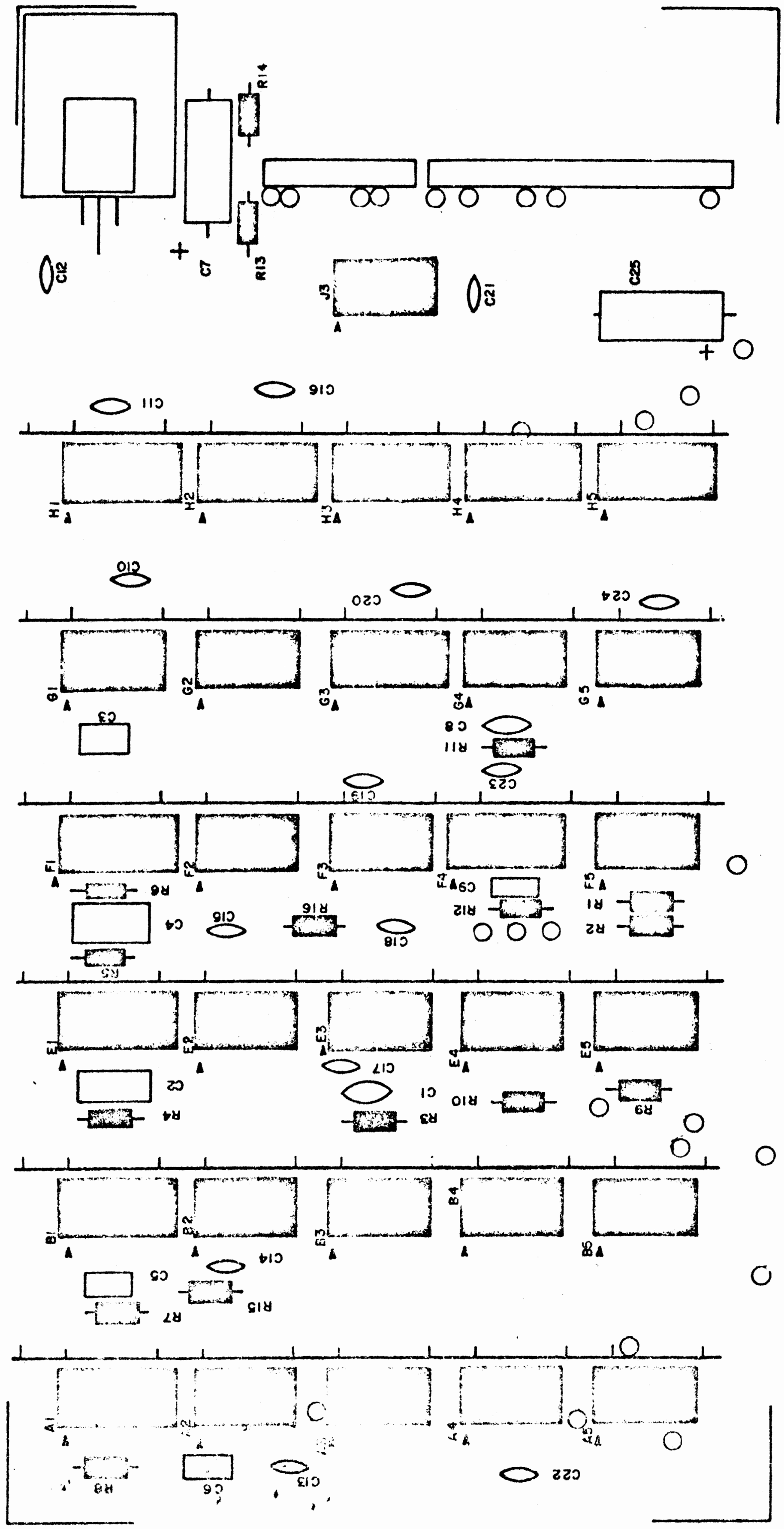
( ) C21, .1 uf

( ) C22, .1 uf

( ) C23, .1 uf

( ) C24, .1 uf

( ) C25, electrolytic, 35 uf

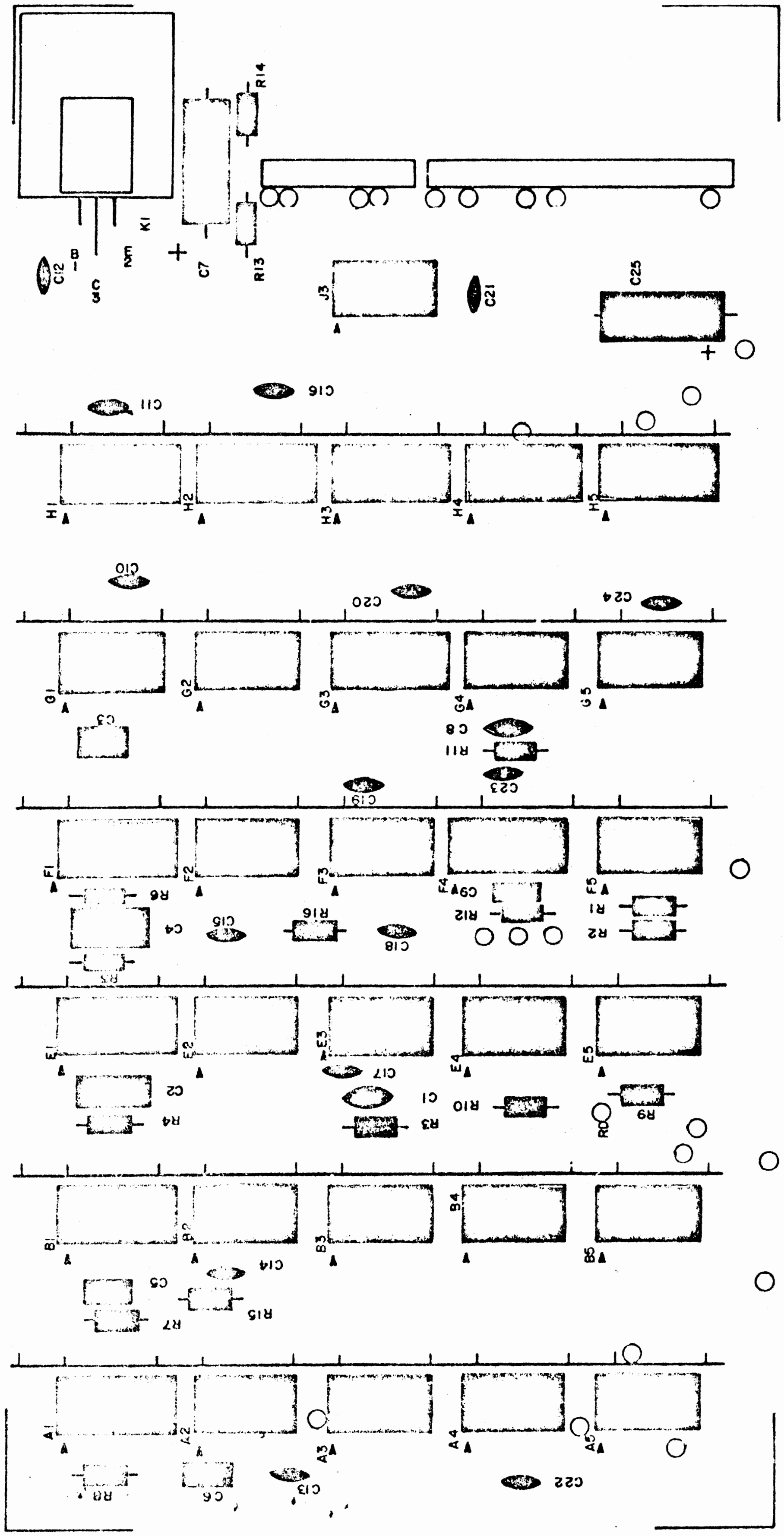

R14
R13
C7
C12
J3
C21
C25
C11
C16
H1
H2
H3
H4
H5
C10
C20
C24
G1
G2
G3
G4
G5
C3
C8
R11
C23
C19
F1
F2
F3
F4
F5
C9
R12
R1
R2
R6
C4
C15
R16
C18
R5
E1
E2
E3
E4
E5
C2
C17
C1
R10
R9
R4
R3
B1
B2
B3
B4
B5
C14
C5
R15
R7
A1
A2
A3
A4
A5
R8
C6
C13
C22

Voltage Regulator Installation

Install the voltage regulator according to the instructions on page 32.

VOLTAGE REGULATOR

( ) K1, 7805


K1
C12
C7
R13
R14
J3
C21
C25
C11
C16
H1
H2
H3
H4
H5
C10
C20
C24
G1
G2
G3
G4
G5
C3
C8
R11
C23
C19
F1
F2
F3
F4
F5
R6
C4
C15
R16
C18
R12
C9
R1
R2
R5
E1
E2
E3
E4
E5
C17
C2
C1
R10
R9
R4
R3
B1
B2
B3
B4
B5
C14
C5
R15
R7
A1
A2
A3
A4
A5
R8
C6
C13
C22

Jumper Installation

There are 13 jumper wires to be installed on board #1.

Install these jumper wires by inserting them on the silk-screened side of the board and soldering them on the back side. Clip off any excess lead length.

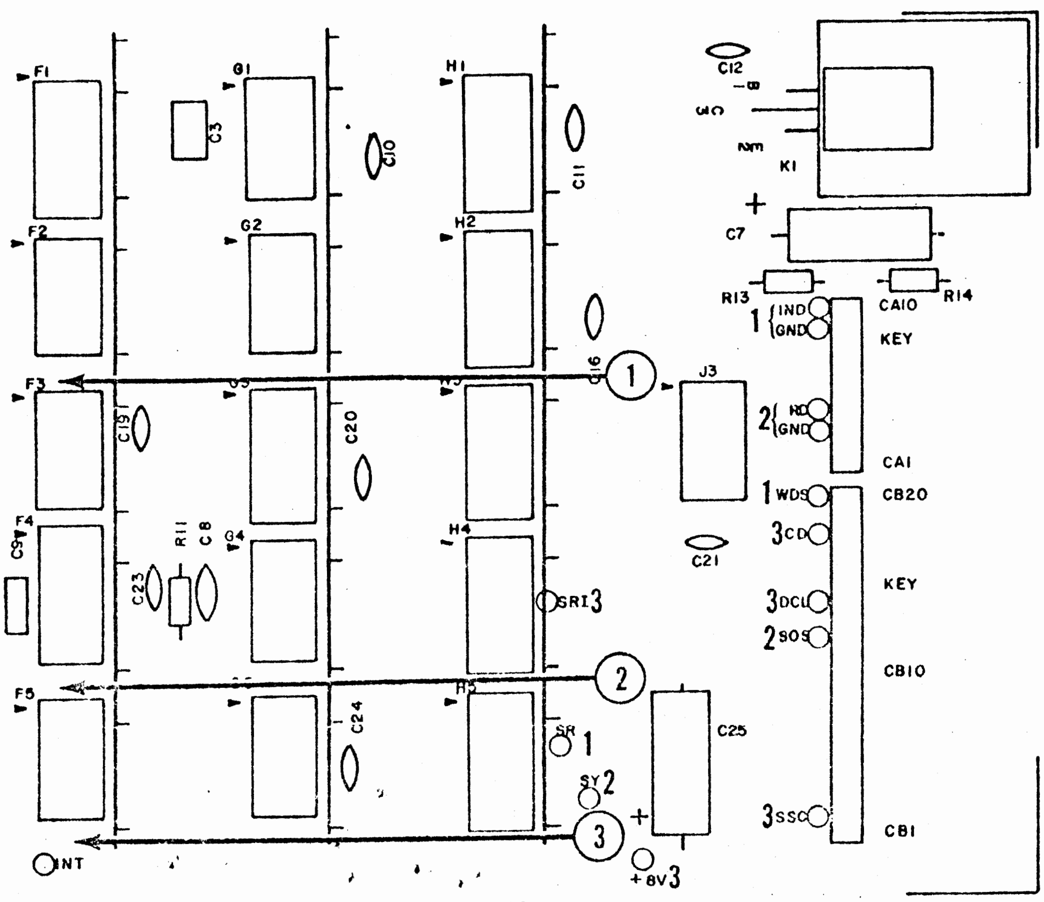
The drawing below shows the proper way to route the wires across the board. Pay close attention to this as it is very important. Pads labeled 1 below route through arrow 1, 2 through 2, and 3 through 3.

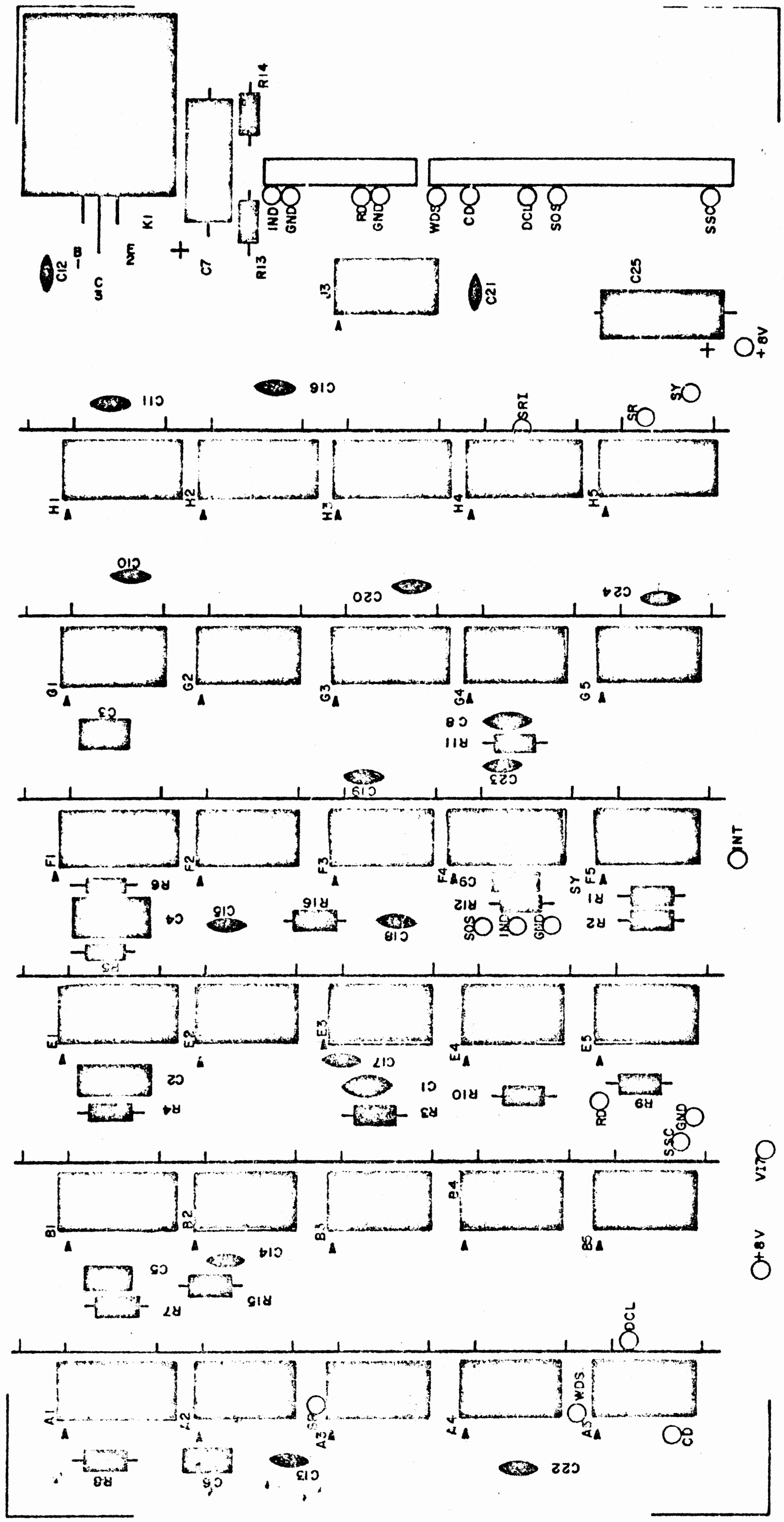
Cut the wires to the necessary length, and install them through the paths as shown. Use ribbon cable wires for the two twisted pair connections. The "GND" pad for the twisted pairs is the one closest to the other connection stated.

Connect the following jumpers:

IND to IND
GND to GND
RD to RD
GND to GND
WDS to WDS
CD to CD
DCL to DCL
SOS to SOS
SSC to SSC
+8V to +8V
SY to SY
SR to SR
SRI to INT*

*or to VI7 (see Theory of Operation)

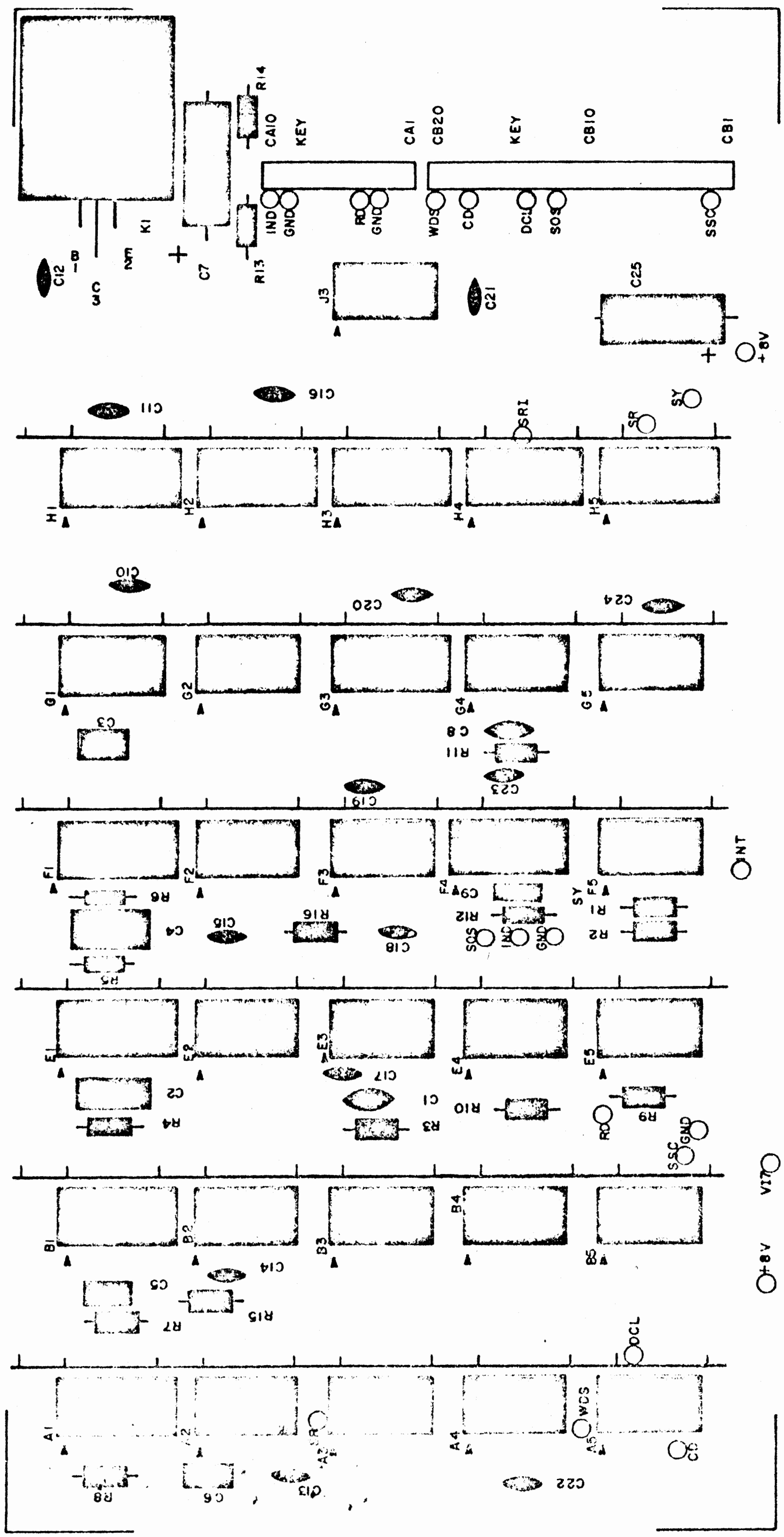



K1
C7
R13
R14
C12
IND
GND
RD
GND
WDS
CD
DCL
SOS
SSC
J3
C21
C25
+8V
C11
C16
SRI
SR
SY
H1
H2
H3
H4
H5
C10
C20
C24
G1
G2
G3
G4
G5
C3
C8
R11
C23
C19
F1
F2
F3
F4
F5
INT
R6
C4
C15
R16
C18
C9
R12
SOS
IND
GND
SY
R1
R2
R5
E1
E2
E3
E4
E5
C2
C17
C1
R10
R9
R4
R3
RD
GND
SSC
VIT
B1
B2
B3
B4
B5
+8V
C14
C5
R15
R7
DCL
A1
A2
A3
A4
A5
SR
WDS
CD
R8
C6
C13
C22

Connector Installation

Install a 10-pin and a 20-pin female connector onto the board in the same manner as described on page 94 for board #2.

NOTE: The only exception to the above statement is that pin 6 is to be cut off instead of pin 4 on the 20-pin connector.


R14
CA10
KEY
CA1
CB20
KEY
CB10
CB1
IND
GND
RD
GND
WDS
CD
DCL
SOS
SSC
K1
C12
C7
R13
J3
C21
C25
+8V
C11
C16
SR1
SR
SY
H1
H2
H3
H4
H5
C10
C20
C24
G1
G2
G3
G4
G5
C3
C8
R11
C23
C19
F1
F2
F3
F4
F5
INT
R6
C4
C15
R16
C18
C9
R12
SOS
INC
GND
R1
R2
R5
E1
E2
E3
E4
E5
C2
C17
C1
R10
R9
R4
R3
RD
SSC
GND
V17
B1
B2
B3
B4
B5
C14
C5
R15
R7
+8V
DCL
A1
A2
A3
A4
A5
WDS
CD
R8
C6
C13
C22

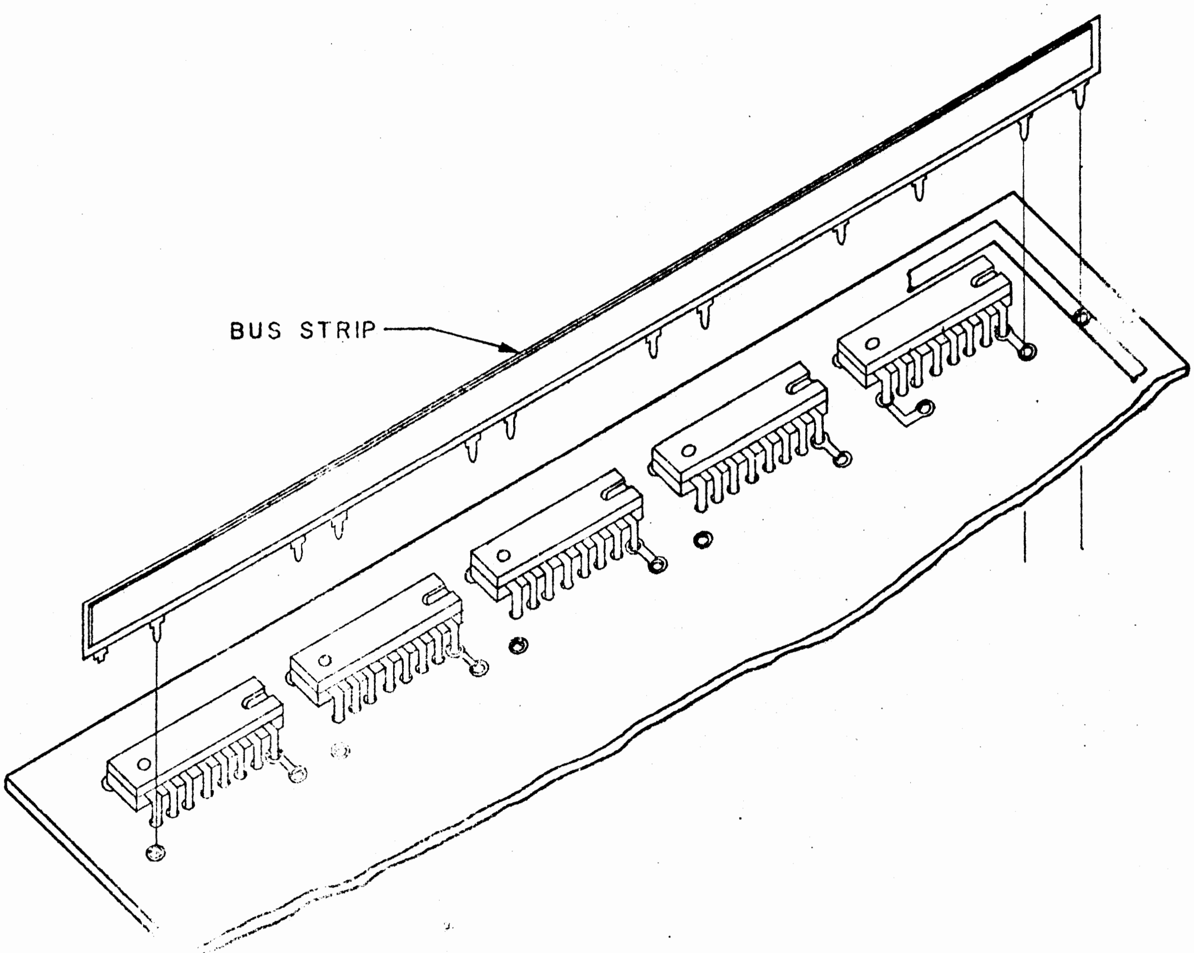
Bus Strip Installation

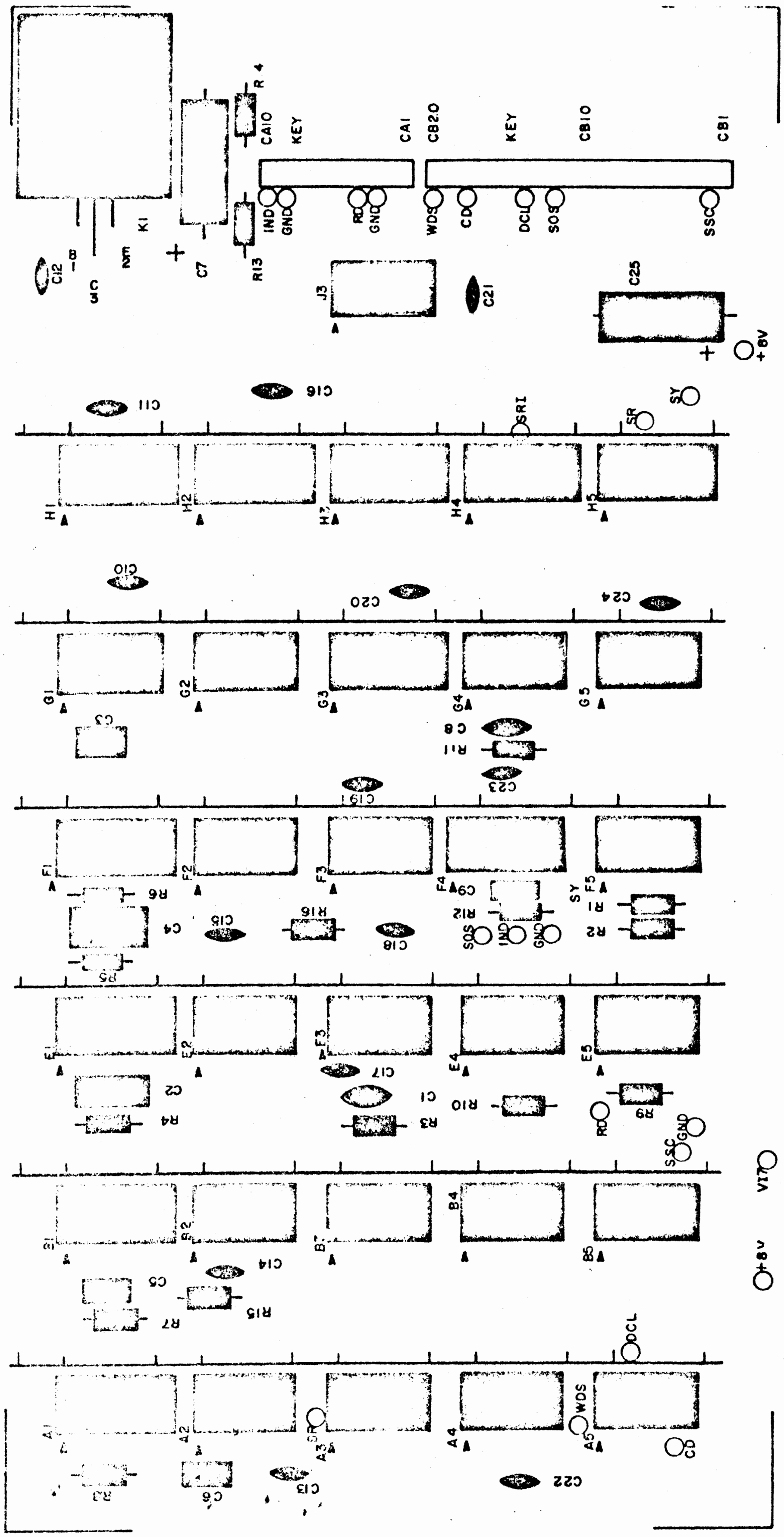
The drawing below illustrates the method for installing the 6 bus strips onto the board.

Note that the last pin (on the bottom side of the board) is to be cut off before installing the strips.

Be careful when installing these strips, that you do not push the strips down tight enough to damage the jumper wires or to short any of the PC lands.

Insert them as shown below and solder them on the non-silk-screened side of the board.




CA10
KEY
CA1
CB20
KEY
CB10
CB1
IND
GND
RD
GND
WDS
CD
DCL
SOS
SSC
R 4
R13
C7
K1
C12
J3
C21
C25
+8V
C11
C16
SRI
SR
SY
H1
H2
H4
H5
C10
C20
C24
G1
G2
G3
G4
G5
C3
C 8
R11
C19
C23
F1
F2
F3
F4
F5
R6
C4
C15
R16
C18
C9
R12
R2
R5
E1
E2
E4
E5
C2
C17
C1
R10
R4
R3
R9
V17
B2
B4
B5
C14
C5
R15
R7
A1
A2
A4
A5
C22
C6
C13

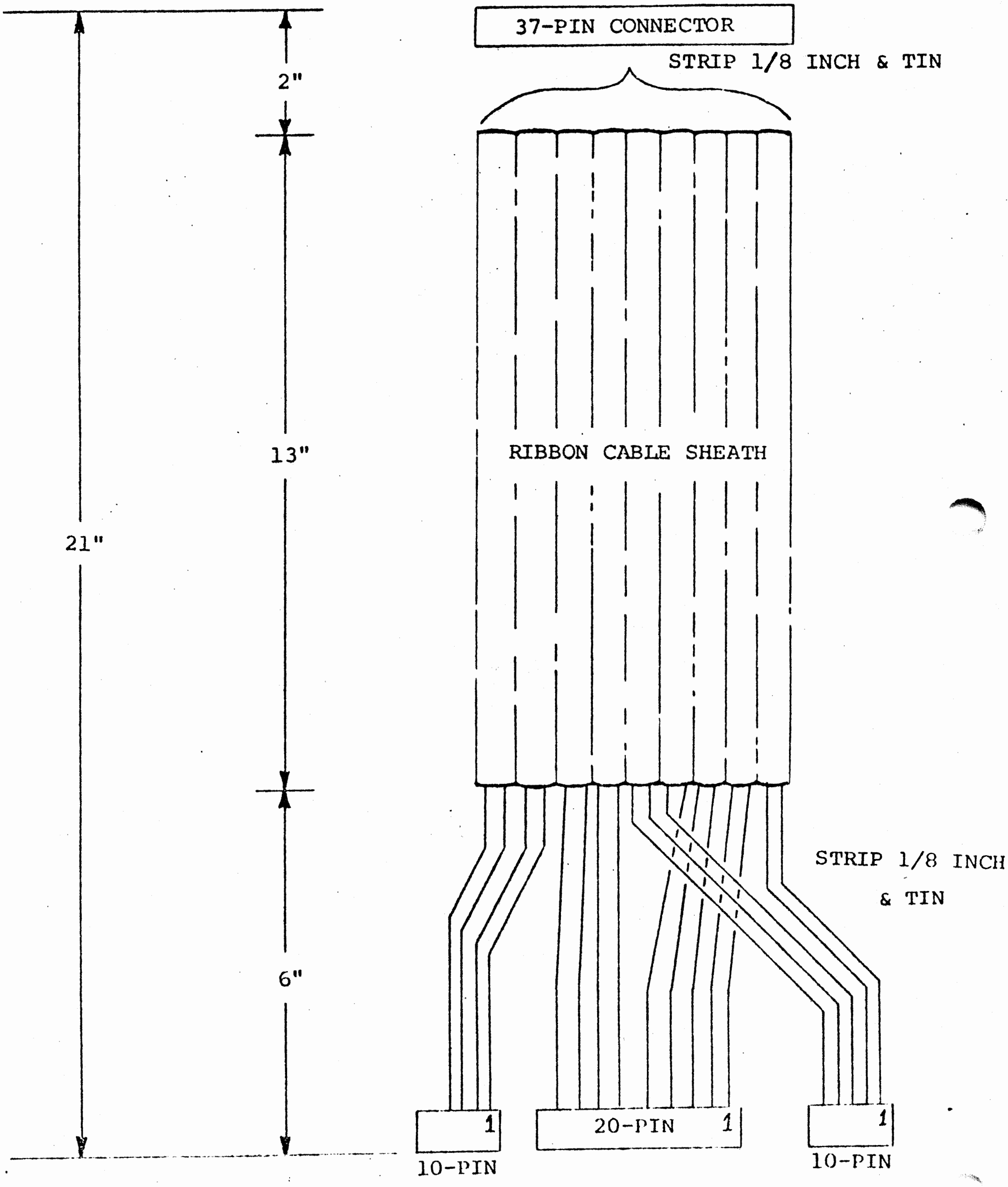
Controller Cable Assembly

Refering to the drawing on the following page, and to the previous instructions beginning on page 44, cut a 21 inch length of ribbon cable and prepare it as shown in the drawing.

The 37-pin connector shown at the top of the drawing is one of the FEMALE connectors included with your kit. The 10 & 20 pin connectors shown at the bottom of the drawing are of the same type as that on page 97 (female connectors), and should be assembled in the same manner.

Use the drawing on the following page, and the chart and drawing following after that, to construct this cable in the same relative manner as the previous ribbon cables.

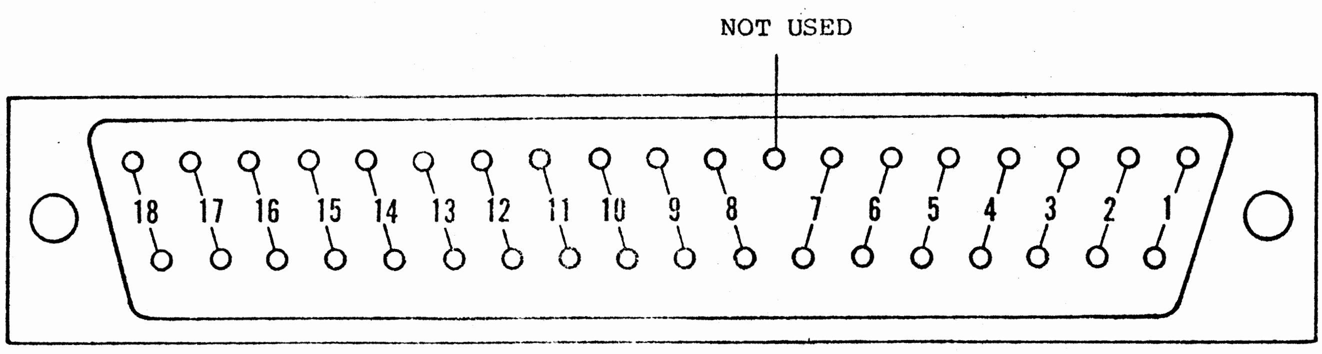
DISK CONTROLLER CABLE



The drawing below illustrates the pin positions where each of the 18 twisted-pairs should be attached to the 37-pin connector. Be sure to use a female connector. This portion of the assembly is essentially identical with that shown on page 51.

Use the orientation for this process shown on page 113. It would be adviseable to connect the varied colored wires from each pair to pins 1 through 19, and the same colored wire from each pair to pins 20 through 37.

37-PIN FEMALE CONNECTOR

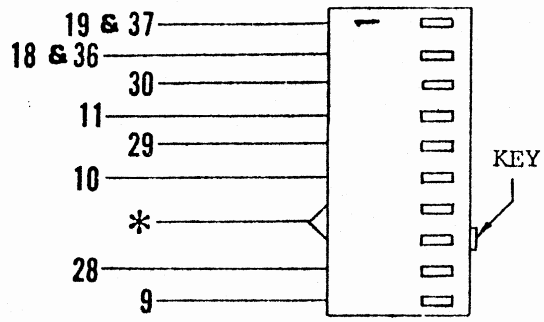


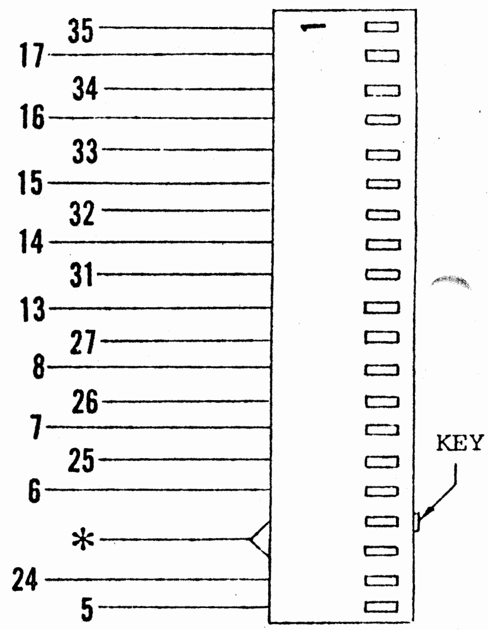
The drawing on the right illustrates the same three female connectors as shown on the bottom of the drawing on page 113. The orientation in the drawing on the right is the same as that on page 113, only rotated 90° counterclockwise.

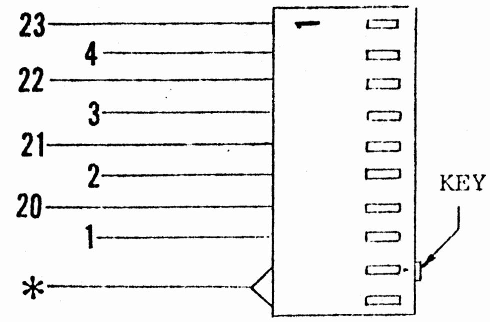
The first step in this assembly process is to attach connector pins to the ends of each of the wires. Do this in the same manner as described on page 97. Note that two of the twisted-pairs have both of their wires attached to a single connector pin.

Once this is completed, the pins can be inserted into the female connectors. The numbers in the drawing on the right refer to the 37-pin connector pin numbers. Use the same procedure as with the previous ribbon cables and insert the pins into the connectors, correlating the 37-pin connector pin numbers on the right with the with the proper wires and positions on the 3 female connectors.

Insert the the plastic keys in the positions shown. Be sure to insert them from the opposite side that the wires are inserted from.







* NO WIRE CONNECTION

CONTROLLER/DRIVE INTERCONNECT CABLE ASSEMBLY

There is one more cable to be assembled for the disk system. This cable will be used to connect the Disk Drive unit with the ALTAIR containing the controller.

1) The first step is to cut a 6 foot length of ribbon cable and remove 2 inches of the cable sheath from each end.

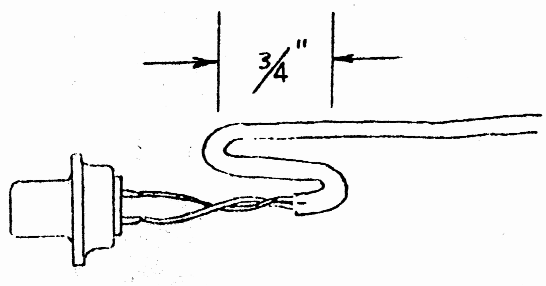
2) There are two grey plastic connector covers included in your kit. Slip one of these over each end of the cable, with the small holes towards the center of the cable and the larger holes towards the free ends. Push the covers down at least a foot so that they will not interfere with the rest of this procedure.

3) Strip 1/8 inch of insulation from both ends of each of the cable wires and tin the exposed portion.

4) Prepare the two remaining 37-pin connectors (one male & one female) in the same manner as the previous 37-pin connectors.

5) For this cable the connections will simply run pin-to-pin. That is, connect pin 1 of the male connector to pin 1 of the female connector. BE SURE NOT TO CONNECT ANY WIRES TO PIN 12 OF EITHER CONNECTOR.

6) Once all 36 wires have been connected on both ends, push the ends of the cable into a fold as shown on the right, and secure it with a double wrap of masking tape. Keep the fold as close as possible to the connector itself.



7) Push the connector covers into place over the two connectors. Do not use any of the hardware supplied with the covers by the factory. Simply mount the 37-pin connectors to the covers using standard 4-40 X 5/16 " screws.

DISK/COMPUTER INTERFACE

Refer to the preliminary documentation release included with this manual for a description of how to hook-up and operate this system.

The above mentioned documentation includes an abreviated version of both the theory and the operation of the ALTAIR FLOPPY DISK SYSTEM.

An updated, complete version of this documention will be sent at a later date, as described in the front of this manual.

DISK CONTROLLER CHECK OUT
WITH DISK DRIVE

A) Preliminary Test

This tests the primary functions of the Disk Drive and Disk Controller.

Enter the following program and then single step through (with Controller and Drive connected).

Address	Instruction		
000,000	076	MVI A	NOTE 1
1	000	Disk Drive Addr (Ø)	
2	323	Output	
3	010	Disk Enable Channel	
4	076	MVI A	NOTE 2
5	004	Head Load (Bit D2=1)	
6	323	Output	
7	011	Disk Control Channel	
10	333	Input	NOTE 3
11	011	Sector Position Channel	
12	333	Input	NOTE 4
13	010	Disk Status Channel	

Note 1

Disk Drive should be enabled at the end of these 4 instructions.

Note 2

Disk Drive Head should be loaded at the end of these 4 instructions.

Note 3

After single stepping these two instructions, the ALTAIR data lights should indicate as follows:

DØ on all the time
D1 on all the time (flashing very fast)
D2 on all the time (flashing very fast)
D3 flashing very fast
D4 flashing slower
D5 flashing slowest
D6 on-not used
D7 on-not used

The flashing lights indicate the index/sector circuits are functioning properly.

Note 4

The last two instructions, when single stepped through, indicate the status or the disk on the data lights as follows:

D∅ - (ENWD) - On
D1 - (MH) - Off
D2 - (HS) - Off
D3 - Not used - Off
D4 - Not used - off
D5 - (INTE) - Off if "INTE" on front panel off
D6 - (TRACK ∅) - Off if disk head on track ∅
D7 - (NRDA) - Flickering, half on - indicates that read circuit is OK.

B) Testing Individual Functions

To test individual disk functions, an output of the correct data pattern must be done on Channel 011.

For example, to step the head in, use this program. Note--The disk must be enabled before doing any disk functions.

Address	Instruction	
000,000	076	MVI A
1	000	Disk Drive Addr.
2	323	Output
3	010	Disk Enable Chan.
4	333	Input
5	377	From Sense SW
6	323	Output
7	011	Disk Control Channel

Set Sense Switch 8 up, others down when single stepping this program. Change switch pattern to control other functions.

SERVICE

Should you have a problem with your unit, it can be returned to MITS for repair. If it is still under warranty any defective part will be replaced free of charge. The purchaser is responsible for all postage. In no case should a unit be shipped back without the outer case fully assembled.

If you need to return the unit to us for any reason, remove the top cover of the drive unit and install the wood block over the door mechanism as it was shipped to you. Secure cover and pack the unit in a sturdy cardboard container and surround it on all sides with a thick layer of packing material. You can use shredded newspaper, foamed plastic or excelsior. The packed carton should be neatly sealed with gummed tape and tied with a stout cord. Be sure to tape a letter containing your name and address, a description of the malfunction, and the original invoice (if the unit is still under warranty) to the outside of the box.

Mail the carton by parcel post or UPS--for extra fast service, ship by air parcel post. Be sure to insure the package.

SHIP TO: MITS, Inc.
2450 Alamo SE
Albuquerque, NM 87106

All warranties are void if any changes have been made to the basic design of the machine or if the internal workings have been tampered with in any way.

mits

2450 Alamo SE
Albuquerque, NM 87106

DISK OPERATORS MANUAL

I. DESCRIPTION OF SYSTEM

A) DISK SPEC SHEET

B) DISK SYSTEM BLOCK DIAGRAM DESCRIPTION:

1. CONTROLLER BOARD 1:

   Controller Board 1 does all input functions to the ALTAIR bus (Read Data, Sector Data, Status Information), as well as Control Addressing of all Disk to ALTAIR I/O.

2. CONTROLLER BOARD 2:

   Controller Board 2 performs all output functions from the ALTAIR bus (Write Data, Disk Control, Disk Enable and Drive Selection).

3. INTERCONNECT CABLE:

   An 18 pair flat cable with two 37 pin connectors, a male on one end, a female on the other. This cable connects the Disk Drive to the ALTAIR Disk Controller and "Daisy Chains" one Disk Drive to another in multiple Disk systems.

4. DISK DRIVE CABINET:

   a) POWER SUPPLY:

      The Disk Drive Cabinet contains a power supply for powering the Disk Buffer and Disk Drive.

   b) THE DISK BUFFER:

      The Disk Buffer board contains the necessary line drivers and receivers for interconnection with long cables to the Disk Drive. In addition, it contains the Disk Drive Address circuitry that allows the Controller to select one of 16 Disk Drives.

      The Disk Buffer board also contains the line drivers for connection of multiple Disk Systems.

c) THE DISK DRIVE:

The Disk Drive, a Pertec FD-400, contains the mechanism and electronics that actually reads and writes data on the Diskette.

II. CONNECTION OF DISK SYSTEM:

A) CONTROLLER BOARDS:

1. Items Supplied:

a) CONTROLLER BOARD 1 (white vert strips)
b) CONTROLLER BOARD 2 (with short cable wired to it)
c) CONTROLLER CABLE (with 37 pin on one end, 3 Molex connectors on the other end)
d) Connector Mounting Bracket and Hardware.

2. Connection of Controller Boards

a) Take cover off ALTAIR (power off!)
b) Feed Molex (flat) connector ends of Controller cable through hole in back of ALTAIR on connector panel: (37 pin connector outside chassis, molex connectors inside chassis).
c) Lay board 1 flat in front of you on the ALTAIR chassis with components up and stab connector to your right (as facing the front of the ALTAIR).
d) Take the short wired cable of board 2 and connect it to the 20 pin connector on board 1 (note polarization key of connector and missing pin on the PC board).
e) Place board 2 flat, to the left of board 1.
f) Connect 20 pin Molex connector on the Controller cable to the 20 pin connector on board 2. Note Keying.
g) Take the 10 pin connector on the Controller cable with the orange and yellow wires connected to it and connect it to the 10 pin connector on board 2. Note Keying.
h) Take the remaining 10 pin connector on the Controller cable with white and gray wires on it and connect it to the 10 pin connector on board 1. Note Keying.
i) Take both boards, hold together and slide into slots, with board 1 on right, board 2 on the left. Be sure wires from connector go out between card guides, and do not catch on card guides.
j) Push cards firmly into connector in ALTAIR mother board.
k) Install 37 pin connector in bracket and on back of ALTAIR, straddling 2 connector holes. Use #4-40 x 5/16 screws, lockwashers and #4-40 nuts.

B) Disk Drive connection to ALTAIR: take the 6 ft flat cable with 1 male and 1 female connector, connect male end to Disk Controller connector on ALTAIR, female end to connector on the Disk Drive marked "To Controller".

C) MULTIPLE DISK DRIVE CONNECTION:

1. With multiple Disk Drives, the Disks should have sequential addresses (ie, for a 3 drive system you should have Disks with addresses Ø, 1, and 2). They may be connected in any order. There serial # sticker has the Disk Address written on it. The Disk Address is determined by four jumper wires in the Disk Buffer P.C. card inside the Drive, and may be changed.

2. Connect the Disks by using the 6 ft. flat cable. Connect the male connector to the connector marked "From Next Disk" on the Disk Drive connected to the Controller. The other end of the cable connects to the next Disk Drive connector marked "To Controller". This procedure is repeated for added Disk Drive.

III. USING THE DISK DRIVE:

A) DISKETTE INFORMATION:

1. Always keep Diskette in envelope when not in use.
2. Keep Diskette away from heat, magnetic fields (flourescent lights, power transformers, etc.) and dust and dirt.
3. Never touch recording surface of Diskette (opposite label side).
4. Always mark your Diskette with what is on them. Use adhesive labels, but don't write on them after they are attached to the Diskette.
5. The Diskette used is hard Sectored (32 Sector holes, 1 index hole). Blank Diskettes are available from MITS for $15.00 each. The Diskettes are <u>not</u> IBM compatible.

B) OPERATING THE DISK DRIVE:

1. Open door to Disk Drive by pulling out and down.
2. Insert Diskette into Drive with label side up, making sure it catches on retaining tab.
3. Close door to Disk Drive.
4. If Disk power is on, wait 10 seconds, after closing door before activating any programs to access the Disk. Wait 10 seconds after turning power on with Diskette in Drive before activating any programs to access the Disk. This is to allow motor speed to stabilize.
5. <u>NEVER</u>: open Disk Drive door or turn power off when Disk Enable and Head Load lights are on. There would be a good possibility that you would interrupt the software during a write function, and destroy data on the Diskette.
6. Consult software documentation on methods used to load basic or use software. For applications where the user wishes to write his own software. See last section, "Controller I/O Information".

ALTAIR DISK CONTROLLER - 15 March 1975

I/O INFORMATION   Revised 4 Sept 1975

A) ADDRESS CODES FOR I/O

	ADDRESS	MODE	FUNCTION
1.	Ø1Ø	Out	Select, Latches and enables controller and Disk Drive
2.	Ø1Ø	In	Indicates Status of Disk Drive and Controller
3.	Ø11	Out	Controls Disk Function
4.	Ø11	In	Indicates sector position of Disk
5.	Ø12	Out	Write data
6.	Ø12	In	Read Data

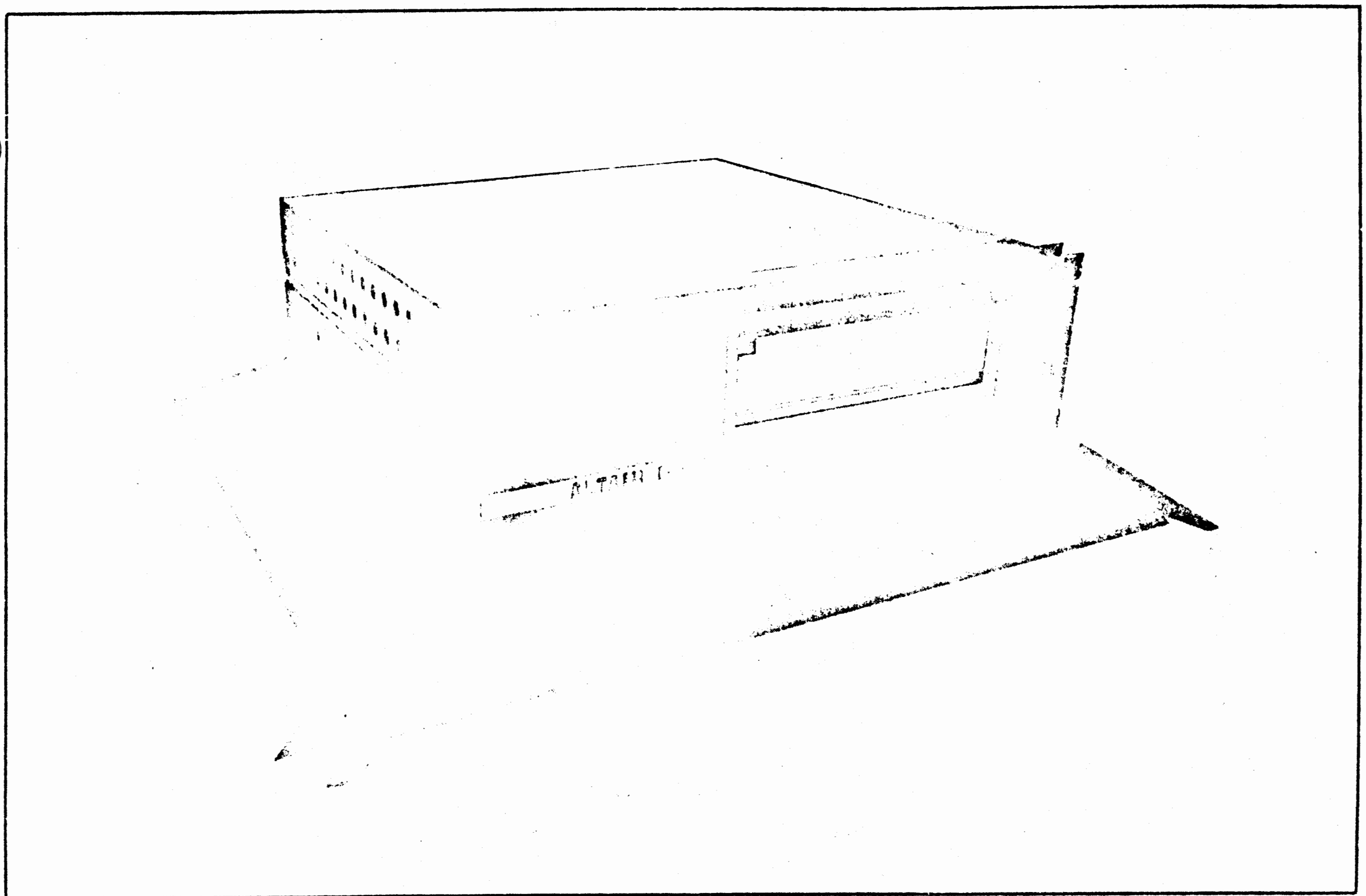
B) DEFINITIONS: In order as listed on Front Page

1. Selection of Disk Drive "OUT" on CH # Ø1Ø

DØ LSB, D1, D2, D3 MSB	Enables 1 of 16 drives (each drive has a unique address, selected by 4 jumper wires) and enables controller
D4, D5, D6	Not used, Don't care
D7	Clears Disk control if set to 1 (DØ-D6 don't care). Disables Disk control

NOTE: a) If Disk Drive door is open, drive and controller cannot be enabled.

b) If Disk power is off, Drive and Controller cannot be enabled.



THE ALTAIR FLOPPY DISK SYSTEM

The *ALTAIR Disk* offers the advantage of nonvolatile memory, plus relatively fast access to data. The *ALTAIR Disk Controller* consists of two PC boards (over 60 I.C.s) that fit in the *ALTAIR* chassis. They inter-connect to each other with 20 wires and connect to the disk through a 37-pin connector mounted on the back of the *ALTAIR*. Data is transferred to and from the disk serially at 250K bits/sec. The disk controller converts the serial data to and from 8-bit parallel words (one word every 32 μ sec). The *ALTAIR CPU* transfers the data, word by word to and from memory, depending on whether the disk is reading or writing. The disk controller also controls all mechanical functions of the disk as well as presenting disk status to the computer. All timing functions are done by hardware to free the computer for other tasks. Since the *floppy diskette* is divided into 32 sectors, a hardware interrupt system can be enabled to notify the CPU at the beginning of each sector. Power consumption is approximately 1.1 amperes from the +8v (VCC) line for the two boards.

The Disk Drive unit, using a PERTEC FD400 mounted in an Optima case (5½" high—same depth and width as computer), includes a *power supply PC board* and a *Buffer/Address/Line Driver P.C. Board*. A cooling fan maintains low ambient temperature for continuous operation. The disk drive cabinet has two 37-pin connectors on the back panel, one is the input from the disk controller, the other is the output to additional disk drives. Up to 16 drives may be attached to one controller.

The 88-DCDD consists of the disk controller and one disk drive with an interconnect cable. The 88-Disk is one disk drive for adding storage capability to the 88-DCDD and includes the interconnect cable.

The *ALTAIR Disk Format* allows storage of over 300,000 bytes. Since the disk is hard sectored (32 sectors for each track), we write 137 bytes on each sector, 9 of which are used internally (track#, checksum) leaving 128 data bytes per sector, 4096 per track. One floppy diskette is supplied with each drive; extra floppies are available for purchase. A *software driver* for the floppy disk is available at no charge and is supplied with the disk as a source listing. The disk operating system—which has a complete file structure and utilities for copying, deleting and sorting files—costs extra. *Extended BASIC*, which uses random and sequential file access for the floppy disk, is also available.

Specifications

Rotational Speed	**360 rpm (166.7 ms/rev)**
Access Times	**Track to track, 10 ms** **Head settle, 20 ms** **Head load, 40 ms** **Average time to read or write, 400 ms** **Worst case, 1 sec**
Head Life	**Over 10,000 hours of head to disk contact**
Disk Life	**Over 1 million passes/track**
Data Transfer Rate	**250K bits/sec**
Power Consumption	**117VAC 110W**
Diskette	**Hard sectored, 32 sectors + index, Dysan 101 floppy disk, 77 tracks**

c) If Disk interconnect cable is not connected between the Controller and the Drive, Drive and Controller cannot be enabled.

2. Status (Ø1Ø - INP) indicates Disk status when Drive and Controller enabled. Also gives valid "INTE" status (D5), with Disk enabled.
True Condition = 0, False = 1
All False if Disk & Controller not enabled, and all false if no Disk in Drive.

D0 - ENWD - Enter new Write data - indicates Write circuit is ready for new data byte to be written. It occurs every 32 μs and starts 28Ø μs after sector true (when Write enabled). It is reset by outputting to the Write data channel (Ø12).

D1 - Move Head - Indicates head movement allowed when true (step IN, step OUT,). Goes False for 1Øms true 1ms, false 2Øms after step command. May step every 1Øms. Goes false for 4Øms after head load. Goes false during Write and 475 μs after Write to allow completion of trim erase.

D2 - HS - Head Status - True 4Øms after head loaded or step command if stepping with head already loaded. Indicates when head is properly loaded for reading and writing. Also enables sector status channel when true.

D3 - Not Used
= Ø
D4 - Not Used

D5 - INTE - Indicates interrupt enabled.

D6 - TRACK Ø - Indicates when head is on outermost track.

D7 - NRDA - New read data available - indicates that the read circuit has 1 byte of data ready to be taken from the read data channel (Ø12). After the SYNC* bit is detected, it occurs every 32 μs and is reset by an input instruction on channel Ø12. The byte containing the SYNC bit is the first byte read from the disk.

* See "WRITE ENABLE".

3. Control (011 - Out) - Controls Disk operations when Disk Drive and Controller enabled. A True signal, logic 1, on a data line will control the Disk as follows:

D∅ - Step IN - Steps Disk head in one position to higher numbered track.

D1 - Step OUT - Steps Disk head out one position to lower numbered track.

D2 - Head Load - Loads Head onto Disk - Enables sector position status.

D3 - Head Unload - Removes Head from Disk surface may be unloaded immediately after "Write Enable" (Write and trim erase circuits hold head on until through).

D4 - IE - Interrupt enable - Enables interrupts to occur when SR∅ True (See Sector Def).

D5 - ID - Interrupt Disable - Disables interrupt circuit. Interrupt circuit also disabled by clearing Disk Control.

D6 - HCS - Head Current Switch - Must be True when outputting a Write instruction with the Head on Tracks 43-76. This reduces head current and optimizes resolution on inner tracks (automatically reset at end of Writing a Sector).

D7 - Write Enable - Initiates Write sequence as follows:

1. Disk selected and enabled, Head loaded, enabling sector status.

2. SR∅ (Sector True) Detected for desired sector, write enabled by software.

3. 2∅∅ µs from Write Enable, trim erase automatically turned on. 280 µs from start of sector, "ENWD" goes True, SYNC byte written (by software).

4. First byte written always has most significant (D7) bit A "1" (SYNC Bit) (most significant bit written first).

5. ENWD goes true every 32 µs. MAX. No. of data bytes per sector 137 (including SYNC).

6. Last or 138th byte written must be A ∅∅∅. This will be written for the remainder of the sector. Ignore "ENWD" from this point on to end of sector.

D7 - Write Enable, Continued.

7. At end of sector, the write circuit automatically disabled, trim erase disabled 475 μs later.

NOTE: a) Write circuit will continue writing last byte outputted on CH # Ø12 to the end of that sector.

b) Head may be unloaded anytime during Write cycle if no read or write function is expected after current write cycle. Once Write is enabled, it holds the head loaded for the required time. (For writing and trim erase.)

4. Sector Position (Ø11-INP) with Disk Drive and Controller enabled, and 4Øms after head is loaded, the sector information is as follows:

DØ - SRØ - Sector True - True when = Ø, and is 30 μs long. The Write mode should begin as close as possible to the time that DØ goes true. Write data will be requested 280 μs after DØ goes true. Read data will be available 140 μs after DØ goes true.

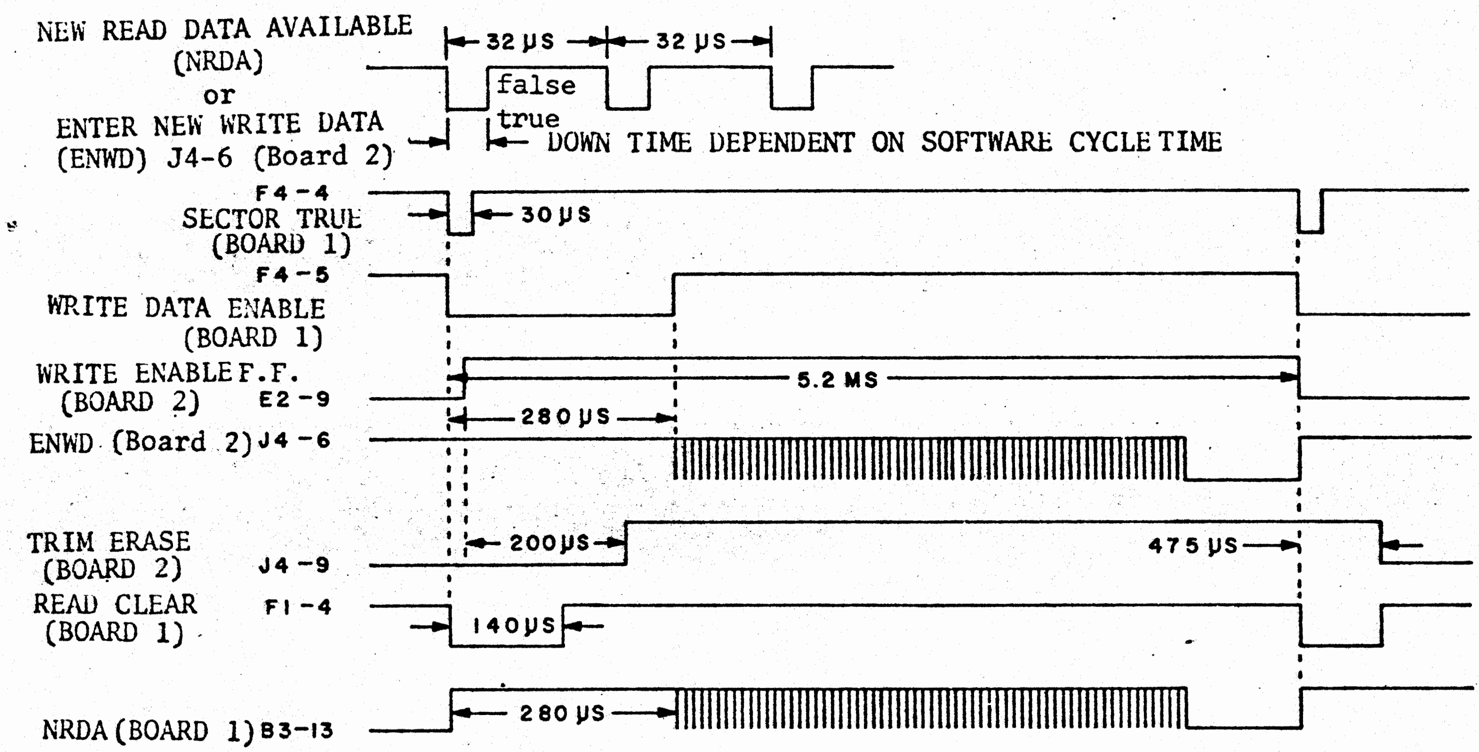
SECTOR #	Ø	1	2	3	31
D1-SR1-	Ø	1	Ø	1	1
D2-SR2-	Ø	Ø	1	1	1
D3-SR3-	Ø	Ø	Ø	Ø	1
D4-SR4-	Ø	Ø	Ø	Ø	1
D5-SR5-	Ø	Ø	Ø	Ø	1

D6 Not Used, = 1
D7 Not Used, = 1

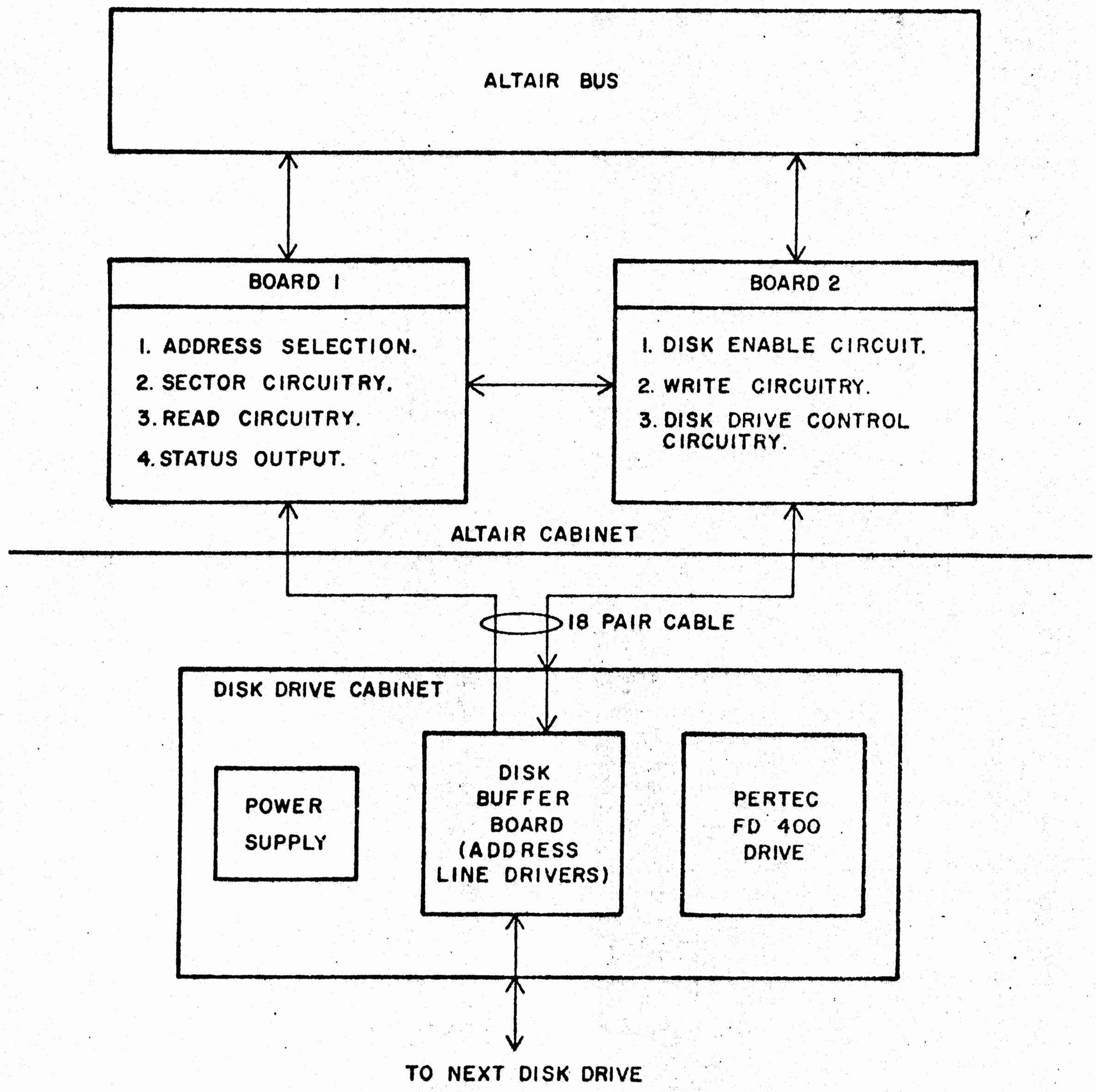
5. Write Data (Ø12-OUT) Outputted on the "ENWD" status request.

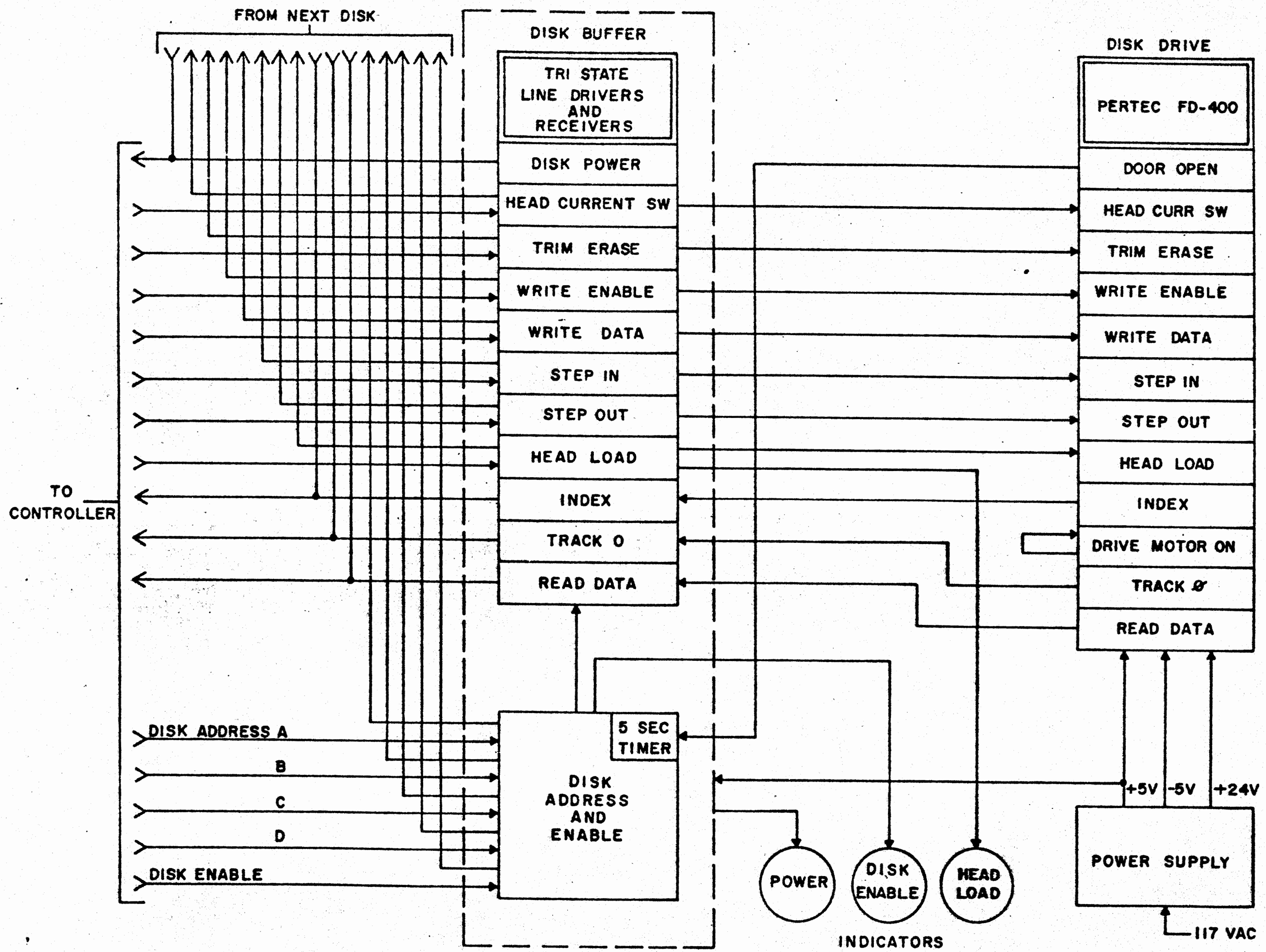
6. Read Data (012-IN) Inputted on the "NRDA" status flag.

READ/WRITE TIMING
DURING READ OR WRITE FUNCTION



DISK SYSTEM BLOCK DIAGRAM

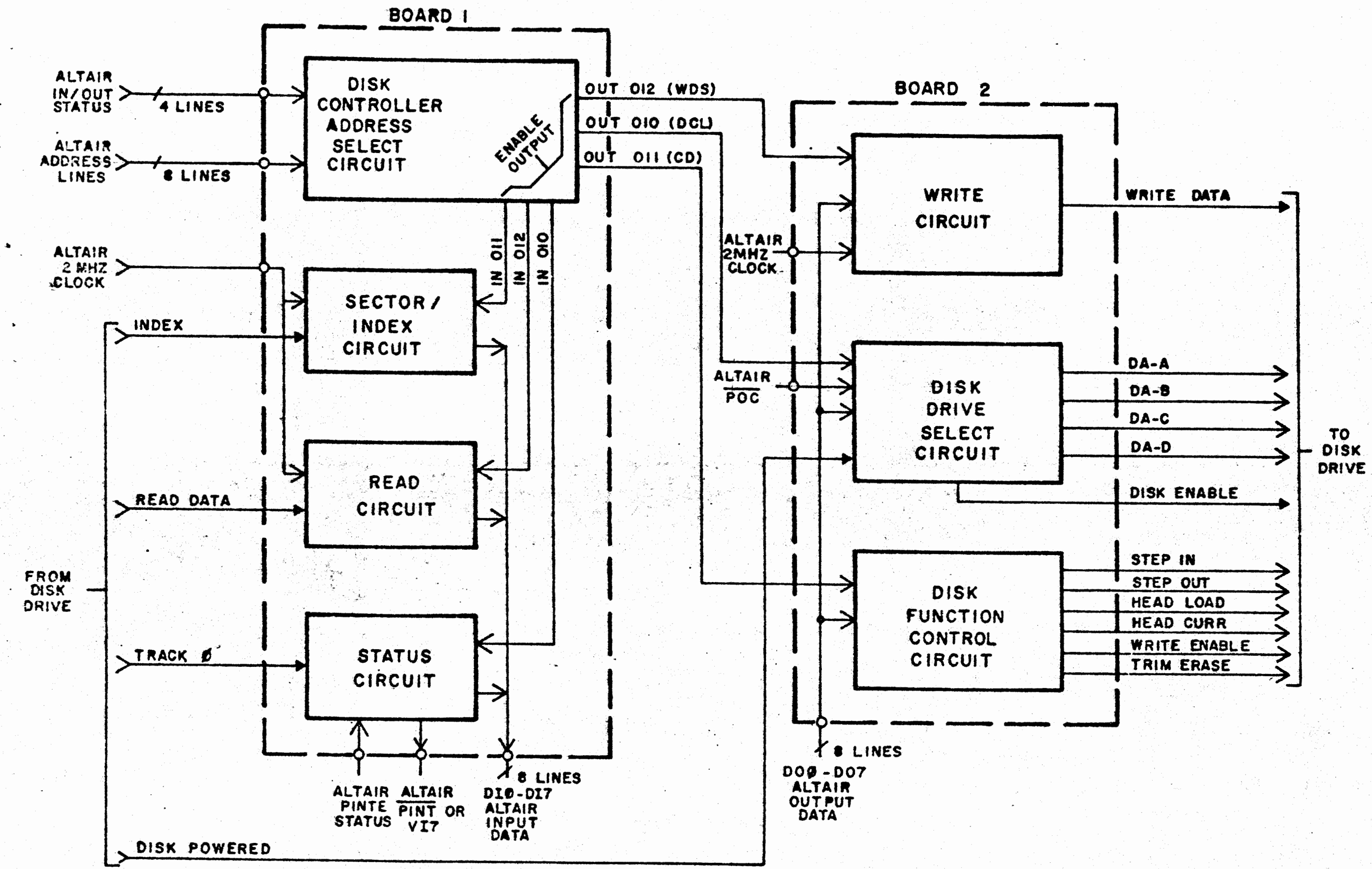




88 DISK BLOCK DIAGRAM

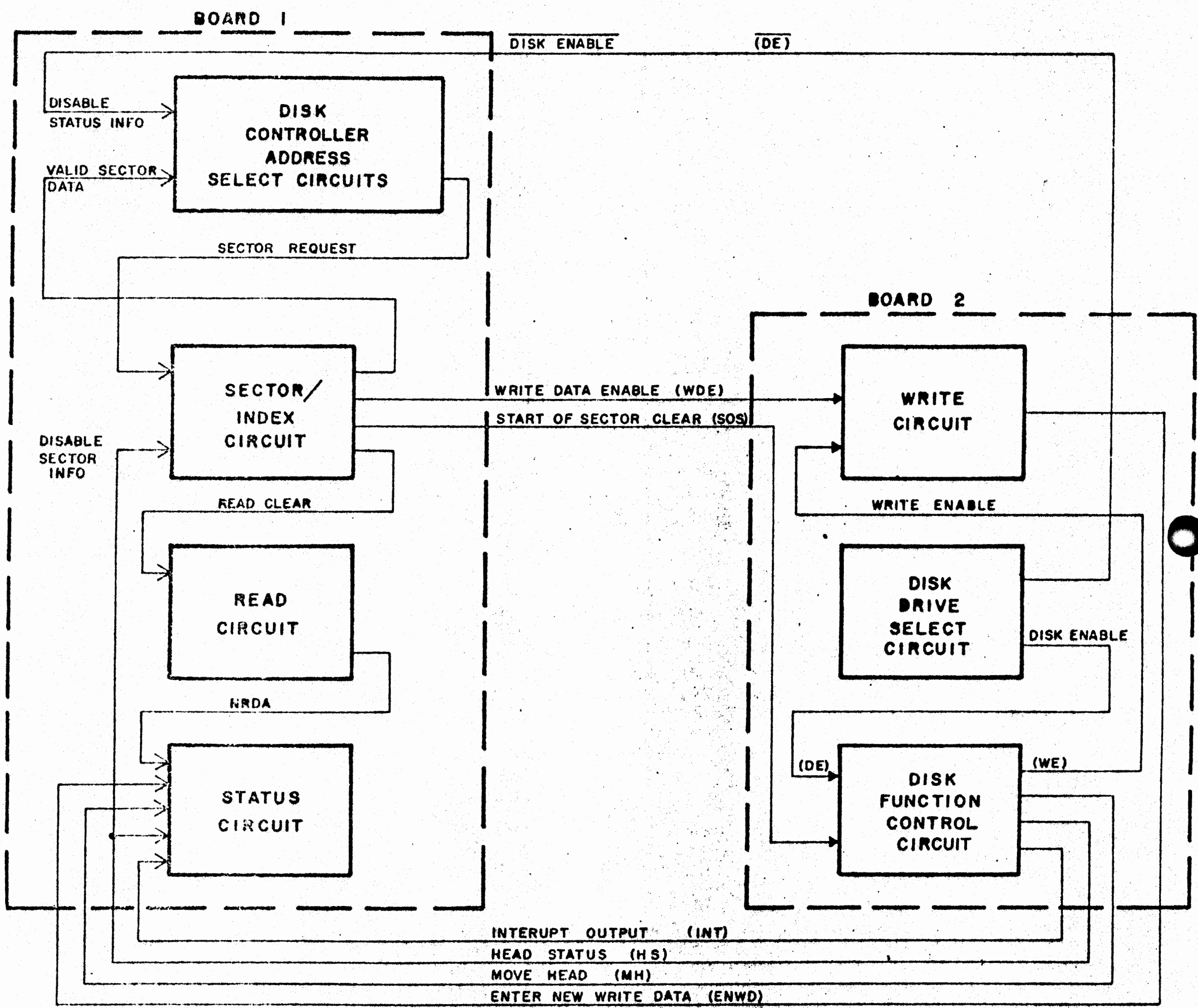
DISK CONTROLLER BLOCK DIAGRAM

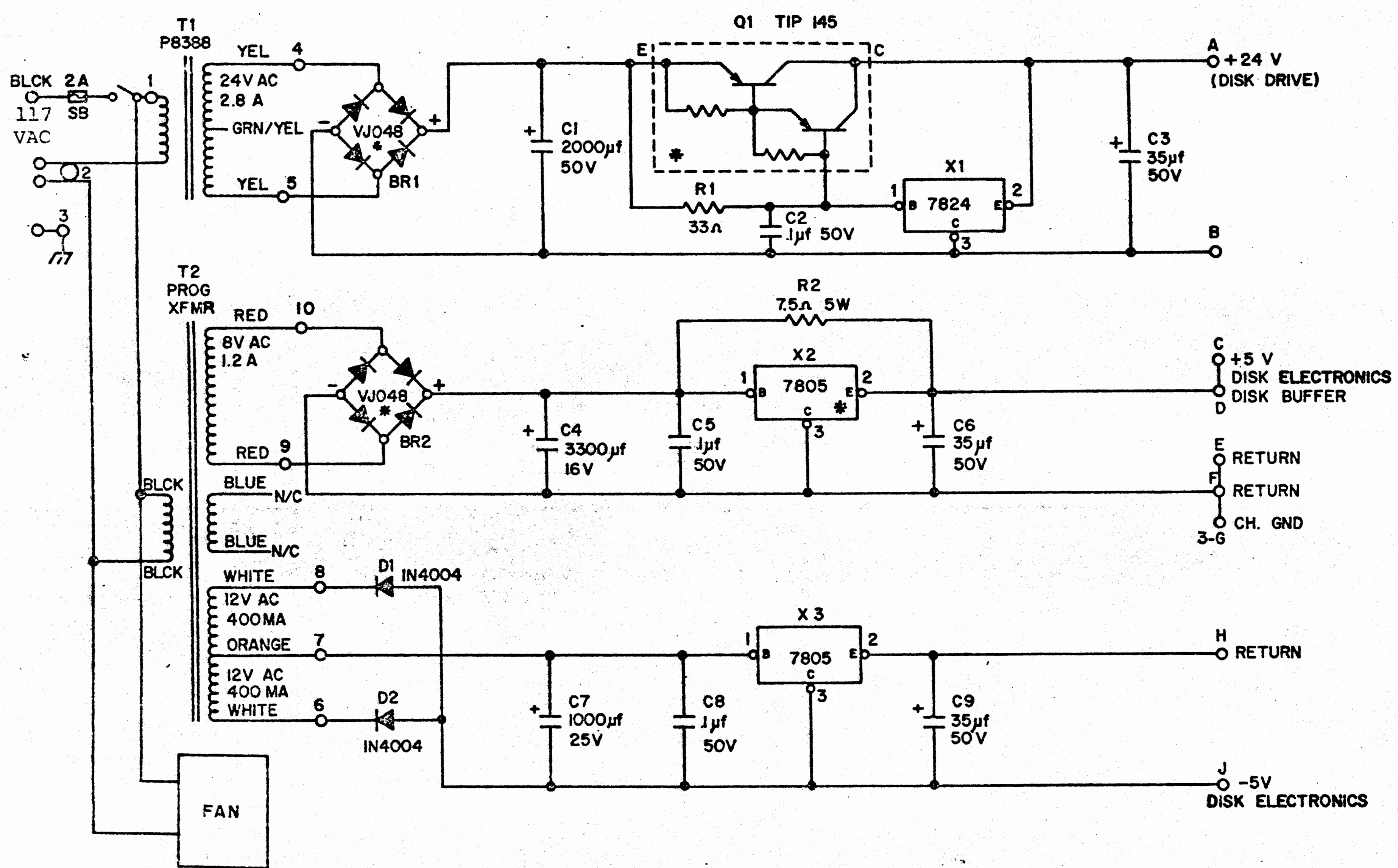
SHEET I EXTERNAL CONNECTIONS AND ADDRESS SELECT



DISK CONTROLLER BLOCK DIAGRAM

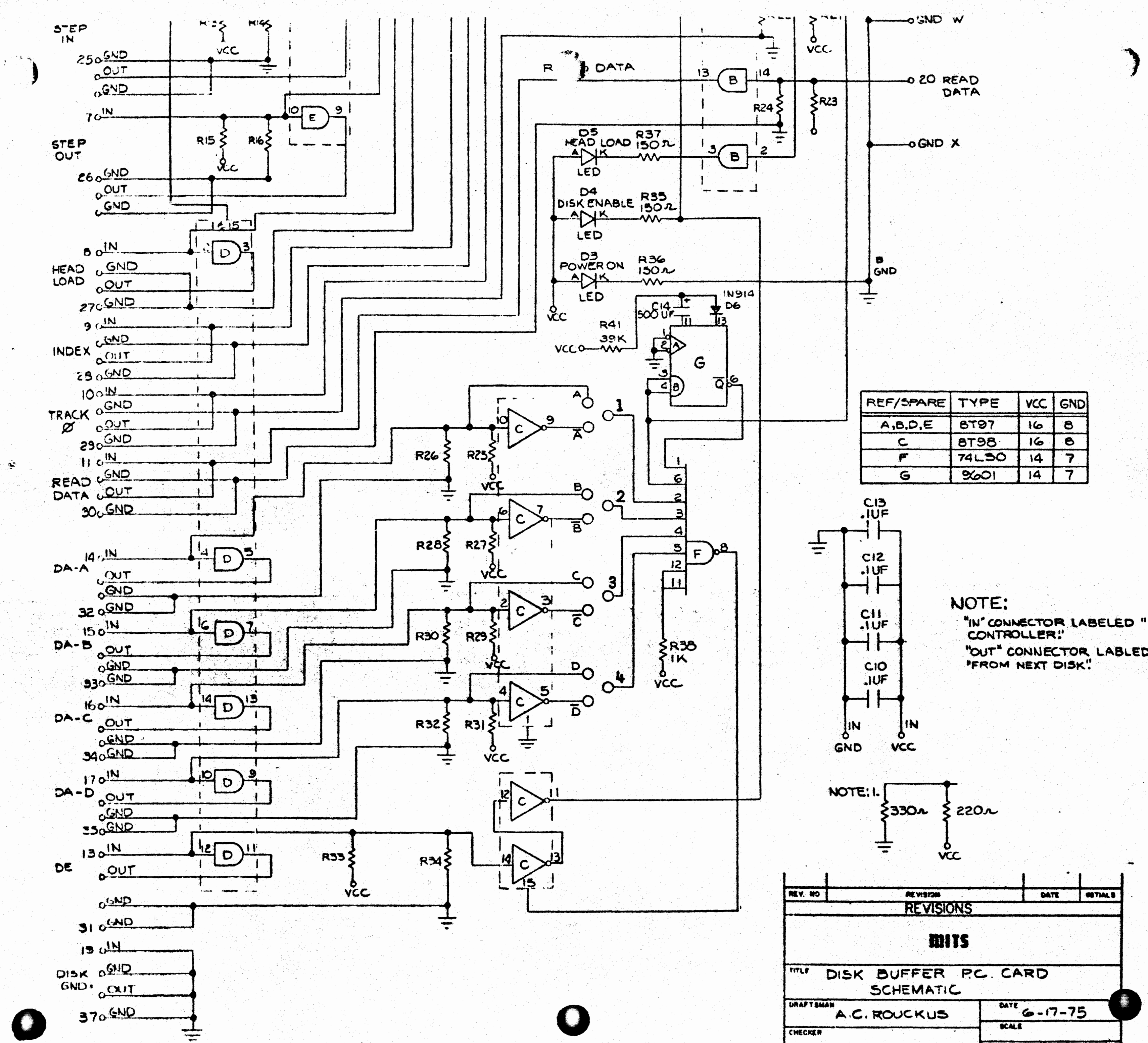
SHEET 2 INTERNAL CONNECTIONS





✱ HEATSINK ON CHASSIS

DISK DRIVE POWER SUPPLY



REF/SPARE	TYPE	VCC	GND
A,B,D,E	8T97	16	8
C	8T98	16	8
F	74LS30	14	7
G	9601	14	7

NOTE:
"IN" CONNECTOR LABELED "TO CONTROLLER."
"OUT" CONNECTOR LABLED "FROM NEXT DISK."

REV. NO	REVISION	DATE	INITIALS

REVISIONS

MITS

TITLE DISK BUFFER P.C. CARD SCHEMATIC

DRAFTSMAN A.C. ROUCKUS

DATE 6-17-75

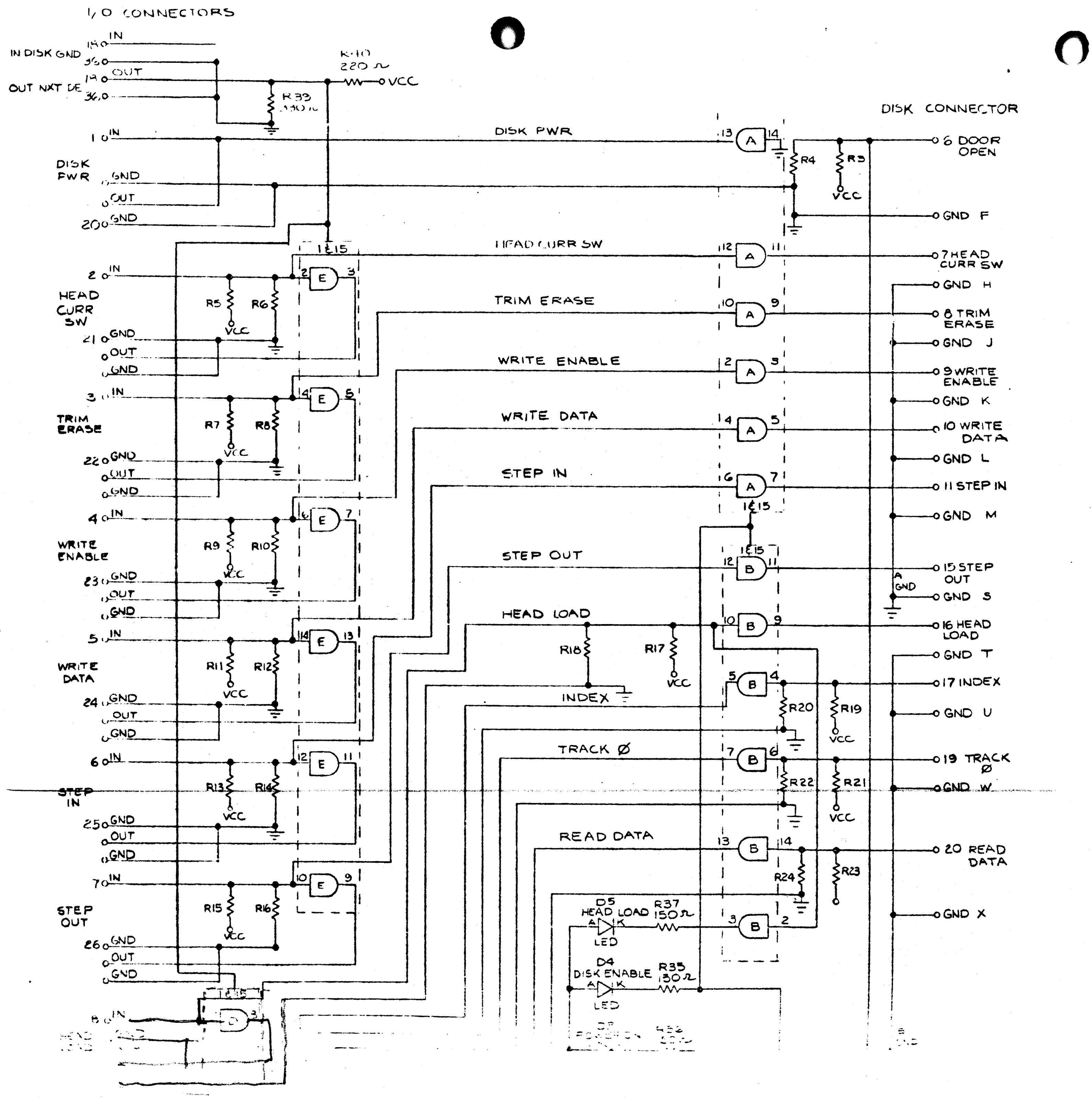
CHECKER

SCALE

DRAWING NO.

ENGINEERING

PROJ MGR


I/O CONNECTORS
IN DISK GND
OUT NXT DE
R40
220 Ω
VCC
R33
DISK CONNECTOR
DISK PWR
6 DOOR OPEN
R4
R5
VCC
GND F
HEAD CURR SW
7 HEAD CURR SW
HEAD CURR SW
R5
R6
VCC
GND H
TRIM ERASE
8 TRIM ERASE
GND J
WRITE ENABLE
9 WRITE ENABLE
TRIM ERASE
R7
R8
GND K
WRITE DATA
10 WRITE DATA
GND L
STEP IN
11 STEP IN
WRITE ENABLE
R9
R10
GND M
STEP OUT
15 STEP OUT
GND S
HEAD LOAD
16 HEAD LOAD
R18
R17
WRITE DATA
R11
R12
GND T
INDEX
17 INDEX
R20
R19
GND U
TRACK Ø
19 TRACK Ø
R22
R21
STEP IN
R13
R14
GND W
READ DATA
20 READ DATA
R24
R23
STEP OUT
R15
R16
GND X
D5
HEAD LOAD
LED
R37
150 Ω
D4
DISK ENABLE
LED
R35
150 Ω

altair™ DISK OPERATING SYSTEM DOCUMENTATION

mits

altair DISK OPERATING SYSTEM DOCUMENTATION

First Printing, June, 1977

mits
2450 Alamo S.E. / Albuquerque, New Mexico 87106

TABLE OF CONTENTS

APPENDICES

ALTAIR DOS DOCUMENTATION

SECTION I

INTRODUCTION

1. INTRODUCTION

1-1. Introduction to This Manual

The Altair Disk Operating System (DOS) is a system for developing and running Assembly Language programs. It consists of a Monitor and several system programs. The parts of this manual describe the various components of the system.

Chapter 2--the Monitor. The Monitor provides control and disk file management for all of DOS. Monitor Input/Output routines are available to any program running under DOS.

Chapter 3--the Text Editor. The Editor (EDIT) creates, modifies and saves ASCII coded files. Typical Editor files include Assembly Language programs and data.

Chapter 4--the Assembler. The Assembler (ASM) converts symbolic Assembly Language programs into relocatable machine code modules.

Chapter 5--the Linking Loader. The Linking Loader (LINK) loads the relocatable object code modules into memory, assigns addresses to symbols and resolves external references.

Chapter 6--Debug. Debug is a versatile symbolic debugging program. With Debug, the programmer can interrupt execution of a program, examine and modify the contents of register and memory locations.

Chapter 7--Miscellaneous System Programs.

Console (CNS) transfers command of the Monitor from one terminal device to another.

Initialize (INIT) allows the system parameters (amount of memory, number of disks, etc.) to be changed without reloading the system.

1-2. Loading and Initializing DOS

When the computer is first turned on, there is nothing of value in the semiconductor read/write memory. Therefore, before DOS can be used, the Monitor must be loaded from disk. This requires another program, the loader. The loader may reside in read-only memory or may be loaded from paper tape or cassette.

A. Systems with a Disk Boot Loader PROM mounted in the proper slot of a PROM Memory Card have the loader program readily available in non-volatile memory. Use the following procedure to load DOS with the DBL PROM:

1. Turn on the power to the computer, disk drives and peripherals.
2. Raise STOP and RESET simultaneously and then release them.
3. Raise switches A15-A8 and lower switches A7-AØ.
4. Actuate EXAMINE.
5. Make sure the DOS diskette is mounted in disk drive 0, that the door is closed and the disk has come up to speed (approximately 5 seconds).
6. Enter sense switch settings for the terminal I/O board from Table 1-A.
7. Press RUN.

DOS should start up and print MEMORY SIZE? For the remainder of the initialization procedure, see Section C below.

B. For systems without the DBL PROM, the loading procedure involves entering a bootstrap loader from the computer front panel, running it to load a disk loader program from paper tape or cassette and then running that loader to load the Monitor from disk. The procedure for doing this is as follows:

1. Turn on the power to the computer and peripheral devices.
2. Raise the STOP and RESET switches simultaneously and then release them.
3. Make sure the terminal is on-line (on a Teletype™, this means the mode switch is set to LINE).

Now enter the proper loader program for the device through which the loader tape is to be entered. The bootstrap loaders are in Appendix F.

The bootstrap loaders are entered on the front panel switches A7 - A0. Each switch has two positions, up and down. By convention, up is designated as 1 and down as 0. Therefore, the eight switches represent one byte of data. Each group of three switches, starting from the right, can represent the digits 0 through 7. The leftmost two switches represent the digits 0 through 3. For example, to enter the octal number 315, the switches A0 through A7 are set to correspond to the following table:

Switch	A7	A6	A5	A4	A3	A2	A1	A0
Position	up	up	down	down	up	up	down	up
Octal Digit	3			1			5	

The data bytes of the loader programs are shown in octal and are to be entered on A0 - A7 in this manner. To enter the programs:

4. Put switches A0 - A15 in the down position.
5. Raise EXAMINE.
6. Put the first loader program data byte in switches A0 - A7.
7. Raise DEPOSIT.
8. Put the next data byte in A0 - A7.
9. Depress DEPOSIT NEXT
10. Repeat steps 8 and 9 for each successive data byte until the loader is completely entered.

Now check the loader to make sure it has been entered correctly:

11. Put switches A0 - A15 in the down position.
12. Raise EXAMINE.
13. Check to see that the lights D0 - D7 correspond to the correct data byte for the first location. A light on indicates 1; off means 0. The rightmost three lights correspond to the rightmost octal digit. The next three lights represent the middle digit and the leftmost two lights represent the left digit.

    If the data byte is correct, go to step 16.

    If the data byte is not correct, go to step 14.
14. Put the correct value in switches A0 - A7.
15. Depress DEPOSIT.
16. Depress EXAMINE NEXT.
17. Check each successive byte by repeating steps 13 - 16 until the whole loader is checked.
18. If there were any incorrect bytes, check the whole loader again to see that they were corrected.

Now the paper tape or cassette labelled DISK LOADER can be read. For the paper tape version, put the tape in the reader and make sure it is positioned on the leader. The leader is the section of tape at the beginning with a series of 302_8 characters (3 of

8 holes punched). For the cassette version, put the cassette in the reader and make sure it is completely rewound.

19. Put switches A0 - A15 in the down position.

20. Raise EXAMINE.

21. Enter the proper sense switch settings for the load and terminal devices in switches A8 - A15. The rightmost four switches contain the load device setting, and the leftmost switches contain the setting for the terminal devices. Table 1-A shows both the octal sense switch setting and the load and terminal switches to be raised for each standard Altair system peripheral. If a device is used for interface to the terminal, the switches in the "Terminal Switches" column must be raised. If the device interfaces the peripheral through which DOS is being loaded, the "Load Switches" are raised.

	Sense Switch Setting	Terminal Switches	Load Switches	Channels
2SIO (2 stop bits)	0	None	None	20,21
2SIO (1 stop bit)	1	A12	A8	20,21
SIO	2	A13	A9	0,1
ACR	3	A13,A12	A9,A8	6,7
4PIO	4	A14	A10	40,41, 42,43
PIO	5	A14,A12	A10,A8	4,5
Non-Standard terminal	14			
No terminal	15			

22. Start the loading process. If the load device is connected to the computer through an 88-SIO A, B or C or an 88-PIO board, start the tape reader and then press the RUN switch on the computer front panel. For the 2SIO or 4PIO boards, press RUN and then start the reader. For the ACR, rewind and start the cassette. Listen to the signal from the tape (through an auxiliary earphone). When the steady tone changes to a warble, press RUN on the computer.

If the checksum loader detects a loading error, it turns on the Interrupt Enable light and stores the ASCII code of an error letter in memory location 0. The error letter is also transmitted over all terminal data channels. If a terminal is connected to one of these ports, it prints the error letter. The error letters are as follows:

C Checksum error. If the checksum on the DOS disk file does not equal the checksum generated by the loader, C error results. The error may not occur if the diskette is loaded again. If it does occur three times consecutively, the loader tape or diskette is at fault and must be replaced.

M Memory error. Data from the disk does not store properly. The location at which the error occurred is stored at locations 1 and 2 absolute.

O Overlay error. An attempt was made to load data over the loader.

I Invalid Load Device. The setting of the sense switches is incorrect.

C. When the Monitor has been loaded correctly, it responds with the first initialization question.

MEMORY SIZE?

Here the programmer may specify the amount of memory, in bytes, to be used by DOS. Typing a carriage return or zero causes DOS to use all of the read/write memory in the system. The next question is

INTERRUPTS?

Typing Y enables input interrupts and Typing N or carriage return disables them. If interrupts are enabled, special characters may be used to control program execution.

> NOTE
>
> Input interrupt features may be used only if the input interface board is strapped to accept interrupts. See Section 2-2 for information on I/O interrupts. If interrupts are not strapped, the answer to the INTERRUPTS? question must be N.

The next question is

HIGHEST DISK NUMBER?

to which the programmer responds with zero if there is one disk in the system, 1 if there are two disks and so on. The next question is

HOW MANY DISK FILES?

to which the programmer responds with the number of disk files (both sequential and random) to be open simultaneously. Responding with a carriage return sets the number of files at zero. Finally, DOS asks

HOW MANY RANDOM FILES?

Again, the programmer responds with a number or with a carriage return, which specifies zero random files.

To save time, especially when a slow terminal is in use, all of the initialization answers can be entered at once with the parameters separated by spaces. For example:

MEMORY SIZE? 0 Y 1 2 0

tells DOS that

1. it is to use all available memory,
2. input interrupts are enabled,
3. there are two disk drives in the system,
4. two sequential and
5. no random disk files are to be open at any given time.

When DOS has been properly initialized, it prints the following prompt message

DOS MONITOR VER x.x

.

The Monitor prints a period to indicate that it is now ready to receive commands.

1-3. Program Development Procedure

DOS is designed to allow the translation of an Assembly language program on paper to an operating Machine Language program with a minimum of time and effort. The process involves entering the Assembly language program into a disk file with the Text Editor, translating the file to Machine language with the Assembler and loading the program into memory with the Linking Loader.

Before the process can proceed, the disks in use must be mounted with the MNT command. To mount disk 0, the following command is used:

. MNT 0 <cr>

where <cr> means carriage return. Other disks may be mounted in the same command by typing their numbers after the zero, separated by spaces.

Mounting the disk(s) tells DOS the location of all the files and free space on each disk. If an attempt is made to run a program before the disk on which it is stored is mounted, a PROGRAM NOT FOUND error will result.

1. The first step in program development is to enter the program into a disk file with the Text Editor. The Editor is loaded from disk and run by the following command:

   .EDIT<cr>

   When it is loaded, it prints

   DOS EDITOR VER x.x
   ENTER FILE NAME

   to which the user replies with the name of the file to be entered or edited. The editor then prints

   ENTER DEVICE NUMBER

   which is answered with the number of the disk drive where the file is stored.

   Assume that an Assembly language program called SAMP is entered into a file on disk drive 0. The Editor is run with the following command:

   .EDIT SAMP 0 <cr>

   The file name (SAMP) and device number (disk 0) can be entered in the EDIT command to avoid the necessity of asking the file name and device number. The Editor searches disk drive 0 for a file name SAMP to edit. If it finds no such file, it prints the following messages:

CREATING FILE

00100

00100 is the number of the first line of the file. Now, all that is necessary is to enter the lines of the program.

```
00100    LDA            IER         LOAD MULTIPLIER<cr>
00110    LHLD           CAND        LOAD MULTIPLICAND<cr>
  .
  .
  .
```

After each carriage return, the next line number is generated automatically so that the next line can be entered. This process continues until all the lines of the program have been entered.

```
00340    PROD DB  0,0 <cr>
00350    END <cr>
00360    <cr>
```

To stop the generation of line numbers, type a null line (just a <cr>). The Editor prints an asterisk (*) to indicate it is ready to accept new commands. To check the file in order to make sure it has been entered without error, type

*P

This prints all of the lines on the current page with their line numbers. In this example, there is only one page (see paging commands, p. 40 , for an explanation of program pages), so the P command prints the whole file. The output appears as follows:

```
*P
00100              LDA     IER
00110              LHLD    CAND
00120    SHFTR     RAR
00130    SHFTR     RAR
             .
             .
             .
00240    CAND      DB      64
00250    PROD      DB      0,0
```

Suppose the line at 120 was inadvertantly entered again at line 130. To eliminate one of them, use the D (for Delete) command.

```
*D 130 <cr>
*
```

It is not necessary to type the leading zeros in the line number. To add another line between number 100 and 110, use the I (for Insert) command.

```
*I 100
00105 ;          A COMMENT LINE <cr>
00107 <cr>
```

The line number specified is that of the existing line immediately before the desired position of the new line. The Editor generates a line number halfway between the two existing lines. After typing the new line, a <cr> causes another number to be generated halfway between the inserted line and the next existing line. New lines can be inserted in this manner until there is no more room. Insertion of new lines is stopped by typing a null line.

When the file is in satisfactory form, the Editor is exited by typing the following command:

```
*E
```

This makes all of the changes, closes all of the files properly and provides a backup file. The backup file is the edited file as it appeared before the latest series of changes were made. If the edited file is unusable for some reason, the backup may be used to replace it.

2. When the program has been entered into a disk file with the Editor, it may be submitted to the Assembler for translation into machine language.

   The Assembler is loaded and run with the following command:

```
.ASM <cr>
```

   The Assembler prints

```
DOS ASM VER x.x
ENTER FILE NAME
```

The user enters the name of the Assembly language program file and a <cr>. The Assembler then prints

ENTER DEVICE NUMBER

to which the user replies with the number of the disk drive on which the file resides and a <cr>.

At this point, the Assembler proceeds immediately to assemble the program in the specified file. In our example, we can type

.ASM SAMP 0 <cr>

to avoid having the computer ask for the file name and drive number.

The Assembler produces a file with the machine language program and a listing. The listing is that of the source code (the input to the Assembler) along with other pertinent information.

The Assembler listing of our sample program appears as follows:

SAMP LISTING

```
000000  072  000033'  000100          LDA    IER     LOAD MULTIPLIER
000003  052  000034'  000110          LHLD   CAND    LOAD MULTIPLICAND
000006  037           000120  SHFTR   RAR            SHIFT 'ER RIGHT
000007  322  000024'  000130          JNC    SCAN    JUMP IF NO CARRY
000012  077           000135          CMC            TURN OFF CARRY
000013  353           000140          XCHG           SAVE 'CAND IN C,D
000014  052  000036'  000150          LHLD   PROD    LOAD PROD IN H,L
000017  031           000160          DAD    D       ADD 'CAND TO PROD
000020  042  000036'  000170          SHLD   PROD    STORE PROD
000023  353           000180          XCHG           RESTORE 'CAND
000024  051           000190  SCAN    DAD    H       SHIFT LEFT
000025  322  000006'  000200          JNC    SHFTR   REPEAT IF NOT FINISHED
000030  303  000000   000225          JMP    000     JUMP TO MONITOR WHEN
000033                000228  ;                      FINISHED
000033  040           000230  IER     DB     32
000034  200  000      000240  CAND    DB     128,0
000036  000  000      000250  PROD    DB     0,0
000040                000260          END
```

The rightmost four columns are the source listing. Note that there is not much room for comments at the end of the line. If the comments are too long for the allotted space, the excess is printed on the next line and operation is not affected.

The next column to the left is the Text Editor's line number. The next two columns are the octal representation of the object code (the output of the Assembler). If the source instruction does not produce a machine instruction (END, for example), this column is left blank. If the source instruction defines the contents of memory (DB or DW, for example), those contents appear in the object code column. Source instructions that produce object code instructions (LDA, for example) are represented by the octal instruction code and the address of the operand. Addresses followed by an apostrophe are to be relocated. Their actual addresses are not determined until the program is loaded into memory.

Finally, the leftmost column is a list of the relative addresses of the object code instructions and memory areas. If a letter precedes the address, it indicates an error. The letter designates the nature of the error and the position indicates the address where the error occurred. A list of error letters and their meanings is in section 4-4, p. 71.

If an error is detected by the Assembler, it can be corrected by reentering the Text Editor and making the necessary changes. The ability to pass programs rapidly from the Text Editor to the Assembler and back makes DOS an extremely effective tool for writing and debugging Assembly language programs.

3. Finally, the Linking Loader is used to load the program into memory and execute the program. The Linking Loader is loaded typing the following command:

    . LINK <cr>

   When the Linking Loader starts, it prints

    DOS LINK VER 1.0
    *

   To load the sample program, type

    *L SAMP 0 <cr>

   If the file name and drive number had been omitted, LINK would have asked for them. This command causes LINK to load our file into memory beginning at location 24000_8. Other starting addresses can be specified (see Linking Loader, L command, p.

76), but the default value is adequate for our purposes. The following command causes the program to be executed:

*X <cr>

This command causes control to be passed to whatever program begins at location 24000_8. Again, other starting addresses = 2800 H can be specified (see Linking Loader, X command, p. 51). If the program does not run as expected (and that is not improbable), the program bugs can be tracked down by Debug. For a description of the use of Debug, see Section 6, p. 83.

1-4. Notation and Definitions

In the specification of command formats and examples, the following notation conventions are used:

< >	Angle brackets enclose information that must be supplied by the user
[ ]	Square brackets enclose information that is optional and may be specified by the user.
<cr>	Carriage return (ASCII 013) on most terminals, <cr> is typed with the Return key.
<space>	a space (ASCII code 032)
Control/x	where x is a character, is typed by holding down the Control key while typing the character.

In examples, characters output by the computer are underlined. Information typed by the user is presented exactly as it is to be typed. All punctuation and spacing must be observed.

The following definitions are used throughout this manual:

byte	eight bits of binary information. Memory locations each contain 1 byte of information and the ASCII code uses 1 byte to represent 1 character.
file	set of information accessible to a program by name or number. Program modules, data blocks and information transferred to or from I/O devices may all be considered to be files. In this manual, files are divided into two broad classes: Sequential and Random.

A Sequential file is organized as a string of bytes of information. From any point in a sequential file, only the next byte may be accessed directly. Data bytes are written after the last existing byte of the file. Sequential files can be divided into two types, depending upon how the data bytes are interpreted:

a) ASCII files in which each byte represents a character according to the American Standard Code for Information Interchange (see Appendix A for a table of ASCII codes) and

b) binary files in which the binary data are taken as such with no code conversions applied. Two special types of binary files are distinguished from other binary files by their contents. Absolute files are those which conform to the Absolute Tape Dump format in Appendix B. The Monitor's SAV command produces absolute files. Relocatable files conform to the relocatable object code module format in Section 5-3. The Assembler produces relocatable files which the Linking Loader can then load into memory.

Random files are organized as a series of records, each of which may be accessed separately from the rest. Each record has a unique number which may be used to read, modify or write on any record in the file at any time.

The various system programs follow certain conventions for file names. See section 2-7 for an explanation of these conventions. Appendix E shows an example of the use of files in a DOS program.

program — an ordered set of machine and/or Assembler instructions that direct the computer to perform a given series of operations. The two major classes of programs are system programs and user programs.

a) <u>system programs</u> are stored on disk in absolute binary files and thus may be loaded and run simply by typing the program's name to the Monitor. System programs run in memory immediately above the Monitor and below user programs.

b) <u>user programs</u> are those programs that run in high memory above the system programs. The usual procedure for developing user programs is to construct them from one or more relocatable code modules produced by the Assembler and linked together by the Linking Loader. For a discussion of relocatable modules, see Section 5-3, page 77.

prompt — When the Monitor or a system program takes control, it prints a message indicating which program is running and whether it is ready to receive commands. The Monitor prompts with a period (.) which precedes each command. Similarly, Editor and Linking Loader commands are typed after an asterisk (*). Debug and the Assembler prompt only once after the program is loaded.

The Monitor also prompts the programmer when insufficient information has been given in a command. For example, if the programmer types

    .MNT <cr>

the computer prints

    ENTER DEVICE NUMBER

Typing the number and a carriage return causes the command to be executed.

1-5. DOS Input Conventions

All input to DOS (as from a terminal) is handled through the Monitor's input routine. This routine has several properties which set constraints on the form of input.

All 128 ASCII characters are accepted by the input routine except characters of the form Control/x where x is any letter. Some Control/ characters are used to control the input routine and the rest are ignored.

<cr> terminates a line. The input buffer is cleared and subsequent input is taken as a new line. <line feed> is considered an input character.

The input buffer accepts the first 72 characters as one line of input. If more than 72 characters are input in a line, the contents of the buffer are discarded and a new line is begun.

Special characters include the following:

a) Rubout deletes the last character in the buffer. When Rubout is typed, a backslash (\) and the last character in the buffer are printed. Each successive Rubout prints the previous character. Typing another character prints another backslash and the character. All of the characters between the backslashes are deleted. If Rubout is typed with no characters in the buffer, a <cr> is printed.

b) Control/U deletes the current contents of the input buffer.

c) Control/R displays the current contents of the input buffer.

   Example:

   EXAMPLE LENE\ENE\INE <Control/R>

   EXAMPLE LINE

   Typing three rubouts deleted the characters between the backslashes. Typing Control/R displayed the final appearance of the line.

d) Control/I is a tab character. When a tab is printed, spaces are printed so that the next character is printed at the start of the next 8 space column.

The following special characters are recognized if input interrupts are enabled (see p. 22).

Control/S — Causes execution of a program to pause until Control/Q is typed. This can be used to pause during a listing or to pause during execution of a program to examine intermediate values.

Control/Q — causes execution to resume after a Control/S. Control/Q has no effect if no Control/S has been typed.

Control/C — causes execution of a program to be suspended and control to be passed to the Monitor. During the execution of certain I/O operations (Mount, Open, Kill, etc.), Control/C does not terminate execution until the operation is completed.

Control/O — prevents output from the computer. Execution proceeds normally, but no output is generated until either another Control/O is typed or another command is requested by the Monitor or Editor. Example: Suppose the following Editor command is typed:

```
*P
00100    LDA    IER
00200    LHLD   CAND
<Control/O>
*
```

The Print command action is completed, but no output appears on the terminal until the Editor's prompt asterisk appears, requesting another command.

Other constraints are imposed by the system programs in use and are discussed in the descriptions of the Editor, Assembler, Debug and miscellaneous programs. Some of the standards which apply to all of the system programs are as follows:

a) All commands must be typed in upper case.
b) The fields of the command are separated by delimiters. These delimiters include space, tab, comma, semicolon and colon. Colons are used specifically to separate multiple commands on a single line.

ALTAIR DOS DOCUMENTATION
SECTION II
MONITOR

2. THE MONITOR

2-1. Introduction to the Monitor

The Monitor is the control center of the DOS system. It is used to load and execute system and user programs and to execute Input/Output routines for all of the system's peripheral devices.

The Monitor is loaded first to load and execute all the other system components. It remains in memory at all times, passing control back and forth to system and user programs and providing I/O services.

The Monitor's device-independent Input/Output system reduces programming effort. The programmer could write a different input or output routine for each I/O device used by a program. But these device handler routines are incorporated into the Monitor, so the programmer can perform the desired information transfer simply by calling the Monitor. Monitor Calls are described in detail in Appendix C.

When DOS has been loaded and initialized, the Monitor starts up and prints the following message.

DOS MONITOR VER x.x

.

This message is also printed when the Monitor is entered from another program. The period indicates that the Monitor is ready to receive commands.

2-2. Input from the Console

Input from the console keyboard is handled by a central Monitor routine regardless of the system program that is running at the time. This routine provides the following special characters and functions.

Rubout	deletes the last character in the input buffer. Typing Rubout causes a backslash (\) and the last character in the buffer to be printed. Subsequent Rubouts print the immediately previous character in the buffer. When a character other than Rubout is typed, a second backslash and the character are printed. All the characters between the backslashes are deleted.
Backarrow (←)	same as Rubout

Control/R causes the current contents of the input buffer to be printed on the console. Example:

EXEMPLE LINE\ENIL ELPME\AMPLE<Control/R>
EXAMPLE

In this example, typing Rubout 10 times deleted the characters between the backslashes; typing Control/R displays the current appearance of the line.

Control/U clears the input buffer.

<cr> terminates a line of input. The current contents of the line buffer are passed to the program and the line buffer is cleared.

If input interrupts are enabled, the following special character functions are available:

Control/C suspends execution of the current program and returns control to the Monitor.

Control/S temporarily suspends execution of a program until Control/Q is typed.

Control/Q causes execution of a program to be resumed after a Control/S

Control/O allows execution to proceed normally, but prevents output to the terminal. No output is printed until another Control/O is typed or another command is requested by the Monitor or Editor.

To enable interrupts on the older I/O interface boards (PIO, SIO A, B, C), install a jumper from the IN interrupt line to PINT or, if the Vector Interrupt board is in use, to VI7.

On newer interface boards (2SIO, 4PIO), install the jumper between PINT or VI7 to the interrupt request line for the input channel. DOS automatically assures that input interrupts are enabled.

For more information, see the manual for the interface board in use.

2-3. Monitor Commands

The Monitor is directed to perform its functions by commands. The general form of a Monitor command is as follows:

<command code> [<field> <field> . . .]

where the command code is the three letter designation of the command to be performed and the fields are the required operands for the specific command. The fields are separated by spaces, tabs or other legal delimiters. If insufficient information is given in the operand fields for a given command, the Monitor asks for the missing information and will not proceed until the information is typed. If the Monitor cannot execute the requested command, it prints an error message which indicates the reason the command could not be executed.

The following abbreviations and definitions are used in the descriptions of the Monitor commands:

delimiter	characters that separate the fields in a command. Legal delimiters are <space>, tab (Control/I), comma, semicolon and colon.
device	number of the device to be used in the command action. The Monitor at present supports only floppy disk drives in the commands, so the term "device" is interchangeable with the term "drive number."
file	name of the data or program file on which the command action is to be performed.
list	a series of device numbers or file names separated by delimiters.

Table 2-A. Monitor Commands

Command	Function
DEL <file><device>	deletes the named file from the indicated device.
DIN <device><list>	initializes the listed disk drives by writing the track and sector number in each sector. Zeros are written into each byte of each sector, destroying any existing files and marking each sector as free. The DOS disk is initialized at the factory and must not be initialized again. Doing so will destroy all system programs as well as user files.

Command	Function
DIR <device>	Prints a directory of the files on the indicated device. See section 2-7 for an explanation of the file name conventions.
DSM <device list>	Dismounts the disks on the listed device or devices. A disk must be dismounted before it is removed from a drive. Failure to do so may cause file link errors the next time the disk is read.
LOA <file><device>	Loads the named file into memory from the specified device. The file must be an absolute binary file. The LOA command automatically adds # to the file name.
MNT <device list>	Mounts the disks on the specified devices. The MNT command causes the system to read each specified diskette and creates a table of unused space. When files are created or modified, the system checks the table for unused sectors. This command must be given before the files on a disk may be accessed.
REN <old name> <new name> <device>	Renames the file <old name> on the specified device to have a name <new name>.
RUN <file><device>	Loads the named file from the specified device and runs it. The file must be an absolute binary file. A # sign is automatically added to the file name.
SAV <file><device> <1st location> <last location><sa>	Contents of memory from the first location to the last location are saved as an absolute binary file With the specified name. A # sign is automatically added to the file name. Any subsequent RUN command causes execution to begin at <sa>.

If the input to the Monitor is not one of these commands, the Monitor searches disk drive 0 for an absolute program file which has a name corresponding to the input. If such a file is found, it is loaded and run. The following system programs are run in this manner:

ASM	Assembler - see chapter 4
EDIT	Text Editor - see chapter 3
DEBUG	Debug package - see chapter 6
LINK	Linking Loader - see chapter 5
INIT	Disk initialization program - see chapter 7
CNS	Console - see chapter 7. Console allows the Monitor command console to be changed to another terminal.

Drive 0 must be mounted before running these programs.

2-4. Monitor Error Messages

When the Monitor detects an error in the execution of a command or a Monitor Call, it prints an error message and terminates execution of the operation. In the case of an error in a Monitor Call, the error message is printed and control returns to the calling program.

A Monitor error message contains the following information:

Error Code	the error codes are given in Table 2-B
File Number	the number of the file that was being accessed when the error occurred
RQCB Address	the address of the Request Control Block of the Monitor Call that caused the error.
Opcode	the operation code of the Monitor Call that caused the error
Return Address	the address to which control would have returned had the error not occurred.

Table 2-B. Error Codes

Error Code	Meaning
1	FILE TABLE ENTRY MISSING

The file table contains entries for thirteen disk files (numbered 0 - 12) and four other I/O files (0 - 3). If a file number other than these is encountered, an error occurs.

2	DEVICE NOT IN PHYSICAL DEVICE TABLE

The following devices are listed in the physical device table:

- Teletype or Teletype compatible terminal
- Audio Cassette
- High-Speed Paper Tape Reader
- Floppy Disk

An attempt to transfer information to or from another device causes an error.

3 HANDLER NOT IN HANDLER TABLE

An attempt was made to perform an invalid operation on an I/O device, for example, to output to a paper tape reader.

4 BOARD NOT IN I/O TABLE

The following I/O boards are in the I/O table:

- 2SIO
- SIO A, B, and C
- 4PIO
- PIO

Use of other boards is not supported.

5 SHORT DATA TRANSFER

The end of data transfer came before the specified number of bytes was read or written.

6 CHECKSUM ERROR

When a program is loaded, the Monitor keeps a running sum of all the bytes in each record. The least significant byte of this sum is the checksum. At the end of the record, it is compared with the checksum byte in the record. If there is a discrepancy between them, an error has occurred in loading the program and the Checksum Error message is printed.

7 MEMORY ERROR

An attempt was made to write into a bad memory location. This could be a non-functioning read/write memory location or a location in read-only memory.

10 BAD FILE NUMBER

A bad file number is one which has not been opened or which is greater than the number of files allocated at initialization.

11 FILE LINK ERROR

During a disk file read, a sector was read which did not belong to the file. A FILE LINK ERROR often occurs after a disk has been removed from a drive without being dismounted first.

12 I/O ERROR

A checksum error occurred in 18 successive disk read operations. A checksum error on a disk read causes the disk controller automatically to re-read the sector. A Disk I/O Error indicates that the error is a permanent defect in the file, disk or disk drive.

13 BAD FILE MODE

A sequential operation was attempted on a random file or vice versa.

14 DEVICE NOT OPEN

An attempt was made to input or output a file through a device which had not been opened to that file.

15 DEVICE NOT ENABLED

The door of a disk drive has not been closed, or the motor of the drive has not had time to come up to full speed.

16 DEVICE ALREADY OPEN

An attempt was made to mount a disk which has already been mounted.

17 INTERNAL ERROR

DOS became confused. Please report the circumstances of this error to the MITS, Inc. Software Department.

20 OUT OF RANDOM BLOCKS

All sectors allotted for random files have been filled.

21 FILE ALREADY OPEN

An open operation was attempted on a file that was already open.

22 FILE NOT FOUND

The file name referred to was not found on the specified device.

23 TOO MANY FILES

An attempt was made to create a file when the disk directory was already full.

24 MODE MISMATCH

A command that expected a character string operand received a number, or vice-versa. This error often occurs when the quotation marks are left out of a character string in a command.

25 END OF FILE

During a read operation, an end of file mark was encountered before the read operation was complete.

26 DISK FULL

All of the sectors of the disk have been used.

27 BAD RECORD NUMBER

An attempt was made to refer to a random file record that was not in the specified file.

30 FILE TABLE FULL

An attempt was made to have more than thirteen disk files or four I/O files open at one time.

31 Unused

32 TOO MANY OPEN DISK FILES

An attempt was made to open more disk files than were specified at initialization.

33 FILE ALREADY EXISTS

An attempt was made to name or rename a file with a name that already exists in the directory.

2-5. File Name Conventions

When a directory of disk files is listed by the DIR command, the file names are preceded by special characters that denote the file type. These characters and their meanings are as follows:

#	absolute binary files. Files with this character are produced by the Monitor's SAV command and are used as input by the LOA and RUN commands. System program names appear in the directory with a pound sign (#).
*	relocatable load module. These files are output by the Assembler and used as input by the Linking Loader.
%	listing file. The optional source listing from ASM carries this designation.
&	Editor source file. The output of the Editor carries this designation.
$	Editor backup file. When a file is modified by the Editor, the old, unmodified file is renamed to have this designation.

These characters are supplied automatically by the system programs and Monitor commands which create the files. Therefore, they need not be supplied by the programmer. For example, the command

    .ASM MULTI 0

is used to assemble the file which appears in the directory as

    &MULTI

Similarly, the command

    .EDIT TEXT 0

creates a source file called &TEXT.

File names in the DEL and REN commands must appear exactly as they do in the directory. For example, the Editor backup file

    $LETTER

may be deleted by

    .DEL $LETTER

without affecting the source file &LETTER or any other file.

ALTAIR DOS DOCUMENTATION
SECTION III
TEXT EDITOR

3. THE TEXT EDITOR

3-1. Introduction

Although the Text Editor is primarily used to create and maintain Assembly Language program files, it can be used for any ASCII coded file. EDIT is a line-oriented Editor, in that its commands operate on lines of text which are addressable by number. Line numbers are assigned automatically as the file is being created. A special command allows automatic renumbering of lines. The Assembler ignores EDIT line numbers in its input file except when producing a source listing.

Once the system disk (on drive 0) has been mounted with the MNT command, EDIT may be loaded and run with the following command:

.EDIT <file><device>

where <file> is the name of the file to be created or modified, and <device> is the number of the disk where the file is stored. When EDIT prints an asterisk (*), it is ready to accept commands. EDIT requires at least 2 disk files to be allocated at initialization.

The Text Editor is designed to minimize memory usage by dividing files into pages. Only one page resides in memory at a time, while the rest of the file remains on disk. The number, length and content of pages are completely under the programmer's control. Access to the pages is sequential; the paging commands refer to the next page in the file. The B command always refers to the first page of the file, so the Editor can go back to the beginning of a multipage file from any point.

Edit commands are provided to add, delete and replace lines, find and substitute character strings and modify individual lines. The form of an EDIT command is as follows:

<x> <field>[<field>] <cr>

where x stands for the EDIT command letter in use, and field is a line number or character string, depending upon the command. The command letter and fields are separated by delimiters.

The EDIT commands operate on individual lines or on ranges of lines. A line is referenced by stating its number in an EDIT command. For example,

P 150

prints line 150 on the console. A range of lines is referenced by stating the beginning and ending lines of the range. Thus,

R 200 230

replaces lines 200 to 230, inclusive. All line and range references are to lines on the current page only. Before a line or range on another page may be referenced, that page must be loaded into memory.

3-2. Edit Commands

A. Inserting, Deleting and Replacing lines. The following commands insert, delete and replace whole lines:

I <number><increment><cr>	Inserts a new line at <number> or the first available line after <number>. After the <cr>, EDIT prints <number> or, if there is already a line at <number>, the number of the first available line after <number>. All input up to the next <cr> is inserted as the new line. In the Insert mode, the Editor automatically assigns numbers to the lines as they are entered. If <increment> is not specified, the line number increment is that last used in an N command. If there has been no previous N command, the default increment is 10. After a line is typed and a carriage return entered, EDIT adds the increment and checks to see that the new line number is less than the next existing line number. If it is not, the increment is reduced to half the difference between

the previous line number and the next existing line number. This process is repeated until no new line numbers are possible. Then the Insert mode is exited and an asterisk is printed. When a file is being created by the Editor, there are no existing lines, so each line is numbered with the specified or default increment.

Example:

```
.EDIT TEST 0
DOS EDITOR VER 0.1
CREATING TEST
00100 THIS IS A TEST <cr>
00110 FILE SHOWING LINE <cr>
00120 NUMBER INCREMENTS <cr>
00130 <cr>
*
```

In this example, new line numbers were generated after every carriage return until a null line (a line with no characters before the carriage return) was typed. Then Insert mode was terminated and the prompt asterisk printed. In the following example, insertions are made into file TEST:

```
*I 110
00115 INSERT ONE <cr>
00117 INSERT TWO <cr>
00118 INSERT THREE <cr>
00119 INSERT FOUR <cr>
*
```

In each case, the increment was halved, until it was not possible to insert another line.

D <1st number> [<2nd number>] <cr>

Deletes all lines from <1st number> to <2nd number>, inclusive. If <2nd number> is omitted, one line is deleted.

R <1st number> <2nd number> <cr>

Replaces the lines from <1st number> to <2nd number>, inclusive, with input from the console. After the <cr>, EDIT displays the number of the first line to be replaced. All input to the next <cr>, replaces the line. After the next <cr>, the number of the next line to be replaced is displayed. Typing a null line causes that line and the remaining lines in the range to be deleted. If <2nd number> is omitted, one line is replaced.

B. Finding a String. The following commands display the next occurrence of a character string:

F <string> <cr>

Finds the next occurrence of <string> on the current page. If <string> is found, the line in which it appears is printed. If it is not found, an asterisk is printed and EDIT is ready for further commands. The search begins on the line immediately after the current line.

S <string> cr>

The same as F, except the search can extend over page boundaries.

C. In-Line Editing: the Alter Command. The Alter command allows adding, deleting or modifying characters within a line without affecting the other lines in the file. The format of the Alter command is as follows:

A <number> <cr>

where <number> is the number of the line to be altered. The Alter command allows the use of several subcommands which order changes to be made. The subcommand action begins with the next character to the right of the current position. Changes are made from left to right.

In the listing of subcommands below, 'n' preceding the subcommand letter means the subcommand may be preceded by a number which indicates the number of times the subcommand is to be repeated. For example:

3CABC

is equivalent to three subcommands

CA
CB
CC

in sequence.

The Alter subcommands are not echoed. When they are used, the only output from the computer is a display of the line as modified.

In the examples that follow, assume the following command has been executed:

A 100

where line 100 is in file TEST on page 35. The Alter subcommands are as follows:

Command	Explanation
n<space>	skips over and prints the next n characters in the line. Typing <space> displays 00100 T
nC<characters>	changes the next n characters in the line to the specified characters. Typing 3CHAT displays 00100 THAT
nD	deletes the next n characters. Typing D displays 00100 THAT and deletes the following space. The effect of the subcommand is not apparent until the next subcommand is executed.
H<string>	deletes the rest of the line and inserts the string in its place. The string is terminated either by <Escape> or by <cr>. (On some terminals, Altmode is used rather than Escape.) Terminating with <Escape> allows the Alter command to receive further subcommands. <cr> exits Alter mode. Typing H'S NO<Escape> displays 0100 THAT'S NO
I<string>	inserts the string before the next character. The string is terminated either by <Escape> (Altmode on some terminals) or by <cr>. Typing <Escape> allows further subcommands to be issued. Typing <cr> exits Alter mode. Typing ILINE <cr> displays

0100 THAT'S NO LINE

and exits Alter mode.

To demonstrate the remaining Alter subcommands, the command

*A 100 <cr>

is executed again. This command reenters Alter mode on the same line as before and moves the current position to the beginning of the line.

nK<character> — deletes everything up to (but not including) the nth occurrence of the character. If the character does not exist, or if there are fewer than n of them, the subcommand does nothing. Typing KO displays

0100

The effect of the subcommand is not apparent until the next subcommand is executed.

R<string> — replaces the next character with the string. The string is terminated by <Escape> or <cr>. Typing <cr> exits Alter mode. Typing RSOME <space> <Escape> displays

0100 SOME

nS<character> — skips over and prints all characters up to, but not including, the nth occurrence of <character>. If no such character exists, or if there are fewer than n of them, the subcommand does nothing. Typing SN displays

0100 SOME LI

X<string> — skips to the end of the line and inserts the string at that point. The string is terminated with <Escape> or <cr>. <Escape> allows further subcommands to be issued. <cr> exits Alter mode. Typing X, THAT! <cr> displays

0100 SOME LINE, THAT!

When all of the desired changes have been ordered, Alter command mode is exited with one of the following subcommands:

<cr>	replaces the existing line with the line as modified and exits Alter mode.
Q	exits Alter mode, but makes none of the ordered changes. The changes are lost.

D. Paging commands. The amount of memory used by the Text Editor may be minimized by dividing the file to be edited into pages and loading one page into memory at a time. Pages are manipulated by the following commands:

B	Loads the first page of the file into memory. Note that after a B command is issued, the line number is unpredictable. An additional command (such as P <number>) is needed to refer to any specific line on the page.
C	Loads the next page of the file into memory and saves the current page on disk.
L	Loads the next page into memory and deletes the current page
W <number>	Writes the lines currently in memory from the first to <number> onto disk as a page.

E. Miscellaneous commands:

N <increment>	Renumbers all of the lines in the file. The difference between successive line numbers is <increment>.

The first line number is always 100.

P [<first number> [<second number>]]	Prints all lines from the <1st number> to the <2nd number>, inclusive. If there is no second number, 1 line is printed. If no line numbers are given, the entire current page is printed.
E <file name> <device number>	As the Editor proceeds through the named file making changes, it copies the modified file into a temporary file called EDIT.TEM. When the E command is executed, the remaining unmodified lines of the file are copied into EDIT.TEM. This file is then assigned the name of the edited file. The first character of the original file name is changed to $. This provides a backup file. Any previous backup file is deleted. If a file name and device number are specified in the E command, EDIT proceeds to edit that file. Thus, another file may be edited without having to reload the Editor. If the file and device are not specified, control is passed to the Monitor.
Q <file name> <device number>	Q exits to the monitor without renaming any files. The changes made by the Editor are ignored. The Q command allows the user to abort an editing session without damaging any files. The file name and device number may be specified as in the E command to edit another file without having to reload the Editor.

ALTAIR DOS DOCUMENTATION
SECTION IV
ASSEMBLER

4. THE ASSEMBLER

The Assembler is a system program that translates programs from Assembly Language into machine language. In principle, machine language can be used to write programs for the computer. A machine language program is one in which the instructions to the computer are represented by binary numbers one, two or three bytes long. The practical problems of machine language programming, however, make its use virtually impossible for all but the simplest programs. First, it is difficult to remember all of the binary machine language codes and enter them into the computer without error. Second, machine language requires the programmer to remember all of the addresses in the program and refer to them explicitly. Finally, if a machine language program does not work as desired, it is extremely difficult to determine what went wrong.

Assembly language programming is preferable to machine language programming because it avoids all of these difficulties. Machine instructions are referred to in Assembly language by mnemonics that are descriptive of the operation and that are relatively easy to remember. Addresses can be specified explicitly, but they can also be referred to symbolically. That is, a memory location can be given a label and referred to subsequently simply by mentioning that label. Finally, Assembly language provides the programmer with a complement of error messages that make the process of debugging much easier than in machine language programming.

The DOS Assembler translates Assembly Language to machine language by means of a two step process. In the first step, the Assembler reads the Assembly Language program and assigns addresses to all of the symbols. In the second step, the program is read again and the instructions are converted to their machine language equivalents. On this second pass through the program, the program may be listed on the terminal or in a disk file. If the Assembler detects an error in the program, the place where the error occurred is marked in the listing with a letter that indicates the nature of the error.

Once the system disk is mounted in drive 0, the Assembler is run by typing the following command to the Monitor:

_ ASM <file name> <device> [<device type> <device number>]

where the <file name> is the name of the disk file that contains the source program and <device> is the number of the drive where that file resides. If a <device type> is specified, an Assembler listing is written in a file on the specified device. If the <device type> is TTY, the listing is printed on the terminal; if the <device type> is FDS, it is sent to floppy disk. The name of the listing disk file is the file name in the ASM command preceded by a percent sign (%). The following message is printed on the terminal upon termination of the assembly:

    xxxxx ERRORS DETECTED

where xxxxx is the number (in octal) of errors encountered in the program.

The machine language, object code module that results from the Assembler's action is written on the same disk as the source code. The name of the object code file is the <file name> preceded by an asterisk (*). For example, after the following command is executed:

    .ASM SOURCE 0 FDS 1

the object code file is named *SOURCE and is written on disk 0. The listing of the source program is named %SOURCE and resides on disk 1.

When the assembly and listing are complete, the Assembler prints

        ANY MORE ASSEMBLIES?

Typing "Y" causes the Assembler to start over and ask for the new file name, device number and listing file parameters. Thus, another file may be assembled without reloading the assembler. Typing N or <cr> exits the Assembler and returns control to the Monitor.

4-1. Statements

The fundamental unit of an Assembly Language program is the statement, whose form is as follows:

    [label]   <op-code>   <operand> [,<operand>]   [comment]

The label is a tag by which other statements in the program can refer to this statement. Not all statements in a program need to be labelled. Since program execution proceeds normally in order from the lowest memory location to the highest, statements that need to be executed in normal sequence need not carry labels. If, on the other hand, a statement needs to be executed out of normal order, it must carry a label. Such out-of-order execution is called branching and it is particularly important in programmed decision making and loops. Labels can also be used to refer to memory locations for storing data. This use will be discussed more fully in section 4-2B below.

The op-code is the mnemonic of the machine instruction or Assembler pseudo-operation to be performed by the statement. Machine instruction op-codes are translated by the Assembler into machine language instructions. Assembler pseudo-ops are not translated, but direct the Assembler itself to allocate storage areas, set up special addresses, etc.

The op-code is followed by one or more operands, depending upon the nature of the instruction. An operand is an address - specified in any one of several manners - where the computer is to find the data to be operated upon. In the case of an ADC (add with carry) instruction, for example, the operand is the address of the location whose contents are to be added to the accumulator. In the MOV (above) instruction, the two operands are the addresses of the location from which a data byte is to be taken and to which it is to be moved.

Comment may be added to the end of a statement if they are separated from the rest of the statement by a semicolon. Comments are ignored by the Assembler, but they do appear in the Assembler listing and may thus be used by the programmer for documentation and explanation.

4-2. Addresses

A program is a series of statements that are stored in memory and executed either in the order in which they are stored or in sequence directed by statements in the program itself. The data operated upon by the program or used to direct the program's actions is stored in memory and referred to by the addresses of the locations in which it is stored. Therefore, addresses are used both to control execution of the program and to manipulate data. Much of the versatility of the Assembly Language programming system in DOS results from the various ways in which addresses may be represented and modified.

The DOS Assembler recognizes addresses in three major forms; constants, labels and address expressions.

A. Constants. A constant is an address that is stated explicitly as a number. For example, the instruction

```
JMP 23000
```

causes execution to proceed from the location whose address is 23000 decimal. A constant address may be expressed in octal, decimal or hexadecimal notation.

1. Octal address constants are strings of octal characters (0 - 7) whose first character is zero. The allowable range of values is -01777777 to 01777777.

   Examples:

   0377

   01345

   017740

2. Decimal address constants are strings of decimal digits (0 - 9) without a leading zero. The allowed range is -65536 to 65536. Examples:

   255

   1024

   23000

3. Hexadecimal address constants have the following form:

   X'hhhh'

   where h is any hexadecimal digit (0 - 9, A - F). The allowed range is -X'FFFF' to X'FFFF'. Examples:

   X'F000'

   X'2300'

   X'00F'

4. Character address constants have the following form:

   "xx"

   where x is any ASCII character except ("). The characters are translated into binary according to their ASCII codes and the resulting two-byte quantity makes up the address.

   Examples:

   "A1"

   "BZ"

   "#"

B. Labels. When a statement is labelled, the label is entered into the symbol table in the Assembler along with the address of the statement. Any subsequent statement can then use the label to represent that address. Two types of labels can be used in the DOS Assembler; names and program points.

1. Names are strings of up to 6 alphanumeric characters. The first character must be a letter and the subsequent characters may be letters, numbers or dollar signs. Examples:

```
SHIFT
LBL1
A$OUT
```

The usual use of labels is to refer to a statement by name. For example:

```
.
.
.
SHIFT   RAR
        JNC     SHIFT
.
.
.
```

The operand of the jump instruction tells the computer to branch back to the RAR (rotate right) instruction if there is no carry out of the shift. If there is a carry, execution proceeds with the next instruction after the jump.

Data bytes can bear labels as well. For example:

```
        ADC     ADDEND
ADDEND  DB      255
```

These instructions add the contents of location ADDEND to the accumulator with carry. In this example, the contents of ADDEND have the value 255 decimal.

For the purposes of clarity and ease of use, names should be systematically applied. That is, they should be logically related to the statements or data locations they represent and should be easily distinguishable from other names in the program.

Sometimes, short branches and lcops require statements to be labelled, but those labels are not important to the whole program. Rather than filling up the symbol table with unique names, the programmer may prefer to label those statements with program points.

2. Program points are special labels with the following form:

   .x

   where x is any letter. A letter may be used any number of times in a single program. Unlike names, program points may be referred to in two ways. The program point reference -x refers to the most recently encountered program point with letter x. The program point reference +x refers to the next program point in the program with the letter x. Therefore, while any number of statements may be labelled with the same program point, a statement may only refer to the two program points bracketting it in the program.

C. Address Expressions. The DOS Assembler allows addresses to be specified relative to other addresses. For example, to refer to the fourth location after the location labelled LOC, the following expression can be used:

   LOC+4

   Expressions of this form are called address expressions. Address expressions may be comprised of any of the following:

   Name
   Constant
   Program point reference
   Address expression $\pm$ constant

   The sixteen bit values of the names, constants, program point references and address expressions are combined and truncated to 16 bits to form the value of the final address expression.

Example:

SHIFT+5

+A-010

LOC+X'F'

D. Special Addresses. The DOS Assembler allows certain addresses to be referred to directly with special notation.

* indicates the present contents of the location counter. That is, * refers to the address of the current instruction or the current data address.

Registers may be addressed symbolically by name. Therefore, such instructions as

```
MOV     H,A
```

are interpreted to refer to the correct registers.

E. Addressing Modes. The addresses of statements or data locations are specified in one of five different modes. The DOS Assembler addressing modes are Absolute, Relative, Common, Data and External.

Absolute addresses are the actual hardware addresses of the designated locations. Address constants in themselves (not in address expressions) refer to absolute mode addresses. If an absolute mode address is specified, all of the other addresses in the program must be relocated to fit it.

Relative addresses are relocated by the action of the Linking Loader. Unless otherwise specified, all symbolic addresses (names, program points, address expressions) are in Relative mode. To calculate a Relative mode address, the Assembler calculates a displacement which the Linking Loader adds to a relocation base address when the program is loaded. In this way, the loader can load the program anywhere in memory and all the addresses bear the correct relation to each other.

An External mode address is one that refers to a location in another program. A name must be mentioned in an EXT statement before it can be used as an External mode address. External addresses allow a program to use routines or data in another program.

Data and Common mode addresses refer to separate blocks of memory locations that may or may not be contiguous with the programs which make the references. Data mode addresses are so designated by being mentioned in a DAT statement. Common mode items are designated by CMN statements. The difference between Common and Data addresses is that Data addresses may only be referenced by the program in which they are defined, whereas Common mode addresses are available to any program. In addition, several Common blocks can exist simultaneously and be referred to by name.

In an address expression, the constituent addresses may have different modes. Any mode expression combined with an Absolute mode address has the mode of the expression. The difference of two expressions of the same mode is of Absolute mode.

4-3. Op-Codes

Op-codes are of two types. One type, the machine codes, are the mnemonic expressions of the 8080 instructions. These op-codes and their associated operands are discussed in section A, below, which is reprinted from the Intel 8080 Microcomputer System Users' Manual. The Assembler can use any address expression to derive the required address for direct or immediate addressing instructions. Register instructions can use any address expression as long as its value is the address of a register (0 - 7 absolute). Before a register indirect mode instruction may be used, the register pair must be loaded with an address. Any address expression can be used to supply that address.

CHAPTER 4
INSTRUCTION SET

A computer, no matter how sophisticated, can only do what it is "told" to do. One "tells" the computer what to do via a series of coded instructions referred to as a **Program**. The realm of the programmer is referred to as **Software**, in contrast to the **Hardware** that comprises the actual computer equipment. A computer's software refers to all of the programs that have been written for that computer.

When a computer is designed, the engineers provide the Central Processing Unit (CPU) with the ability to perform a particular set of operations. The CPU is designed such that a specific operation is performed when the CPU control logic decodes a particular instruction. Consequently, the operations that can be performed by a CPU define the computer's **Instruction Set**.

Each computer instruction allows the programmer to initiate the performance of a specific operation. All computers implement certain arithmetic operations in their instruction set, such as an instruction to add the contents of two registers. Often logical operations (e.g., OR the contents of two registers) and register operate instructions (e.g., increment a register) are included in the instruction set. A computer's instruction set will also have instructions that move data between registers, between a register and memory, and between a register and an I/O device. Most instruction sets also provide **Conditional Instructions.** A conditional instruction specifies an operation to be performed only if certain conditions have been met; for example, jump to a particular instruction if the result of the last operation was zero. Conditional instructions provide a program with a decision-making capability.

By logically organizing a sequence of instructions into a coherent program, the programmer can "tell" the computer to perform a very specific and useful function.

The computer, however, can only execute programs whose instructions are in a binary coded form (i.e., a series of 1's and 0's), that is called **Machine Code**. Because it would be extremely cumbersome to program in machine code, programming languages have been developed. There are programs available which convert the programming language instructions into machine code that can be interpreted by the processor.

One type of programming language is **Assembly Language**. A unique assembly language mnemonic is assigned to each of the computer's instructions. The programmer can write a program (called the **Source Program**) using these mnemonics and certain operands; the source program is then converted into machine instructions (called the **Object Code**). Each assembly language instruction is converted into one machine code instruction (1 or more bytes) by an **Assembler** program. Assembly languages are usually machine dependent (i.e., they are usually able to run on only one type of computer).

THE 8080 INSTRUCTION SET

The 8080 instruction set includes five different types of instructions:

- **Data Transfer Group**—move data between registers or between memory and registers
- **Arithmetic Group** – add, subtract, increment or decrement data in registers or in memory
- **Logical Group** – AND, OR, EXCLUSIVE-OR, compare, rotate or complement data in registers or in memory
- **Branch Group** – conditional and unconditional jump instructions, subroutine call instructions and return instructions
- **Stack, I/O and Machine Control Group** – includes I/O instructions, as well as instructions for maintaining the stack and internal control flags.

Instruction and Data Formats:

Memory for the 8080 is organized into 8-bit quantities, called Bytes. Each byte has a unique 16-bit binary address corresponding to its sequential position in memory.

The 8080 can directly address up to 65,536 bytes of memory, which may consist of both read-only memory (ROM) elements and random-access memory (RAM) elements (read/write memory).

Data in the 8080 is stored in the form of 8-bit binary integers:

DATA WORD

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
MSB							LSB

When a register or data word contains a binary number, it is necessary to establish the order in which the bits of the number are written. In the Intel 8080, BIT 0 is referred to as the **Least Significant Bit (LSB)**, and BIT 7 (of an 8 bit number) is referred to as the **Most Significant Bit (MSB)**.

The 8080 program instructions may be one, two or three bytes in length. Multiple byte instructions must be stored in successive memory locations; the address of the first byte is always used as the address of the instructions. The exact instruction format will depend on the particular operation to be executed.

Single Byte Instructions

	D_7 … D_0	Op Code

Two-Byte Instructions

Byte One	D_7 … D_0	Op Code
Byte Two	D_7 … D_0	Data or Address

Three-Byte Instructions

Byte One	D_7 … D_0	Op Code
Byte Two	D_7 … D_0	Data or Address
Byte Three	D_7 … D_0	

Addressing Modes:

Often the data that is to be operated on is stored in memory. When multi-byte numeric data is used, the data, like instructions, is stored in successive memory locations, with the least significant byte first, followed by increasingly significant bytes. The 8080 has four different modes for addressing data stored in memory or in registers:

- Direct – Bytes 2 and 3 of the instruction contain the exact memory address of the data item (the low-order bits of the address are in byte 2, the high-order bits in byte 3).
- Register – The instruction specifies the register or register-pair in which the data is located.
- Register Indirect – The instruction specifies a register-pair which contains the memory address where the data is located (the high-order bits of the address are in the first register of the pair, the low-order bits in the second).
- Immediate – The instruction contains the data itself. This is either an 8-bit quantity or a 16-bit quantity (least significant byte first, most significant byte second).

Unless directed by an interrupt or branch instruction, the execution of instructions proceeds through consecutively increasing memory locations. A branch instruction can specify the address of the next instruction to be executed in one of two ways:

- Direct – The branch instruction contains the address of the next instruction to be executed. (Except for the 'RST' instruction, byte 2 contains the low-order address and byte 3 the high-order address.)
- Register indirect – The branch instruction indicates a register-pair which contains the address of the next instruction to be executed. (The high-order bits of the address are in the first register of the pair, the low-order bits in the second.)

The RST instruction is a special one-byte call instruction (usually used during interrupt sequences). RST includes a three-bit field; program control is transferred to the instruction whose address is eight times the contents of this three-bit field.

Condition Flags:

There are five condition flags associated with the execution of instructions on the 8080. They are Zero, Sign, Parity, Carry, and Auxiliary Carry, and are each represented by a 1-bit register in the CPU. A flag is "set" by forcing the bit to 1; "reset" by forcing the bit to 0.

Unless indicated otherwise, when an instruction affects a flag, it affects it in the following manner:

Zero: If the result of an instruction has the value 0, this flag is set; otherwise it is reset.

Sign: If the most significant bit of the result of the operation has the value 1, this flag is set; otherwise it is reset.

Parity: If the modulo 2 sum of the bits of the result of the operation is 0, (i.e., if the result has even parity), this flag is set; otherwise it is reset (i.e., if the result has odd parity).

Carry: If the instruction resulted in a carry (from addition), or a borrow (from subtraction or a comparison) out of the high-order bit, this flag is set; otherwise it is reset.

Auxiliary Carry: If the instruction caused a carry out of bit 3 and into bit 4 of the resulting value, the auxiliary carry is set; otherwise it is reset. This flag is affected by single precision additions, subtractions, increments, decrements, comparisons, and logical operations, but is principally used with additions and increments preceding a DAA (Decimal Adjust Accumulator) instruction.

Symbols and Abbreviations:

The following symbols and abbreviations are used in the subsequent description of the 8080 instructions:

SYMBOLS	MEANING
accumulator	Register A
addr	16-bit address quantity
data	8-bit data quantity
data 16	16-bit data quantity
byte 2	The second byte of the instruction
byte 3	The third byte of the instruction
port	8-bit address of an I/O device
r,r1,r2	One of the registers A,B,C,D,E,H,L
DDD,SSS	The bit pattern designating one of the registers A,B,C,D,E,H,L (DDD=destination, SSS=source):

DDD or SSS	REGISTER NAME
111	A
000	B
001	C
010	D
011	E
100	H
101	L

rp — One of the register pairs:

B represents the B,C pair with B as the high-order register and C as the low-order register;

D represents the D,E pair with D as the high-order register and E as the low-order register;

H represents the H,L pair with H as the high-order register and L as the low-order register;

SP represents the 16-bit stack pointer register.

RP — The bit pattern designating one of the register pairs B,D,H,SP:

RP	REGISTER PAIR
00	B-C
01	D-E
10	H-L
11	SP

rh	The first (high-order) register of a designated register pair.
rl	The second (low-order) register of a designated register pair.
PC	16-bit program counter register (PCH and PCL are used to refer to the high-order and low-order 8 bits respectively).
SP	16-bit stack pointer register (SPH and SPL are used to refer to the high-order and low-order 8 bits respectively).
r_m	Bit m of the register r (bits are number 7 through 0 from left to right).
Z,S,P,CY,AC	The condition flags: Zero, Sign, Parity, Carry, and Auxiliary Carry, respectively.
( )	The contents of the memory location or registers enclosed in the parentheses.
←	"Is transferred to"
∧	Logical AND
∀	Exclusive OR
∨	Inclusive OR
+	Addition
−	Two's complement subtraction
*	Multiplication
↔	"Is exchanged with"
—	The one's complement (e.g., $(\overline{A})$)
n	The restart number 0 through 7
NNN	The binary representation 000 through 111 for restart number 0 through 7 respectively.

Description Format:

The following pages provide a detailed description of the instruction set of the 8080. Each instruction is described in the following manner:

1. The MAC 80 assembler format, consisting of the instruction mnemonic and operand fields, is printed in **BOLDFACE** on the left side of the first line.
2. The name of the instruction is enclosed in parenthesis on the right side of the first line.
3. The next line(s) contain a symbolic description of the operation of the instruction.
4. This is followed by a narative description of the operation of the instruction.
5. The following line(s) contain the binary fields and patterns that comprise the machine instruction.

6. The last four lines contain incidental information about the execution of the instruction. The number of machine cycles and states required to execute the instruction are listed first. If the instruction has two possible execution times, as in a Conditional Jump, both times will be listed, separated by a slash. Next, any significant data addressing modes (see Page 4-2) are listed. The last line lists any of the five Flags that are affected by the execution of the instruction.

Data Transfer Group:

This group of instructions transfers data to and from registers and memory. **Condition flags are not affected** by any instruction in this group.

MOV r1, r2 (Move Register)

(r1) ← (r2)

The content of register r2 is moved to register r1.

<table><tr><td>0</td><td>1</td><td>D</td><td>D</td><td>D</td><td>S</td><td>S</td><td>S</td></tr></table>

Cycles: 1
States: 5
Addressing: register
Flags: none

MOV r, M (Move from memory)

(r) ← ((H) (L))

The content of the memory location, whose address is in registers H and L, is moved to register r.

<table><tr><td>0</td><td>1</td><td>D</td><td>D</td><td>D</td><td>1</td><td>1</td><td>0</td></tr></table>

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

MOV M, r (Move to memory)

((H) (L)) ← (r)

The content of register r is moved to the memory location whose address is in registers H and L.

<table><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>S</td><td>S</td><td>S</td></tr></table>

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

MVI r, data (Move Immediate)

(r) ← (byte 2)

The content of byte 2 of the instruction is moved to register r.

<table><tr><td>0</td><td>0</td><td>D</td><td>D</td><td>D</td><td>1</td><td>1</td><td>0</td></tr><tr><td colspan="8">data</td></tr></table>

Cycles: 2
States: 7
Addressing: immediate
Flags: none

MVI M, data (Move to memory immediate)

((H) (L)) ← (byte 2)

The content of byte 2 of the instruction is moved to the memory location whose address is in registers H and L.

<table><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td colspan="8">data</td></tr></table>

Cycles: 3
States: 10
Addressing: immed./reg. indirect
Flags: none

LXI rp, data 16 (Load register pair immediate)

(rh) ← (byte 3),

(rl) ← (byte 2)

Byte 3 of the instruction is moved into the high-order register (rh) of the register pair rp. Byte 2 of the instruction is moved into the low-order register (rl) of the register pair rp.

<table><tr><td>0</td><td>0</td><td>R</td><td>P</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td colspan="8">low-order data</td></tr><tr><td colspan="8">high-order data</td></tr></table>

Cycles: 3
States: 10
Addressing: immediate
Flags: none

LDA addr (Load Accumulator direct)

(A) ← ((byte 3)(byte 2))

The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to register A.

0	0	1	1	1	0	1	0
low-order addr							
high-order addr							

Cycles: 4
States: 13
Addressing: direct
Flags: none

STA addr (Store Accumulator direct)

((byte 3)(byte 2)) ← (A)

The content of the accumulator is moved to the memory location whose address is specified in byte 2 and byte 3 of the instruction.

0	0	1	1	0	0	1	0
low-order addr							
high-order addr							

Cycles: 4
States: 13
Addressing: direct
Flags: none

LHLD addr (Load H and L direct)

(L) ← ((byte 3)(byte 2))
(H) ← ((byte 3)(byte 2) + 1)

The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to register L. The content of the memory location at the succeeding address is moved to register H.

0	0	1	0	1	0	1	0
low-order addr							
high-order addr							

Cycles: 5
States: 16
Addressing: direct
Flags: none

SHLD addr (Store H and L direct)

((byte 3)(byte 2)) ← (L)
((byte 3)(byte 2) + 1) ← (H)

The content of register L is moved to the memory location whose address is specified in byte 2 and byte 3. The content of register H is moved to the succeeding memory location.

0	0	1	0	0	0	1	0
low-order addr							
high-order addr							

Cycles: 5
States: 16
Addressing: direct
Flags: none

LDAX rp (Load accumulator indirect)

(A) ← ((rp))

The content of the memory location, whose address is in the register pair rp, is moved to register A. Note: only register pairs rp=B (registers B and C) or rp=D (registers D and E) may be specified.

0	0	R	P	1	0	1	0

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

STAX rp (Store accumulator indirect)

((rp)) ← (A)

The content of register A is moved to the memory location whose address is in the register pair rp. Note: only register pairs rp=B (registers B and C) or rp=D (registers D and E) may be specified.

0	0	R	P	0	0	1	0

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

XCHG (Exchange H and L with D and E)

(H) ↔ (D)
(L) ↔ (E)

The contents of registers H and L are exchanged with the contents of registers D and E.

1	1	1	0	1	0	1	1

Cycles: 1
States: 4
Addressing: register
Flags: none

Arithmetic Group:

This group of instructions performs arithmetic operations on data in registers and memory.

Unless indicated otherwise, all instructions in this group affect the Zero, Sign, Parity, Carry, and Auxiliary Carry flags according to the standard rules.

All subtraction operations are performed via two's complement arithmetic and set the carry flag to one to indicate a borrow and clear it to indicate no borrow.

ADD r (Add Register)

(A) ← (A) + (r)

The content of register r is added to the content of the accumulator. The result is placed in the accumulator.

1	0	0	0	0	S	S	S

Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ADD M (Add memory)

(A) ← (A) + ((H) (L))

The content of the memory location whose address is contained in the H and L registers is added to the content of the accumulator. The result is placed in the accumulator.

1	0	0	0	0	1	1	0

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ADI data (Add immediate)

(A) ← (A) + (byte 2)

The content of the second byte of the instruction is added to the content of the accumulator. The result is placed in the accumulator.

1	1	0	0	0	1	1	0
data							

Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

ADC r (Add Register with carry)

(A) ← (A) + (r) + (CY)

The content of register r and the content of the carry bit are added to the content of the accumulator. The result is placed in the accumulator.

1	0	0	0	1	S	S	S

Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ADC M (Add memory with carry)

(A) ← (A) + ((H) (L)) + (CY)

The content of the memory location whose address is contained in the H and L registers and the content of the CY flag are added to the accumulator. The result is placed in the accumulator.

1	0	0	0	1	1	1	0

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ACI data (Add immediate with carry)

(A) ← (A) + (byte 2) + (CY)

The content of the second byte of the instruction and the content of the CY flag are added to the contents of the accumulator. The result is placed in the accumulator.

1	1	0	0	1	1	1	0
data							

Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

SUB r (Subtract Register)

(A) ← (A) – (r)

The content of register r is subtracted from the content of the accumulator. The result is placed in the accumulator.

1	0	0	1	0	S	S	S

Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

SUB M (Subtract memory)

(A) ← (A) – ((H) (L))

The content of the memory location whose address is contained in the H and L registers is subtracted from the content of the accumulator. The result is placed in the accumulator.

1	0	0	1	0	1	1	0

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

SUI data (Subtract immediate)

(A) ← (A) – (byte 2)

The content of the second byte of the instruction is subtracted from the content of the accumulator. The result is placed in the accumulator.

1	1	0	1	0	1	1	0
data							

Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

SBB r (Subtract Register with borrow)

(A) ← (A) – (r) – (CY)

The content of register r and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

1	0	0	1	1	S	S	S

Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

SBB M (Subtract memory with borrow)

(A) ← (A) – ((H) (L)) – (CY)

The content of the memory location whose address is contained in the H and L registers and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

1	0	0	1	1	1	1	0

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

SBI data (Subtract immediate with borrow)

(A) ← (A) – (byte 2) – (CY)

The contents of the second byte of the instruction and the contents of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

1	1	0	1	1	1	1	0
data							

Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

INR r (Increment Register)

(r) ← (r) + 1

The content of register r is incremented by one. Note: All condition flags **except CY** are affected.

0	0	D	D	D	1	0	0

Cycles: 1
States: 5
Addressing: register
Flags: Z,S,P,AC

INR M (Increment memory)

((H) (L)) ← ((H) (L)) + 1

The content of the memory location whose address is contained in the H and L registers is incremented by one. Note: All condition flags **except CY** are affected.

0	0	1	1	0	1	0	0

Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,AC

DCR r (Decrement Register)

(r) ← (r) – 1

The content of register r is decremented by one. Note: All condition flags **except CY** are affected.

0	0	D	D	D	1	0	1

Cycles: 1
States: 5
Addressing: register
Flags: Z,S,P,AC

DCR M (Decrement memory)

((H) (L)) ← ((H) (L)) – 1

The content of the memory location whose address is contained in the H and L registers is decremented by one. Note: All condition flags **except CY** are affected.

0	0	1	1	0	1	0	1

Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,AC

INX rp (Increment register pair)

(rh) (rl) ← (rh) (rl) + 1

The content of the register pair rp is incremented by one. Note: **No condition flags are affected.**

0	0	R	P	0	0	1	1

Cycles: 1
States: 5
Addressing: register
Flags: none

DCX rp (Decrement register pair)

(rh) (rl) ← (rh) (rl) – 1

The content of the register pair rp is decremented by one. Note: **No condition flags are affected.**

0	0	R	P	1	0	1	1

Cycles: 1
States: 5
Addressing: register
Flags: none

DAD rp (Add register pair to H and L)

(H) (L) ← (H) (L) + (rh) (rl)

The content of the register pair rp is added to the content of the register pair H and L. The result is placed in the register pair H and L. Note: Only the **CY flag is affected.** It is set if there is a carry out of the double precision add; otherwise it is reset.

0	0	R	P	1	0	0	1

Cycles: 3
States: 10
Addressing: register
Flags: CY

DAA (Decimal Adjust Accumulator)

The eight-bit number in the accumulator is adjusted to form two four-bit Binary-Coded-Decimal digits by the following process:

1. If the value of the least significant 4 bits of the accumulator is greater than 9 or if the AC flag is set, 6 is added to the accumulator.
2. If the value of the most significant 4 bits of the accumulator is now greater than 9, or if the CY flag is set, 6 is added to the most significant 4 bits of the accumulator.

NOTE: All flags are affected.

0	0	1	0	0	1	1	1

Cycles: 1
States: 4
Flags: Z,S,P,CY,AC

Logical Group:

This group of instructions performs logical (Boolean) operations on data in registers and memory and on condition flags.

Unless indicated otherwise, all instructions in this group affect the Zero, Sign, Parity, Auxiliary Carry, and Carry flags according to the standard rules.

ANA r (AND Register)

(A) ← (A) ∧ (r)

The content of register r is logically anded with the content of the accumulator. The result is placed in the accumulator. **The CY flag is cleared.**

1	0	1	0	0	S	S	S

Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ANA M (AND memory)

(A) ← (A) ∧ ((H) (L))

The contents of the memory location whose address is contained in the H and L registers is logically anded with the content of the accumulator. The result is placed in the accumulator. **The CY flag is cleared.**

1	0	1	0	0	1	1	0

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ANI data (AND immediate)

(A) ← (A) ∧ (byte 2)

The content of the second byte of the instruction is logically anded with the contents of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

1	1	1	0	0	1	1	0
data							

Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

XRA r (Exclusive OR Register)

(A) ← (A) ⊻ (r)

The content of register r is exclusive-or'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

1	0	1	0	1	S	S	S

Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

XRA M (Exclusive OR Memory)

(A) ← (A) ⊻ ((H) (L))

The content of the memory location whose address is contained in the H and L registers is exclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

1	0	1	0	1	1	1	0

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

XRI data (Exclusive OR immediate)

(A) ← (A) ⊻ (byte 2)

The content of the second byte of the instruction is exclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

1	1	1	0	1	1	1	0
data							

Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

ORA r (OR Register)

(A) ← (A) V (r)

The content of register r is inclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

1	0	1	1	0	S	S	S

Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ORA M (OR memory)

(A) ← (A) V ((H) (L))

The content of the memory location whose address is contained in the H and L registers is inclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

1	0	1	1	0	1	1	0

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ORI data (OR Immediate)

(A) ← (A) V (byte 2)

The content of the second byte of the instruction is inclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

1	1	1	1	0	1	1	0
data							

Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

CMP r (Compare Register)

(A) – (r)

The content of register r is subtracted from the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. **The Z flag is set to 1 if (A) = (r). The CY flag is set to 1 if (A) < (r).**

1	0	1	1	1	S	S	S

Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

CMP M (Compare memory)

(A) – ((H) (L))

The content of the memory location whose address is contained in the H and L registers is subtracted from the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. The Z flag is set to 1 if (A) = ((H) (L)). The CY flag is set to 1 if (A) < ((H) (L)).

1	0	1	1	1	1	1	0

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

CPI data (Compare immediate)

(A) – (byte 2)

The content of the second byte of the instruction is subtracted from the accumulator. The condition flags are set by the result of the subtraction. The Z flag is set to 1 if (A) = (byte 2). The CY flag is set to 1 if (A) < (byte 2).

1	1	1	1	1	1	1	0
data							

Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

RLC (Rotate left)

$(A_{n+1}) \leftarrow (A_n)$; $(A_0) \leftarrow (A_7)$
$(CY) \leftarrow (A_7)$

The content of the accumulator is rotated left one position. The low order bit and the CY flag are both set to the value shifted out of the high order bit position. **Only the CY flag is affected.**

0	0	0	0	0	1	1	1

Cycles: 1
States: 4
Flags: CY

RRC (Rotate right)

$(A_n) \leftarrow (A_{n-1})$; $(A_7) \leftarrow (A_0)$
$(CY) \leftarrow (A_0)$

The content of the accumulator is rotated right one position. The high order bit and the CY flag are both set to the value shifted out of the low order bit position. **Only the CY flag is affected.**

0	0	0	0	1	1	1	1

Cycles: 1
States: 4
Flags: CY

RAL (Rotate left through carry)

$(A_{n+1}) \leftarrow (A_n)$; $(CY) \leftarrow (A_7)$
$(A_0) \leftarrow (CY)$

The content of the accumulator is rotated left one position through the CY flag. The low order bit is set equal to the CY flag and the CY flag is set to the value shifted out of the high order bit. **Only the CY flag is affected.**

0	0	0	1	0	1	1	1

Cycles: 1
States: 4
Flags: CY

RAR (Rotate right through carry)

$(A_n) \leftarrow (A_{n+1})$; $(CY) \leftarrow (A_0)$
$(A_7) \leftarrow (CY)$

The content of the accumulator is rotated right one position through the CY flag. The high order bit is set to the CY flag and the CY flag is set to the value shifted out of the low order bit. **Only the CY flag is affected.**

0	0	0	1	1	1	1	1

Cycles: 1
States: 4
Flags: CY

CMA (Complement accumulator)

$(A) \leftarrow (\overline{A})$

The contents of the accumulator are complemented (zero bits become 1, one bits become 0). **No flags are affected.**

0	0	1	0	1	1	1	1

Cycles: 1
States: 4
Flags: none

CMC (Complement carry)

(CY) ← $(\overline{CY})$

The CY flag is complemented. No other flags are affected.

0	0	1	1	1	1	1	1

Cycles: 1
States: 4
Flags: CY

STC (Set carry)

(CY) ← 1

The CY flag is set to 1. No other flags are affected.

0	0	1	1	0	1	1	1

Cycles: 1
States: 4
Flags: CY

Branch Group:

This group of instructions alter normal sequential program flow.

Condition flags are not affected by any instruction in this group.

The two types of branch instructions are unconditional and conditional. Unconditional transfers simply perform the specified operation on register PC (the program counter). Conditional transfers examine the status of one of the four processor flags to determine if the specified branch is to be executed. The conditions that may be specified are as follows:

CONDITION		CCC
NZ	— not zero (Z = 0)	000
Z	— zero (Z = 1)	001
NC	— no carry (CY = 0)	010
C	— carry (CY = 1)	011
PO	— parity odd (P = 0)	100
PE	— parity even (P = 1)	101
P	— plus (S = 0)	110
M	— minus (S = 1)	111

JMP addr (Jump)

(PC) ← (byte 3) (byte 2)

Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.

1	1	0	0	0	0	1	1
low-order addr							
high-order addr							

Cycles: 3
States: 10
Addressing: immediate
Flags: none

Jcondition addr (Conditional jump)

If (CCC),

(PC) ← (byte 3) (byte 2)

If the specified condition is true, control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction; otherwise, control continues sequentially.

1	1	C	C	C	0	1	0
low-order addr							
high-order addr							

Cycles: 3
States: 10
Addressing: immediate
Flags: none

CALL addr (Call)

((SP) − 1) ← (PCH)

((SP) − 2) ← (PCL)

(SP) ← (SP) − 2

(PC) ← (byte 3) (byte 2)

The high-order eight bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order eight bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by 2. Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.

1	1	0	0	1	1	0	1
low-order addr							
high-order addr							

Cycles: 5
States: 17
Addressing: immediate/reg. indirect
Flags: none

Ccondition addr (Condition call)

If (CCC),

((SP) – 1) ← (PCH)

((SP) – 2) ← (PCL)

(SP) ← (SP) – 2

(PC) ← (byte 3) (byte 2)

If the specified condition is true, the actions specified in the CALL instruction (see above) are performed; otherwise, control continues sequentially.

1	1	C	C	C	1	0	0
low-order addr							
high-order addr							

Cycles: 3/5
States: 11/17
Addressing: immediate/reg. indirect
Flags: none

RET (Return)

(PCL) ← ((SP));

(PCH) ← ((SP) + 1);

(SP) ← (SP) + 2;

The content of the memory location whose address is specified in register SP is moved to the low-order eight bits of register PC. The content of the memory location whose address is one more than the content of register SP is moved to the high-order eight bits of register PC. The content of register SP is incremented by 2.

1	1	0	0	1	0	0	1

Cycles: 3
States: 10
Addressing: reg. indirect
Flags: none

Rcondition (Conditional return)

If (CCC),

(PCL) ← ((SP))

(PCH) ← ((SP) + 1)

(SP) ← (SP) + 2

If the specified condition is true, the actions specified in the RET instruction (see above) are performed; otherwise, control continues sequentially.

1	1	C	C	C	0	0	0

Cycles: 1/3
States: 5/11
Addressing: reg. indirect
Flags: none

RST n (Restart)

((SP) – 1) ← (PCH)

((SP) – 2) ← (PCL)

(SP) ← (SP) – 2

(PC) ← 8 * (NNN)

The high-order eight bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order eight bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two. Control is transferred to the instruction whose address is eight times the content of NNN.

1	1	N	N	N	1	1	1

Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	N	N	N	0	0	0

Program Counter After Restart

PCHL (Jump H and L indirect – move H and L to PC)

(PCH) ← (H)

(PCL) ← (L)

The content of register H is moved to the high-order eight bits of register PC. The content of register L is moved to the low-order eight bits of register PC.

1	1	1	0	1	0	0	1

Cycles: 1
States: 5
Addressing: register
Flags: none

Stack, I/O, and Machine Control Group:

This group of instructions performs I/O, manipulates the Stack, and alters internal control flags.

Unless otherwise specified, condition flags are not affected by any instructions in this group.

FLAG WORD

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	Z	0	AC	0	P	1	CY

PUSH rp (Push)

((SP) − 1) ← (rh)
((SP) − 2) ← (rl)
(SP) ← (SP) − 2

The content of the high-order register of register pair rp is moved to the memory location whose address is one less than the content of register SP. The content of the low-order register of register pair rp is moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by 2. **Note: Register pair rp = SP may not be specified.**

1	1	R	P	0	1	0	1

Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none

PUSH PSW (Push processor status word)

((SP) − 1) ← (A)
$((SP)-2)_0$ ← (CY), $((SP)-2)_1$ ← 1
$((SP)-2)_2$ ← (P), $((SP)-2)_3$ ← 0
$((SP)-2)_4$ ← (AC), $((SP)-2)_5$ ← 0
$((SP)-2)_6$ ← (Z), $((SP)-2)_7$ ← (S)
(SP) ← (SP) − 2

The content of register A is moved to the memory location whose address is one less than register SP. The contents of the condition flags are assembled into a processor status word and the word is moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two.

1	1	1	1	0	1	0	1

Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none

POP rp (Pop)

(rl) ← ((SP))
(rh) ← ((SP) + 1)
(SP) ← (SP) + 2

The content of the memory location, whose address is specified by the content of register SP, is moved to the low-order register of register pair rp. The content of the memory location, whose address is one more than the content of register SP, is moved to the high-order register of register pair rp. The content of register SP is incremented by 2. **Note: Register pair rp = SP may not be specified.**

1	1	R	P	0	0	0	1

Cycles: 3
States: 10
Addressing: reg. indirect
Flags: none

POP PSW (Pop processor status word)

(CY) ← $((SP))_0$
(P) ← $((SP))_2$
(AC) ← $((SP))_4$
(Z) ← $((SP))_6$
(S) ← $((SP))_7$
(A) ← ((SP) + 1)
(SP) ← (SP) + 2

The content of the memory location whose address is specified by the content of register SP is used to restore the condition flags. The content of the memory location whose address is one more than the content of register SP is moved to register A. The content of register SP is incremented by 2.

1	1	1	1	0	0	0	1

Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

XTHL (Exchange stack top with H and L)

(L) ↔ ((SP))

(H) ↔ ((SP) + 1)

The content of the L register is exchanged with the content of the memory location whose address is specified by the content of register SP. The content of the H register is exchanged with the content of the memory location whose address is one more than the content of register SP.

1	1	1	0	0	0	1	1

Cycles: 5
States: 18
Addressing: reg. indirect
Flags: none

SPHL (Move HL to SP)

(SP) ← (H) (L)

The contents of registers H and L (16 bits) are moved to register SP.

1	1	1	1	1	0	0	1

Cycles: 1
States: 5
Addressing: register
Flags: none

IN port (Input)

(A) ← (data)

The data placed on the eight bit bi-directional data bus by the specified port is moved to register A.

1	1	0	1	1	0	1	1
port							

Cycles: 3
States: 10
Addressing: direct
Flags: none

OUT port (Output)

(data) ← (A)

The content of register A is placed on the eight bit bi-directional data bus for transmission to the specified port.

1	1	0	1	0	0	1	1
port							

Cycles: 3
States: 10
Addressing: direct
Flags: none

EI (Enable interrupts)

The interrupt system is enabled following the execution of the next instruction.

1	1	1	1	1	0	1	1

Cycles: 1
States: 4
Flags: none

DI (Disable interrupts)

The interrupt system is disabled immediately following the execution of the DI instruction.

1	1	1	1	0	0	1	1

Cycles: 1
States: 4
Flags: none

HLT (Halt)

The processor is stopped. The registers and flags are unaffected.

0	1	1	1	0	1	1	0

Cycles: 1
States: 7
Flags: none

NOP (No op)

No operation is performed. The registers and flags are unaffected.

0	0	0	0	0	0	0	0

Cycles: 1
States: 4
Flags: none

INSTRUCTION SET

Summary of Processor Instructions

Mnemonic	Description	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Clock[2] Cycles
		Instruction Code[1]								
MOV r_1, r_2	Move register to register	0	1	D	D	D	S	S	S	5
MOV M, r	Move register to memory	0	1	1	1	0	S	S	S	7
MOV r, M	Move memory to register	0	1	D	D	D	1	1	0	7
HLT	Halt	0	1	1	1	0	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
INR r	Increment register	0	0	D	D	D	1	0	0	5
DCR r	Decrement register	0	0	D	D	D	1	0	1	5
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	4
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
ANA r	And register with A	1	0	1	0	0	S	S	S	4
XRA r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA r	Or register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	10
JZ	Jump on zero	1	1	0	0	1	0	1	0	10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	10
JP	Jump on positive	1	1	1	1	0	0	1	0	10
JM	Jump on minus	1	1	1	1	1	0	1	0	10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	10
CALL	Call unconditional	1	1	0	0	1	1	0	1	17
CC	Call on carry	1	1	0	1	1	1	0	0	11/17
CNC	Call on no carry	1	1	0	1	0	1	0	0	11/17
CZ	Call on zero	1	1	0	0	1	1	0	0	11/17
CNZ	Call on no zero	1	1	0	0	0	1	0	0	11/17
CP	Call on positive	1	1	1	1	0	1	0	0	11/17
CM	Call on minus	1	1	1	1	1	1	0	0	11/17
CPE	Call on parity even	1	1	1	0	1	1	0	0	11/17
CPO	Call on parity odd	1	1	1	0	0	1	0	0	11/17
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	5/11
RNC	Return on no carry	1	1	0	1	0	0	0	0	5/11

Mnemonic	Description	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Clock[2] Cycles
		Instruction Code[1]								
RZ	Return on zero	1	1	0	0	1	0	0	0	5/11
RNZ	Return on no zero	1	1	0	0	0	0	0	0	5/11
RP	Return on positive	1	1	1	1	0	0	0	0	5/11
RM	Return on minus	1	1	1	1	1	0	0	0	5/11
RPE	Return on parity even	1	1	1	0	1	0	0	0	5/11
RPO	Return on parity odd	1	1	1	0	0	0	0	0	5/11
RST	Restart	1	1	A	A	A	1	1	1	11
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	11
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	11
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	11
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	11
POP B	Pop register pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4
XTHL	Exchange top of stack, H & L	1	1	1	0	0	0	1	1	18
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	5
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	5
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	5
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	5
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	5
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	5
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	5
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	5
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	5
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	5
CMA	Complement A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Complement carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable interrupt	1	1	1	1	0	0	1	1	4
NOP	No-operation	0	0	0	0	0	0	0	0	4

NOTES: 1. DDD or SSS – 000 B – 001 C – 010 D – 011 E – 100 H – 101 L – 110 Memory – 111 A.
2. Two possible cycle times, (5/11) indicate instruction cycles dependent on condition flags.

B. Pseudo-Ops. "Pseudo-op" is the name given to Assembly Language instructions that do not produce any machine code, but which direct the Assembler to perform its operations. The DOS Assembler provides op-codes for reserving storage space, defining the contents of memory locations and controlling the parameters of the Assembler's operation.

The following table is an alphabetical list of pseudo-ops along with their formats and functions. In these descriptions, e designates an address expression, and n designates a name. All other notation conventions are the same as in the rest of the DOS manual.

Table 4-A. DOS Assembler Pseudo-Ops

Instruction Format	Description
CMN[/<block name>/] <n1>, [<n2>, ...]	Common definition. The names n1, n2, . . . are declared to be in the Common block with the designated block name. If the block name is omitted, Blank Common is used. Each name is assumed to require one byte unless it is written in the form N(m) where m is an address expression that gives the length in bytes of the area assigned to the name N. If another CMN statement is encountered with the same block name, the first address assigned by the second statement directly follows the last address assigned by the first statement.
DATA <n1> [,<n2>],...	The names n1, n2, . . . are

defined to be in the Data area. Each name is assumed to require one byte unless it has the form

N(m)

where m is an address expression that gives the length in bytes of the area assigned to N.

DB <e1> [e2] [,...]
or
DB"<character string>"

Define Byte. The address expressions e1, e2, ... are evaluated and stored in successive bytes in memory. The character string form stores the ASCII codes of each character in successive bytes. The two forms may be mixed in a single statement. Character Constants are treated as character strings unless they are components of address expressions.

DC "<character string>"

Define Character. The characters in the string are stored one byte per character. The high-order bit of each byte is set to zero except for the last byte which has its high order bit set to 1. This arrangement allows quick searches for the end of the string.

DS <e>

The address expression e is evaluated and defines the number of bytes of space that are allocated. The contents of the space are not affected. All names used in e must be defined prior to the DS statement.

DW <e1>[,e2] [,...]	Define Word. The address expressions e1, e2, ... are evaluated and stored as 16 bit (two-byte) words. The addresses conform to the 8080 address convention that the low-order byte comes first and the high-order byte comes second. All addresses and address offsets are handled in this way, so the DW statement must be used to define addresses.
END <e>	END is the last statement of each program. The address expression e is the execution address of the program. Specifying e=0 (absolute) is equivalent to specifying no execution address.
ENDIF	Terminates the conditional assembly started by a previous IFF or IFT statement.
ENTRY <n1>[,n2] [,...]	Define Entry Points. The names n1, n2, ... are names of entry points in other programs and are defined as names in the program being assembled. The names must appear in an ENTRY statement before they appear as labels.
EQU <e>	Define Equivalence. The address expression e is evaluated and assigned to the label of the EQU statement. The label is required and may not have appeared previously as a label or in a DMN

	or DATA statement. All names used in e must have been defined previous to the EQU statement.
EXT <n1> [,n2] [,...]	The names n1, n2, ... are defined to be external references. They may not have been used as labels or in a CMN or DATA statement.
IFF <e>	Conditional Assembly - False. If the value of the address expression e is false, (=0 absolute), then all of the statements until the next ENDIF are assembled. If the value is true, the statements are not assembled. Conditional assemblies may not be nested.

4-4. Assembler Error Messages

Assembler error messages are printed in the leftmost column of the source code listing on the line in which the error occurred. The error codes are as follows:

Table 4-B. Assembler Error Messages

Code	Meaning
2	Second operand missing. An instruction that requires two operands was only given one.
A	Absolute required. Data, Common, External or Relative address was given where an Absolute value was required.
B	Block Name error. A Common or Data block name was invalid.
C	Too many Common blocks. Only 17 Common blocks are allowed.
D	Digit invalid. Valid digits are 0 - 9 in decimal, 0 - 7 in octal and 0 - 9 and A - F in hexadecimal.
E	Expression error. Error in the syntax, symbols or position of an address expression.
F	Operand field too long.
L	Label error.
M	Multiply defined name.

N	Name too long. Six characters is maximum.
O	Op-code invalid. An Op-code was encountered which is not in the list of op-codes recognized by the Assembler.
P	Phase error. Probably an error in the Assembler. Please report errors to the MITS, Inc., Software Department.
Q	Quoted string error. The ending quotation mark was missing from a character string.
T	Field or line terminated too soon.
U	Undefined name.
V	Value invalid. An address expression value was negative, too large or otherwise unusable.

ALTAIR DOS DOCUMENTATION
SECTION V
LINKING LOADER

5. THE LINKING LOADER

5-1. Introduction

The output file of the Assembler is a relocatable object code module. That is, it is a machine language program module (object code) that can be loaded by the appropriate loading program--anywhere in memory and executed (relocatable). Moreover, the Assembler allows the module produced by an assembly to refer symbolically to addresses in other modules as long as all of the modules that refer to each other are loaded into memory at the same time (see page 71, EXT pseudo-op).

The program that loads relocatable modules into memory and links their symbolic references to the proper addresses is called the Linking Loader (LINK). In the simplest case, where an entire program is contained in one module, LINK loads the program into memory and causes control to jump to its starting address.

In the more complex case, where several modules are to be loaded into memory and linked together to form a single large program, LINK serves many functions. It loads the modules and makes sure that bytes of a module are not destroyed by loading subsequent modules in overlapping locations. It makes the connections between all external references and the addresses to which they refer. It prints lists of those external references for which no addresses have been defined. It can even search the disks for files to resolve these undefined references and automatically load them. All of these functions are controlled by the Linking Loader's commands which are described in Table 5-A. For an explanation of the use of LINK in this case, see Appendix E.

If the system disk is mounted on drive zero, the Linking Loader is loaded and run by typing the following command to the Monitor:

```
.LINK
```

When LINK starts, it prints the following message:

```
DOS LINK VER x.x
*
```

The asterisk means LINK is ready to receive commands.

Table 5-A. Linking Loader Commands

L <file> <device> [<address at which to load relocatable module>]	Loads a module at the specified address. The module is loaded from the specified disk. The module must be in LINK's relocatable code format. If the loading address is not specified, the default address is 24000_8 for the first module to be loaded and the next available location above the previous module for all subsequent modules. The L command automatically adds a * to the file name. For an example of the use of the L command, see Appendix E, Section 2.
A	Displays the names in all of the currently loaded modules and their assigned addresses. Undefined names are displayed with asterisks instead of addresses.
U	Displays all undefined names in all current modules.
S <device>	For each undefined entry point name, LINK searches the specified device for a relocatable file by that name and loads it. For an example of the use of the S command, see Appendix E, Section 2.
E	Exits to the Monitor
X [execution address]	Begins execution of the program at execution address . If the execution address is omitted, X branches to the address in the

last encountered END statement. If no END statement has been encountered, X branches to location 24000_8.

5-2. Address Chaining

Each time LINK encounters a reference to a symbol that has not yet been defined, it enters the address of the reference into a chain. Each entry in the chain contains a pointer to the previous entry. The last entry contains zero absolute. When the symbol is defined, LINK goes through the chain again from the last entry to the first, replacing the contents of each entry with the assigned address of the symbol. As a result of this process, each reference to the symbol points to the correct address.

LINK handles external references by saving the unresolved chains from all of the modules. The contents of the first entry in a chain for one module is the address of the top of the chain for the previously loaded module.

The U command can be used to display the undefined symbols in all loaded modules.

5-3. Relocatable Object Code Module Format

The Assembler creates and LINK uses files which conform to the Relocatable Object Code Module format. Each module consists of records of 1024 bits each. A record is made up of a number of load items, each one of which is preceded by at least one control bit.

A. If the first bit is 0, the next eight bits are loaded as an absolute data byte. If the first bit is 1, the next two bits are input as a control field as follows:

B.

Control Bits	Action
01	The following 16 bits are loaded as a relocated address after adding the relocation base address.
10	The following 16 bits are to be loaded as a Data block reference address after adding the Data base.

11	The following 16 bits are to be loaded as a Common block reference address by adding the current Common base.
00	The next 9 bits are to be input as a control field and the following 16 bits as an address.

C. The 9-bit control field has the following format:

aannnxxxx

where aa designates the type of the address

aa	Type
00	Absolute
01	Relocated, relocation base is added before loading.
10	Data reference. Data base is added before loading.
11	Common reference, current Common base is added before loading.

nnn is the length, in bytes, of the program or common block name. When nnn = 0, the name is blank. If a name is specified, it immediately follows the address in the module.

xxxx is a 4 bit control field as follows:

xxxx	Action
1	Define Common Size. The address is interpreted as the size of the Common block that has the specified name. This type of item may be preceded only by Define Entry Name items. The program with the largest blank Common block must be loaded first. All programs which refer to named Common blocks must define them to be the same size.
2	Define Data Size. The address is interpreted as the size of the Data area. If this item is preceded only by Define Entry Name and Define Common Size items, normal memory allocation takes place.

If, however, Data block references occur before this item is loaded, the Data base is assigned to be the address of the first location from the top of memory, and all Data block reference addresses are subtracted from rather than added to the base.

3 Set Location Counter. The address is loaded into the loading location counter.

4 Address Chain. The current value of the loading location counter is placed in each element of the chain whose top element is the address.

5 Set Common Base. The assigned address of the named Common block is the current Common base.

6 Chain & Call an External Name. The name is placed into the loader table, if it is not already there. The address chain whose top element has the specified address is linked to the chain for the name if it has not yet been loaded or to the name (if it has been loaded).

7 Define Entry Point. The address is assigned to the named entry point.

8 Define Program Limit. The address is that of the first location after the program.

14 End of Record. This record indicates the end of the program being loaded and the end of data in this record. A is the execution address.

15 End of Module. End of load module. Control returns to the loader.

ALTAIR DOS DOCUMENTATION
SECTION VI
DEBUG

6. DEBUG PACKAGE

6-1. Introduction

The Debug package is a system program which provides facilities for debugging Assembly Language programs. Commands allow the following operations:

1. Display the contents of memory locations, registers or flags in several modes (octal, decimal, etc.)
2. Modify the contents of memory locations, registers or flags.
3. Insert, display and remove breakpoints to initiate pauses in program execution.
4. Start execution of the program at any address or at any breakpoint.

A. Running Debug. After the system disk is mounted in drive zero, Debug is entered from the Monitor by typing

.DEBUG

Debug indicates that it is loaded and running by printing

DOS DEBUG VER x.x

on the terminal. At this point, it is ready to receive commands. The Monitor may be reentered by typing R.

B. Addressing Modes. Debug can display, modify or transfer program control to any point in memory. In addition, entry to Debug causes the registers and condition flags to be stored in memory, making them available for display or modification.

Most of the Debug commands may be preceded by an address. This address may be expressed in any one of several modes.

1) Explicit. Anywhere an address is expected, a number is interpreted as an octal address. A number preceded by a pound sign (#) is interpreted as a decimal address. The address is entered into an address pointer in Debug. All commands operate on the location in the address pointer. The current contents of the address pointer may be accessed by typing a period (.). Thus,

the Debug command

./

displays the contents of the location whose address is currently in the address pointer. The use of the period is optional, in this case, since

./

and

/

cause the same operation to be performed.

2) Relative. An address may be specified in the following form:

<address> $\pm$ <offset>

For example:

100 + 10, the location whose address is 100_8 + 10_8 or . - 2 refers to the location whose address is that of the current location minus 2_8.

Two special cases of indirect addressing involve the <line feed> and <↑> commands.

<line feed> increments the address pointer and displays the contents of the resulting location.

<↑> (<^> on some terminals) decrements the address pointer and displays the contents of the resulting location.

In both cases, the increment in the symbolic I/O mode (see Section 2-1) is the length of the current instruction - 1, so that the next location displayed is that of the next instruction. In the W mode, the increment is 2 bytes and in all other modes the increment is one byte.

Typing an equal sign (=) after a relative address specification causes Debug to print the resultant address.

3) Indirect. Typing <tab> (Control/I) refers to the location whose address is the contents of the current

location. For example:

70/ JMP 5000 <tab>

5000/ SHLD 4750

Typing 70/ in the symbolic I/O mode W causes Debug to display the instruction at 70 which is a JMP to location 5000. Typing <tab>, which is equivalent to .<tab>, causes Debug to reference the instruction at location 5000. Subsequently, typing / causes the instruction at location 5000 to be displayed.

Typing <tab> when the current location is the low order byte of a two-byte address or the low order register of a register pair causes the address pointer to be loaded with the contents of both bytes of the address or the pair of registers.

4) Register. When Debug is entered, or when a breakpoint is encountered, Debug stores the contents of the registers and condition flags in memory in the following order:

Register	Remarks	
F	Condition Flags	
	Bit	Meaning
	0	Carry
	2	Even Parity
	4	Half Carry (for decimal arithmetic)
	6	Zero
	7	Sign (One means the MSB of result was 1)
A	Accumulator	
C	Note: The low order register of a pair is first)	
B		
E		
D		
L		

H	
S	Low order byte
S	High order byte

Once a register has been opened, typing <line feed> or <↑> causes the next or preceding register in the list to be accessed and displayed.

6-2. Display

Typing the following command:

<address>/

where the address is in any mode, causes Debug to display the contents of the specified location in the current I/O mode.

A. I/O Modes. Debug displays the contents of locations in several modes which may be specified by the programmer. The I/O mode is specified by typing dollar sign ($) or <ESCAPE> (<Altmode> on some terminals) followed by a letter.

Letter	I/O Mode
O	Octal
D	Decimal
W	Double byte octal. Displays contents of two successive locations. This is used primarily to display addresses.
A	ASCII. The characters displayed have ASCII codes equal to the contents of the location.
S	Symbolic. The instruction at the location is displayed in Assembly Language symbolic form. All bytes of the instruction are displayed, but address bytes are displayed in octal form.

If no I/O mode is specified, Debug proceeds as if the mode were specified as octal. Typing a semicolon (;) instead of / displays the contents of the current location in octal, regardless of the current I/O mode.

B. Displaying a Range of Locations. Typing the following command:

<address 1>, <address 2>T

displays the contents of all the locations from <address 1> to <address 2>, inclusive, in the current I/O mode.

6-3. Modify

The contents of a location may be modified by displaying the current contents of the location and then typing the new contents. For example

```
50/ XRA A  ORA A  <cr>
./ ORA A
```

The instruction ORA A replaces the original XRA A. All input after the display is used to modify the current location until the location is filled or until a delimiter is typed. The normal delimiter is <cr>. Other delimiters are as follows:

<line feed>	displays the next location
<↑>	displays the previous location
/ or ;	displays the modified contents of the current location
<tab>	displays contents of the location addressed by current location (typed as Control/I).
<ESCAPE>, +, @, !, =	are special and terminate input even though they have no specific function in this context

Input is interpreted according to the current I/O mode. If the input cannot be interpreted, "?" is printed on the terminal and the command must be repeated.

6-4. Breakpoints

Breakpoints provide the ability to pause in the execution of a program at any point and examine the contents of memory locations, registers and condition flags. A breakpoint is set by the X command, which has the following form:

<address> X

This command sets the next available breakpoint at the specified address. Eight breakpoints are available (numbered 0 - 7). When a breakpoint is encountered during execution of the program, the following message is printed on the terminal:

<number> BREAK@ <address>

Execution is suspended until it is restarted by a P or G command.

The positions of all the breakpoints in use can be displayed by the Q command:

Q<cr>

Example:

10X

20X

377X

Q

0 @ 10

1 @ 20

2 @ 377

Any (or all) breakpoints may be removed by the Y command:

Y

or

Y<number>

If no number is specified, all breakpoints are removed. If a number is specified, only that breakpoint is removed.

6-5. <u>Controlling Execution</u>

Debug may be used to control the execution of a program by means of the G and P commands.

A. The G Command. Execution can be started at any location by the G command:

<address>G

where the address is the location where execution is to start.

B. The P Command. Execution can be made to proceed from a breakpoint by means of the P command:

[<number>] P

If the number is typed, execution proceeds from the specified breakpoint. If the number is omitted, the most recently encountered breakpoint is specified. The P command cannot be used if no breakpoint has been encountered or if the breakpoint with the specified number has not been assigned.

C. Breakpoints and Execution Commands. When a G or P command is executed, Debug replaces the bytes at the breakpoint addresses with RST instructions. These instructions cause control to be transferred to locations 0, 7, 17, 27, 37, ... 77. At these locations, JMP instructions branch to a breakpoint handling routine in Debug. The bytes that were replaced are saved in a table and stored after the breakpoint is executed.

When a P command is executed, Debug reconstructs the instruction at the breakpoint by referring to the table and executes that instruction before branching to the instruction after the breakpoint. If the instruction at the breakpoint is itself a CALL, JMP or RST instruction, Debug branches to the proper location.

When a breakpoint RST is executed, the breakpoint routine saves all registers and condition flags and restores the original byte in the instruction string. In operation, the breakpoint processing procedure is transparent to the programmer and program execution is unaffected, except for the pauses initiated by the breakpoints.

6-6. Using Debug with Relocated Programs

The Assembler produces relocatable code modules that can be loaded in any place in memory by the Linking Loader. Thus, the addresses of program statements are not determined until the program is loaded. In order to use Debug on such programs, special functions are provided for handling base addresses.

Typing an apostrophe (') recalls the execution address returned by the Linking Loader for the current load module. Thus, the statement

```
'G
```

causes Debug to start execution of the module at the Linking Loader execution address.

The execution address may or may not be the first location in the program. For this reason, Debug also includes the capability of storing any address and recalling it for use in any Debug command. The statement

```
<address>%
```

stores the address and

```
&
```

recalls it for use. The address may be that of the first location in a module, common or data block, etc.

ALTAIR DOS DOCUMENTATION
SECTION VII
MISCELLANEOUS SYSTEM PROGRAMS

7. MISCELLANEOUS SYSTEM PROGRAMS

7-1. INIT

INIT is a system program that allows the initialization of the system (the number of disks, disk files, etc.) to be changed without reloading the system. INIT is run by typing

    .INIT

to the Monitor. INIT then prints the question

    MEMORY SIZE?

and the initialization dialog proceeds exactly as it does when the system is loaded (see Section 1-2c, p. 7).

7-2. CNS

CNS allows the console through which the user issues commands to be changed to another terminal. To use CNS, type

    .CNS <channel> <sense switch>

to the Monitor, where <channel> is the octal data channel number of the new console's I/O board, and <sense switch> is the new I/O board's octal sense switch setting. The data channel is the low order channel of the board and the sense switch settings are shown in Table 1-A on page 5.

For example, to switch to a terminal using a 2SIO board with 2 stop bits through channel 20, the following command is typed:

    .CNS 20 0

7-3. SYSENT

SYSENT is a system program file that contains addresses of several Monitor routines that are available for user program use. The following routines are available:

ABORT	exits to the Monitor and prints "PROGRAM ABORTING" on the terminal
EXIT	exits to the Monitor and prints "PROGRAM EXITING" on the terminal

ABORT and EXIT both return control from the program to the Monitor and close all files. The program name is found in location TASKNM (see below). ABORT is generally used to exit under error conditions while EXIT is used under normal exit conditions.

IO — allows access to the Monitor Call I/O routines. The following sequence is used in the calling program

```
CALL IO
DW  (address of Request Control
     Block)
```

See Appendix C for more information on Monitor Calls and Request Control Blocks.

Two special routines are used to print text messages.

TASKNM — contains the address of the memory area where ABORT and EXIT find the name of the calling program. The program name must be stored at this location before an ABORT or EXIT call is executed.

MSG — prints a user selected message on the terminal. The following sequence is used:

```
CALL   MSG
DW     (address of first byte
       of message)
```

MSG prints the message bytes until it prints a byte with the most significant bit set to one. Thus, the message should be stored with a DC pseudo-op.

To use the routine in SYSENT, the desired names must be defined as External names in the calling program. (See EXT statement, Table 4-A.) When the calling program is loaded into memory for execution, SYSENT must also be loaded. The following Linking Loader command is used for this purpose:

L SYSENT 0

This command loads SYSENT just above the user program.

7-4. LIST

LIST is a BASIC language routine that allows DOS Assembler listing files to be printed on a line printer. To use LIST, BASIC must be running and the DOS disk must be mounted. The following command runs LIST

RUN"LIST",<device number>

where the device number is that of the disk drive upon which the DOS disk is mounted.

LIST asks for the name of the program (the % sign is added automatically) and the device number of the disk on which the listing file resides. The listing is then printed on the system line printer.

ALTAIR DOS DOCUMENTATION APPENDICES

APPENDIX A. ASCII CHARACTER CODES

DECIMAL	CHAR.	DECIMAL	CHAR.	DECIMAL	CHAR.
ØØØ	NUL	Ø43	+	Ø86	V
ØØ1	SOH	Ø44	,	Ø87	W
ØØ2	STX	Ø45	-	Ø88	X
ØØ3	ETX	Ø46	.	Ø89	Y
ØØ4	EOT	Ø47	/	Ø9Ø	Z
ØØ5	ENQ	Ø48	Ø	Ø91	[
ØØ6	ACK	Ø49	1	Ø92	\
ØØ7	BEL	Ø5Ø	2	Ø93	]
ØØ8	BS	Ø51	3	Ø94	^
ØØ9	HT	Ø52	4	Ø95	<
Ø1Ø	LF	Ø53	5	Ø96	'
Ø11	VT	Ø54	6	Ø97	a
Ø12	FF	Ø55	7	Ø98	b
Ø13	CR	Ø56	8	Ø99	c
Ø14	SO	Ø57	9	1ØØ	d
Ø15	SI	Ø58	:	1Ø1	e
Ø16	DLE	Ø59	;	1Ø2	f
Ø17	DC1	Ø6Ø	<	1Ø3	g
Ø18	DC2	Ø61	=	1Ø4	h
Ø19	DC3	Ø62	>	1Ø5	i
Ø2Ø	DC4	Ø63	?	1Ø6	j
Ø21	NAK	Ø64	@	1Ø7	k
Ø22	SYN	Ø65	A	1Ø8	l
Ø23	ETB	Ø66	B	1Ø9	m
Ø24	CAN	Ø67	C	11Ø	n
Ø25	EM	Ø68	D	111	o
Ø26	SUB	Ø69	E	112	p
Ø27	ESCAPE	Ø7Ø	F	113	q
Ø28	FS	Ø71	G	114	r
Ø29	GS	Ø72	H	115	s
Ø3Ø	RS	Ø73	I	116	t
Ø31	US	Ø74	J	117	u
Ø32	SPACE	Ø75	K	118	v

DECIMAL	CHAR.	DECIMAL	CHAR.	DECIMAL	CHAR.
033	!	076	L	119	w
034	"	077	M	120	x
035	#	078	N	121	y
036	S	079	O	122	z
037	%	080	P	123	{
038	&	081	Q	124	|
039	'	082	R	125	
040	(	083	S	126	
041	)	084	T	127	DEL
042	*	085	U		

LF=Line Feed   FF=Form Feed   CR=Carriage Return   DEL=Rubout

APPENDIX B

DISK INFORMATION

1. FORMAT OF THE ALTAIR FLOPPY DISK

1-1. Track Allocation

Track	Use
0 - 5	DOS Memory Image
6 - 69	Space for either Random or Sequential files
70	Directory Track
71 - 76	Space for Sequential files only

1-2. Sector Format

There are 32 sectors per track and 137 bytes per sector. Of these bytes, 128 are available for data storage.

Tracks 0 - 5

Byte	Use
0	Track number + 128 decimal
1 - 2	Sixteen bit address of the next higher location in memory than the highest location saved on this ~~sector~~
3-130	128 bytes of DOS code
131	Stop byte (255 decimal)
132	Checksum. Sum of the bytes 3 - 130 with no carry out of one byte

Tracks 6 - 76

Byte	Use
0	Most significant bit always on. Contains track number plus 200 octal.
1	(Sector number)*17 MOD 32
2	File number from directory. Zero means this sector is not part of any file. If the sector is the first of a group of 8 sectors, 0 means the whole group is free.
3	Number of data bytes written (0 to 128). This is always 128 for random file data blocks. For random file index blocks, this number is the number of groups allocated for this file.
4	Checksum. Sum of bytes 3 - 134 with no carry out of one byte.

Byte	Use
5, 6	Pointer to the next group of the file. The first byte is the track number and the second byte is the sector number. Zero indicates the end of the file.
7 - 134	Data
135	Stop byte (255 decimal)
136	Unused

1-3. The Directory Track

THIS DESCRIBES DATA SECTION

The Directory takes all of track 70. Each sector has 8 file name records, each 16 bytes long. The format of the sector is as follows:

Byte	Use
0 - 7	File name
8, 9	Pointer to the start of the file (track, sector).
10	File mode. 2=sequential, 4=random
11 - 15	Unused

If the first byte of the file name is 0, the file has been deleted. If the first byte is 255 decimal, the file is the last in the directory and all file name records after it are ignored.

2. RANDOM FILES

2-1. Format of Random Files

A random file may contain any number of sectors. The first two sectors are the "index blocks." The "Number of Data Bytes" field in the first block indicates the number of groups currently allocated to this file. The next 256 bytes in the two blocks give the designations of the data sectors in the file in the order they occupy in the file. The upper two bits in the byte give the group number and the lower 6 bits give the track number - 6.

2-2. Using Random Files

The user must allocate a 128 byte buffer for each random file to be open at one time in the program. A Random Read or Write transfers an entire 128 byte block at a time into or out of the buffer assigned to that file.

The format of the data in the buffer is defined by the user.

APPENDIX C. MONITOR CALLS

Since the Monitor contains all the I/O routines for all of the peripheral devices in the system, it is not necessary for the programmer to write I/O routines for each program. Instead, the program can call the Monitor to handle all input and output.

For this reason, DOS I/O is device-independent. The programmer need not consider the idiosyncracies of individual I/O devices when a program is being written, and the I/O device can be chosen at the time the program is executed.

The instruction sequence for calling the Monitor from an Assembly language routine is as follows:

```
CALL IO                              ;IO IS DEFINED IN SYSENT
DW (Request Control Block address) ;A SYSTEM PROGRAM FILE (SEE
                                       SECTION 7-3).
```

The Request Control Block (RCB) is a block of data which provides the information the Monitor needs to perform the requested operation.

The first two bytes in every Request Control Block have the same significance. The first byte is always the operation code byte which tells the Monitor the action being requested. The second byte is a status byte which is set to zero if the operation is completed successfully and to a non-zero value if an error occurred. The error codes are in Appendix

In the list that follows, the Request Control Blocks for each I/O Monitor call are given, beginning with the third byte. When an RCB is constructed, DB statements can be used to define the byte quantities and DW to define the two-byte quantities. This is because the two-byte quantities are interpreted as addresses and must conform to the 8080's format for addresses (first byte is the low order byte).

I/O MONITOR CALLS

Operation	Code	Description
Open	104	Prepares a file for input or output. Assigns a file number to the file. A file must be opened before information can be transferred to or from it. The next Read or Get operation after Open begins with the first byte in the file.

Byte	Function
3	File number. The file is referred to by this number until it is closed.
4	File type. The bits of the file type byte have the following significance: 0 - sequential input 1 - sequential output 2 - random. Open for input and output simultaneously. 7 - explicit device specification. If bit 7 is on, transfer takes place through the device specified in bytes 5 and 6. Otherwise, bytes 5 and 6 are ignored and transfer takes place through the last device used for this file. Note: Bit 0 is the least significant bit. Only one bit may be on at one time.
5	Kind of Device 0 - Teletype 1 - cassette tape 6 - floppy disk
6	Device number
7, 8	Address of file name area

Close	105	Ends the connection between a file number and a file. Normal exit from a system program or jumping to location zero causes all files to be closed.

	Byte	Function
	3	File number

Read	102	Reads a number of bytes from a sequential file - either on disk or on another I/O device

	Byte	Function
	3	File number
	4	Mode. The bits of the mode byte have the following significance: Bit 1 on - Echo. Prints all characters as they are entered. Bit 1 off - no echo. Bit 2 on - ASCII. Control/R Control/U and Rubout recognized, input terminates on <cr>. Bit 2 off - Absolute binary code. Note: Bit 0 is the least significant bit.
	5, 6	Address of input buffer.
	7, 8	Number of bytes to be transferred (two-byte quantity interpreted as an address)
	9, 10	Number of bytes actually transferred (interpreted as an address). This operation begins by reading the next byte after the last byte to be read and reads the specified number of bytes.

Write	103	Writes a number of bytes into a file on a disk or another I/O device. The bytes are written after the last byte in the file.
	Byte	Function
	3	File number
	4	Mode. The bits of the mode byte have the following significance: Bit 2 on - ASCII. Adds nulls to the end of the line, expands tabs. Bit 2 off - Absolute. Note: Bit zero is the least significant bit.
	5, 6	Address of write buffer
	7, 8	Number of characters to be written (interpreted as an address)
	9, 10	Number of bytes actually transferred (interpreted as an address)
Random Read	4	Reads a 128-byte record from a random file on disk. The record is read into a 128 byte buffer in memory which must have been previously allocated. An error results if a Random Read is performed on a sequential file.
	Byte	Function
	3	File number
	4, 5	Address of memory buffer
	6, 7	Record number (interpreted as an address)
Random Write	5	Writes a 128 byte record into a random file. The record is written from a 128 byte memory buffer. An error results if a Random Write is performed to a sequential file.

	Byte	Function
	3	File number
	4, 5	Address of memory buffer
	6, 7	Record number (interpreted as an address)
Get Character	2	Reads the next character (1 byte) from an input file. If the file is on disk, it must be opened for input. The first Get after Open reads the first character in the file.
	Byte	Function
	3	File number
	4	Byte reserved for the character to be read
Put Character	3	Writes a character (1 byte) on an output file. The character is added to the end of the file. If it is a disk file, the file must be opened for output first.
	Byte	Function
	3	File number
	4	Character to be written
Block Input	107	Reads a sector (128 bytes) from a disk file* into a buffer in memory. Returns the address of the first data byte in the buffer and a pointer to the number of bytes in the block.
	Byte	Function
	3	File number
	4, 5	Pointer to number of bytes in the block
	6, 7	Pointer to first available data byte

*Block Input may be used to input data from a terminal. In that case, only 1 byte is transferred into the buffer. Use of Block Input in this way may save programming effort, but Get Character is much faster and more efficient.

Block Output	110	Writes a sector (128 bytes) to a disk file*. Returns the addresses of the first byte of the next 128-byte buffer to be written and the number of empty bytes in the buffer. To write a block of data, the Block Output routine is called to get pointers to the memory buffer. The buffer is then filled with data to be output and the Block Output routine is called again to write the data. Each successive Block Output call returns pointers to be used by the next Block Output call.

Byte	Function
3	File number
4, 5	Pointer to the number of bytes left empty in the buffer. When this number is zero, the buffer is full.
6, 7	Address of the first byte in the buffer.

*Block Output may be used to output data to a terminal. In that case, each Block Output call outputs one byte.

These Monitor calls are used in the following manner: The Input or Output routine is called to get the pointers to the buffer. In the Input case, the buffer is filled with input data. In the Output case, the program must fill the buffer with data to be output. As each byte is transferred either to or from the buffer, the byte counter (pointed to by bytes 4 and 5) is decremented. When the counter reaches zero, the transfer to or from the buffer is complete. Calling Block Output again writes the buffer onto the specified disk file and returns new pointers. Calling Block Input again reads another sector of data and returns new pointers.

In addition to these I/O Monitor Calls, Monitor Calls are available which perform the operations of the Monitor commands. These calls allow files to be opened, saved and deleted; disks to be mounted and dismounted, etc. without having to return control to the Monitor. The first two bytes of each of the command Monitor Calls are the same as the I/O Monitor Calls except for the codes. The listings below show the rest of the bytes of the Request Control Blocks.

Operation	Code	Description
Initialize	45	Same as DIN command
	Byte	Function
	3	Kind of device (disks are the only devices currently supported). Byte = 6.
	Byte	Function
	4	Device number
Rename	44	Same as REN command
	Byte	Function
	3	Kind of device = 6 for disk
	4	Device number
	5, 6	Address of 8-byte old name field
	7, 8	Address of 8-byte new name field
Delete	43	Same as DEL command
	Byte	Function
	3	Kind of device = 6 for disk
	4	Device number
	5, 6	Address of 8 byte file name
Directory	42	Same as DIR command
	Byte	Function
	3	Kind of device = 6 for disk.
	4	Device number
	5, 6	File number where the output of the directory is to be written. The file must be open for output.
Dismount	41	Same as DSM command.

	<u>Byte</u>	<u>Function</u>
	3	Kind of device = 6 for disk
	4	Device number
Mount	40	Same as MNT command.
	<u>Byte</u>	<u>Function</u>
	3	Kind of device = 6 for disk
	4	Device number
Save	106	Same as SAV command.
	<u>Byte</u>	<u>Function</u>
	3	Kind of device
		6 for disk
		0 for Teletype
	4	Device number
	5, 6	Address of 8 byte file name
Load	100	Same as LOA command
	<u>Byte</u>	<u>Function</u>
	3	Kind of device
		0 for Teletype
		1 for cassette tape
		6 for floppy disk
	4	device number
	5, 6	address of 8 byte file number
	7, 8	address of first byte to be saved
	9, 10	address of last byte to be saved
	11, 12	starting address

APPENDIX D. ABSOLUTE LOAD TAPE FORMAT

The paper tape dump of an object program consists of 3 records. The Begin/Name record is first, and carries the name of the program and comments (version number, date, etc.) The program records follow the Begin/Name record. The last record is an end-of-file record. The formats of the records are as follows:

A. Begin/Name Record

Byte 1	125Q	Begin record sync byte
2-4	Name	Program name
5-N	15Q	Terminates the Begin/Name record

B. Program Record

Byte 1	74Q	Program record sync byte
2	[illegible]	Number of bytes in this record
3, 4	Load Address	Low order byte is first
5-N	Program Data	
N+5	Checksum	All bytes except the first two are added with no carry to generate a checksum byte used to detect load errors.

C. End-of-File Record

Byte 1	170Q	EOF Record sync byte
2, 3	Begin Execution Address	

APPENDIX E. THE FILE COPY UTILITY

1. As an example of the use of the various facilities of DOS to solve a specific problem, the listing of a file copying routine is given in this appendix.

   This program copies a file from one file and device to another. Any file on any device in the system may be copied to any other device with this program.

   The program is highly structured, with a central routine (COP) that calls a number of other routines to perform specific actions.

   To copy a file, run the copy program by typing the following command to the Monitor:

   <u>.</u>COP

   The program is stored on disk as an absolute binary file so it is loaded and run immediately. When the program starts, it prints the following messages:

   <u>COPY FILE</u>

   <u>SET UP INPUT</u>

   It then asks for the type of device from which the file is to be copied. The user answers with "FDS" for a disk or "TTY" for the terminal. At this point, the copy program asks the device number (0, if there is only one device of that type) and the name of the file to be copied. If the device is "TTY", no file name need be specified. After the input parameters have been entered, the program prints

   <u>SET UP OUTPUT</u>

   and asks the device type, number and file name for output. If the output device is "TTY", no output file name need be specified.

   When the copy action is complete, the program exits.

   This Appendix lists the main routine COP and some of the more important or instructive subroutines. For a complete listing of the routines, use COP to copy them to the terminal. To do this, specify the output device as TTY and copy the following routines.

&DN	&TABLE	&ASK
&DTYP	&COP	&SYSENT
&LDEM	&CMPB	
&MOVB	&AANS	

2. To run the copy program from the Assembly Language source files on the system disk, it is first necessary to assemble all of the files in the list above. To do this, type the following command:

    .ASM COP 0

when the file is assembled, ASM prints

    000000 ERRORS DETECTED
    ANY MORE ASSEMBLIES?

The programmer replies to this question with the name of the next program to be assembled. This process continues until all of the programs in the list have been assembled. To load these modules into memory and link them together into the copy program, the Linking Loader is run with the following command:

    *LINK

When LINK prints its prompt asterisk, the main copy program module COP can be run with the following command:

    *L COP 0

At this point, LINK loads the module into memory and resolves the references to all symbolic addresses. Since numerous other symbols are as yet undefined, DOS prints a list of these symbols as follows:

```
     TSKNM     *  MSG      *  DTYP     *  DN       *  ASK
    *  MOV8    *  IO       *  EXIT     *  BDEX        *
     ABORT     *  GDEX        *
```

The asterisks after each file number indicate that the names are undefined. These names are all those of entry points in the modules that have not been loaded.

To load some of the required modules, the following command may be typed:

    *S 0

The S command adds asterisks to the undefined names and searches the specified disk for files with the resulting names. When LINK finds such a file, it loads and links it. Finally, LINK prints a list of those entry names that are still undefined:

```
     TSKNM     *     MSG      *     MOV8     *     IO
*    EXIT      *     ABORT       *
```

Entry point MOV8 is contained in file MOVB, so that it can be defined by the following command:

    *L MOVB 0

The remaining entry names are in file SYSENT which is loaded with the following command:

    *L SYSENT 0

Now that all of the required modules are loaded and linked together, the entire program is ready to be executed with the following command:

    *X

The copy program starts up and prints its prompt questions as above.

COP LISTING

The following statements define the entry point and external references.

```
        ENTRY   COP
        EXT     EXIT,ABORT
        EXT     TASKNM,MSG
        EXT     MOV8,IO
        EXT     DTYP,DN,ASK
        EXT     GDEX,BDEX
;
;IDENTIFY PROGRAM AND SET RADIX
;
COP     LXI     H,COPID ;GET PRGID
        SHLD    TASKNM  ;PUT AWAY
        CALL    MSG     ;DISPLAY IT
        DW      COPID
```

The setup routines are basically a series of Monitor Calls. They ask the operator for the file name and disk number, open the required files and check to make sure everything is operating properly.

```
;
;SET UP INPUT FILE
;
        CALL    MSG     ;TEL OPR WHATS GOING ON
        DW      SETUIN
        CALL    DTYP    ;INPUT DEVICE TYPE
        STA     DTIN
        CALL    DN      ;DEVICE NUMBER
        STA     DNIN
        CALL    ASK     ;FILE NAME
        DW      ASFNM
        LXI     D,FNIN  ;PUT IT AWAY
```

```
                CALL    MOV8
                CALL    IO          ;OPEN FILE
                DW      RBINOP
                LDA     STINOP      ;CHECK STATUS
                ORA     A
                JNZ     NOINOP      ;UNABLE TO OPEN
                LDA     DTIN        ;IS INPUT DEVICE A DISK
                CPI     6
                JNZ     CHRIN       ;NO - DO INPUT BY CHARACTERS
                LXI     H,BLKGC     ;SET UP GC FOR
        ;¬                            BLOCK INPUT ROUTINE
                SHLD    GCROUT
                CALL    IO          ;SET UP BLOCK GET POINTERS
                DW      BLGCRB
                JMP     SETO        ;GO SET UP OUTPUT
        CHRIN   LXI     H,CHRGC     ;USE CHRGC ROUTINE
                SHLD    GCROUT
        ;
        ;SET UP OUTPUT FILE
        ;
        SETO    CALL    MSG         ;TELL OPR WHATS GOING ON
                DW      SETUOU
                CALL    DTYP        ;DEVICE TYPE
                STA     DTOU
                CALL    DN          ;DEVICE NUMBER
                STA     DNOU
                CALL    ASK         ;FILE NAME
                DW      ASFNM
                LXI     D,FNOU      ;PUT IT AWAY
                CALL    MOV8
                CALL    IO          ;OPEN FILE
                DW      RBOUOP
                LDA     STOUOP      ;CHECK STATUS
                ORA     A
                JNZ     NOOUOP      ;UNABLE TO OPEN
                LDA     DTOU        ;IS OUTPUT DEVICE DISK
                CPI     6
                JNZ     CHROU       ;NO DO OUTPUT BY CHAR
                LXI     H,BLKPC     ;SET UP PC FOR
        ;                            BLOCK PUT ROUTINE
                SHLD    PCROUT
                CALL    IO          ;SET UP BLOCK PUT POINTERS
                DW      BLPCRB
                JMP     MINIT       ;GO DO MISC INIT
        CHROU   LXI     H,CHRPC     ;SET UP OUTPUT BY CHAR
                SHLD    PCROUT
        ;
        ;MISC INIT
        ;
        MINIT   CALL    ILD         ;INPUT LEADER
                CALL    OLD         ;OUTPUT LEADER
```

The copy loops call the get character and put character routines to copy binary bytes or ASCII coded characters.

```
;
;MAIN COPY LOOPS
;
        LDA     FNIN     ;GET FILE TYPE
        CPI     "&"      ;EDIT SOURCE?
        JZ      ASCCOP   ;YES - IS ASCII FILE
        CPI     "$"      ;EDIT BACKUP FILE?
        JZ      ASCCOP   ;YES - IS ASCII FILE
        CPI     "%"      ;LISTING FILE?
        JZ      ASCCOP   ;YES - IS ASCII FILE
;                        ;NO - MUST BE BINARY
;
;BINARY COPY LOOP
;
BINCL1  MVI     B,15     ;SET COUNTER
BINCLP  CALL    GC       ;GET CHARACTER
        DW      BINEOF   ;EOF ROUTINE
        CALL    PC       ;PUT BINARY BYTE
        CPI     0377     ;RUBOUT?
        JNZ     BINCL1   ;NO - RESET COUNTER & LOOP
        DCR     B        ;ONE LESS RUBOUT TO GO
        JZ      EXIT     ;ALL DONE
        JMP     BINCLP   ;LOOP
BINEOF  MVI     B,15     ;ADD RUBOUT EOF MARKER
        MVI     A,0377   ;RUBOUT
BINEO1  CALL    PC       ;OUTPUT RUBOUT
        DCR     B        ;ONE LESS TO GO
        JNZ     BINEO1   ;LOOP IF NOT DONE
        JMP     EXIT     ;ALL DONE
;
;ASCII COPY
;
ASCCOP  LDA     DTOU     ;CHECK DEVICE TYPE
        CPI     6        ;IS IT FDS
        JNZ     ASCCL2   ;NO - MUST EXPAND CTL I,ETC.
ASCCL1  CALL    GC       ;GET CHARACTER
        DW      ASCEOF   ;EOF ROUTINE
        CALL    PC       ;OUTPUT ASC CHAR TO DISK,
;                         NO TAB EXPAND
        CPI     032      ;IS CHAR CTL Z
        JZ      EXIT     ;YES - ALL DONE
        JMP     ASCCL1   ;NO LOOP
ASCEOF  MVI     A,032    ;ADD CTL Z TO FILE
        CALL    PC       ;OUTPUT IT
        JMP     EXIT     ;ALL DONE
ASCCL2  CALL    GC       ;GET CHARACTER
        DW      ASCEOF   ;EOF ROUTINE
        STA     DAPC2    ;PUT CHAR AWAY
        CALL    IO       ;OUTPUT IT
```

```
                DW      RBPC2
                CPI     032         ;IS CHAR CTL Z?
                JZ      EXIT        ;YES - ALL DONE
                JMP     ASCCL2      ;NO LOOP
```

Get character uses block input Monitor Calls to read data from the input file. The routine checks for input errors and end-of-file marks.

```
;
;GET CHARACTER ROUTINES
;
GC              PUSH    H           ;SAVE [H,L]
                LHLD    GCROUT      ;GET ADDRESS OF ROUTINE TO USE
                PCHL                ;JUMP TO IT
GCNWBL          CALL    IO          ;SET UP POINTERS FOR NEW BLOCK
                DW      BLGCRB
                LDA     BLGCST      ;CHECK STATUS
                CPI     025         ;IS IT EOF
                POP     H           ;RESTORE [H,L]
                JZ      BDEX        ;TAKE EOF EXIT
                PUSH    H           ;SAVE [H,L]
                ORA     A           ;ANY ERRORS
                JNZ     ABORT       ;YES - BAIL OUT
BLKGC           LHLD    BLGCCP      ;GET POINTER TO
;                                    NUMBER OF BYTES LEFT
                MOV     A,M         ;GET NBR BYTES LEFT
                ORA     A
                JZ      GCNWBL      ;IS ZERO MUST GET ANOTHER BLOCK
                DCR     M           ;ONE LESS
                LHLD    BLGCDP      ;GET POINTER TO DATA
                MOV     A,M         ;GET DATA
                INX     H           ;ADVANCE POINTER
                SHLD    BLGCDP      ;PUT POINTER AWAY
                POP     H           ;RESTORE [H,L]
                JMP     GDEX        ;TAKE NORMAL EXIT
CHRGC           POP     H           ;RESTORE [H,L]
                CALL    IO          ;GET CHARACTER
                DW      RBGC        ;CHECK STATUS
                LDA     STGC
                CPI     025         ;EOF?
                JZ      BDEX        ;YES
                ORA     A           ;ERROR STATUS
                JNZ     ABORT       ;YES - BAIL OUT
                LDA     DAGC
                JMP     GDEX
```

Put character uses block output Monitor Calls to write data into the output file.

```
;
;PUT CHARACTER ROUTINES
;
PC              PUSH    H           ;SAVE [H,L]
                LHLD    PCROUT      ;GET ADDRESS OF ROUTINE TO USE
```

```
        PCHL             ;JUMP TO IT
BLKPC   PUSH    PSW      ;SAVE DATA
        LHLD    BLPCCP   ;POINTER TO NUMBER
;                         OF BYTES LEFT IN BUFFE
        MOV     A,M      ;GET NUMBER OF BYTES LEFT
        ORA     A        ;IS IT ZERO?
        JNZ     BLKPCS   ;NO STUFF BYTE
        CALL    IO       ;SET UP POINTERS FOR NEW BLOCK
        DW      BLPCRB
        LDA     BLPCST   ;CHECK STATUS
        ORA     A
        JNZ     ABORT    ;NO GOOD - BAIL OUT
BLKPCS  DCR     M        ;ONE LESS BYTE
        LHLD    BLPCDP   ;GET POINTER TO DATA
        POP     PSW      ;RESTORE DATA
        MOV     M,A      ;PUT DATA IN BUFFER
        INX     H        ;ADVANCE POINTER
        SHLD    BLPCDP   ;PUT POINTER AWAY
        POP     H        ;RESTORE [H,L]
        RET              ;ALL DONE
CHRPC   POP     H        ;RESTORE [H,L]
        PUSH    PSW      ;SAVE CHARACTER
        STA     DAPC     ;STORE CHARACTER
        CALL    IO       ;OUTPUT IT
        DW      RBPC
        LDA     STPC     ;CHECK STATUS
        JNZ     ABORT
        POP     PSW      ;RESTORE CHARACTER
        RET              ;ALL DONE
;
;TAKE CARE OF LEADER
;
ILD     RET              ;***
OLD     RET              ;***
;
;ERROR BAILOUTS
;
NOINOP  CALL    MSG
        DW      MSNOIN
        JMP     ABORT
NOOUOP  CALL    MSG
        DW      MSNOOU
        JMP     ABORT
MSNOIN  DB      015
        DB      012
        DC      "INPUT FILE OPEN ERROR"
MSNOOU  DB      015
        DB      012
        DC      "OUTPUT FILE OPEN ERROR"
```

The following Request Control Blocks correspond to COP's Monitor Calls.

```
021600 ;
021700 ;OPEN INPUT FILE REQUEST BLOCK
021800 ;
021900 ;OPEN W/ ERROR MSG SUPPRESSION
022000 RBINOP  DB     0104+0200
022100 STINOP  DS     1          ;STATUS
022200         DB     1          ;FIL NBR
022300         DB     1+0200     ;SEQ IN,EXP DEV
022400 DTIN    DS     1          ;DEV TYPE
022500 DNIN    DS     1          ;DEV NBR
022600         DW     FNIN       ;PTR TO FILE NAME
022700 FNIN    DS     8          ;FILE NAME
022800 ;
022900 ;OPEN OUTPUT FILE REQUEST BLOCK
023000 ;
023100 ;OPEN W/ ERROR MSG SUPPRESSION
023200 RBOUOP  DB     0104+0200
023300 STOUOP  DS     1          ;STATUS
023400         DB     2          ;FILE NBR
023500         DB     2+0200     ;SEQ OUT,EXP DEV
023600 DTOU    DS     1          ;DEVICE TYPE
023700 DNOU    DS     1          ;DEV NUMBER
023800         DW     FNOU       ;PTR TO FILE NAME
023900 FNOU    DS     8          ;FILE NAME
024000 ;
024100 ;CHARACTER GET REQUEST BLOCK
024200 ;
024300 RBGC    DB     2          ;CHRGET
024400 STGC    DS     1          ;STATUS
024500         DB     1          ;FILE NBR
024600 DAGC    DS     1          ;DATA
024700 ;
024800 ;CHARACTER PUT REQUEST BLOCK
024900 ;
025000 RBPC    DB     3          ;CHRPUT
025100 STPC    DS     1          ;STATUS
025200         DB     2          ;FILE NBR
025300 DAPC    DS     1          ;DATA
025400 ;
025500 ;REQUEST BLOCK TO SET UP CHRGET POINTERS INTO D
025600 ;
025700 BLGCRB  DB     0107       ;SET UP BLK GET POINTERS
025800 BLGCST  DS     1          ;STATUS BYTE
025900         DB     1          ;INPUT FILE NUMBER
026000 BLGCCP  DS     2          ;POINTER TO NUMBER
026010 ;                          LEFT IN BLOCK
026100 BLGCDP  DS     2          ;POINTER TO DATA
026200         DS     2          ;RESERVED FOR MONITOR
026300 ;
026400 ;REQUEST BLOCK TO SET UP CHRPUT POINTERS INTO D
```

```
;
BLPCRB  DB      0110    ;SET UP BLK PUT POINTERS
BLPCST  DS      1       ;STATUS BYTE
        DB      2       ;OUTPUT FILE NBR
BLPCCP  DS      2       ;POINTER TO SPACE
;                        LEFT IN BLOCK
BLPCDP  DS      2       ;POINTER TO DATA
        DS      2       ;RESERVED FOR MONITOR
;
;CHAR PUT W/ TAB EXPANSION
;
RBPC2   DB      0103    ;WRITE
        DS      1       ;STATUS
        DB      2       ;OUTPUT FILE NUMBER
        DB      0       ;ASCII
        DW      DAPC2   ;PTR TO BUFFER
        DW      1       ;SIZE OF BUFFER
        DS      2       ;NUMBER TRANSFERED
DAPC2   DS      1       ;DATA
;
;MISC
;
GCROUT  DS      2       ;ADDRESS OF GC ROUTINE TO USE
PCROUT  DS      2       ;ADDRESS OF PC ROUTINE TO USE
COPID   DB      015     ;CR
        DB      012     ;LF
        DC      "COPY FILE"
```

The following are messages for the dialog with the operator.

```
ASFNM   DB      015
        DB      012
        DC      "ENTER FILE NAME "


SETUIN  DB      015
        DB      012
        DC      "SET UP INPUT"


SETUOU  DB      015
        DB      012
        DC      "SET UP OUTPUT"


        END     COP
```

APPENDIX F. BOOTSTRAP LOADERS

2SIO

Load Sense Switches — 2 stop bits - none up
1 stop bit - A8 up

Bootstrap Loader

Octal Address	Octal Data
ØØØ	Ø76
ØØ1	ØØ3
ØØ2	323
ØØ3	Ø2Ø
ØØ4	Ø76
ØØ5	ØXX (XX = 21 for 2 stop bits, 25 for 1 stop bit)
ØØ6	323
ØØ7	Ø2Ø
Ø1Ø	Ø41
Ø11	3Ø2
Ø12	Ø77
Ø13	Ø61
Ø14	Ø32
Ø15	ØØØ
Ø16	333
Ø17	Ø2Ø
Ø2Ø	Ø17
Ø21	32Ø
Ø22	333
Ø23	Ø21
Ø24	275
Ø25	31Ø
Ø26	Ø55
Ø27	167
Ø3Ø	3ØØ
Ø31	351
Ø32	Ø13
Ø33	ØØØ

PIO

Load Sense Switches A10, A8 - up

Bootstrap Loader

Octal Address	Octal Code
ØØØ	Ø41
ØØ1	3Ø2
ØØ2	Ø77
ØØ3	Ø61
ØØ4	Ø23
ØØ5	ØØØ
ØØ6	333
ØØ7	ØØ4
Ø1Ø	346
Ø11	ØØ1
Ø12	31Ø
Ø13	333
Ø14	ØØ5
Ø15	275
Ø16	31Ø
Ø17	Ø55
Ø2Ø	167
Ø21	3ØØ
Ø22	351
Ø23	ØØ3
Ø24	ØØØ

SIO

Load Sense Switches A9 - up

Bootstrap Loader

Octal Address	Octal Data
ØØØ	Ø41
ØØ1	3Ø2
ØØ2	Ø77
ØØ3	Ø61
ØØ4	Ø22
ØØ5	ØØØ
ØØ6	333
ØØ7	ØØØ
Ø1Ø	Ø17
Ø11	33Ø
Ø12	333
Ø31	ØØ1
Ø14	275
Ø15	31Ø
Ø16	Ø55
Ø17	167
Ø2Ø	3ØØ
Ø21	351
Ø22	ØØ3
Ø23	ØØØ

ACR

Load Sense Switches A9, A8 - up

Bootstrap Loader

Octal Address			Octal Data		
ØØØ	0000	BEGIN	Ø41	21	LXI H 3FC2
ØØ1	0001		3Ø2	C2	—
ØØ2	0002		Ø77	3F	—
ØØ3	0003	LOOP	Ø61	31	LXI SP Ø012
ØØ4	0004		Ø22	12	—
ØØ5	0005		ØØØ	ØØ	—
ØØ6	0006		333	DB	IN STAT
ØØ7	0007		ØØ6	Ø6	—
Ø1Ø	0008		Ø17	ØF	RRC
Ø11	0009		33Ø	D8	RC
Ø12	000A		333	DB	IN DATA
Ø13	000B		ØØ7	Ø7	—
Ø14	000C		275	BD	CMP L
Ø15	000D		31Ø	C8	RZ
Ø16	000E		Ø55	2D	DCR L
Ø17	000F		167	77	MOV M,A
Ø2Ø	0010		3ØØ	C0	RNZ
Ø21	0011	EXIT	351	E9	PCHL
Ø22	0012		ØØ3	Ø3	DB 3
023	0013		000	00	DB Ø

Ø23	ØØØ

4PIO

Load Sense Switches A 10 - up

Bootstrap Loader

Octal Address	Octal Data
ØØØ	257
ØØ1	323
ØØ2	Ø4Ø
ØØ3	323
ØØ4	Ø41
ØØ5	Ø76
ØØ6	Ø54
ØØ7	323
Ø1Ø	Ø4Ø
Ø11	Ø41
Ø12	3Ø2
Ø13	Ø77
Ø14	Ø61
Ø15	Ø33
Ø16	ØØØ
Ø17	333
Ø2Ø	Ø4Ø
Ø21	ØØ7
Ø22	33Ø
Ø23	333
Ø24	Ø41
Ø25	275
Ø26	31Ø
Ø27	Ø55
Ø3Ø	167
Ø31	3ØØ
Ø32	351
Ø33	Ø14
Ø34	ØØØ

INDEX

mits

2450 Alamo SE
Albuquerque, NM 87106

Altair Disk Operating System

Errata, June, 1977

1. Page 105, Read Monitor Call, Byte 4, Bit 2:

   Bit 2 on - ASCII. Control/R Control/U

   Bit 2 off - Absolute binary code.

   CHANGE TO:

   Bit 2 off - ASCII. Control/R, Control/U

   Bit 2 on - Absolute binary code.

2. Page 106, Write Monitor Call, Byte 4, Bit 2:

   Bit 2 on - ASCII. Adds nulls to the . . .

   Bit 2 on - Absolute.

   CHANGE TO:

   Bit 2 off - ASCII. Adds nulls to the . . .

   Bit 2 off - Absolute.

3. Page 110, Save Monitor Call. Add after Byte 5, 6:

   ADDITION:

7, 8	address of first byte to be saved
9, 10	address of last byte to be saved
11, 12	starting address

4. Page 110, Load Monitor Call.

   DELETE Bytes 7, 8, 9, 10, 11 and 12.

Altair Disk Operating System

Errata, July, 1977

Page 71. Addition to the end of Section 4-3:

ADDITION:

ORG<e>	Define Origin. The address expression <e> is evaluated and defines the starting address of the generated object code. All names used in <e> must have been defined prior to the ORG statement, and the mode of <e> must not be external.

<u>Disk Operating System</u>

Addendum, July, 1977

1. Page 71, addition after "IFF <e>"

   ADDITION:

IFT <e>	Conditional Assembly - True. If the value of the address expression e is true (≠ 0), then all of the statements until the next END IF are assembled. If the value of e is not true, then the statements are ignored. Conditional assemblies may not be nested.

SOFTWARE AGREEMENT

This software is copyrighted and the property of MITS, Inc., 2450 Alamo SE, Albuquerque, New Mexico, and has been supplied by MITS to you. This software is furnished subject to the following restrictions: It shall not be reproduced or copied without express written permission of MITS, Inc.

To do any of the above without approval by MITS, Inc., will make you liable and open for MITS, Inc. to take legal action against you.

This agreement shall be considered, accepted, and binding upon your receipt of this and any software.

88-DCDD
PARTS LIST
JANUARY, 1976

BAG 1

[illegible]	74L00	101080
[illegible]	74L02	101072
[illegible]	74L04	101073
[illegible]	74L10	101081
[illegible]	74LS11	101089
[illegible]	74L20	[illegible]
[illegible]	74L30	101082
[illegible]	74L73	101084
2	74LS74	101088
[illegible]	74L75	101075
[illegible]	7493	101030
[illegible]	74123	101060
[illegible]	74164	101091
[illegible]	74166	101092
[illegible]	93L16	101093
[illegible]	8T97	101040
[illegible]	8T98	101045
[illegible]	7805	101074

BAG 2

37	.1mf 12v 20%	100348

BAG 3

1	430pf 500v 5%	100322
1	910pf 500v 5%	100356
2	.001mf 1kv 20%	100328
1	.01mf 16v 20%	100321
2	.047mf 100v 5%	100332
2	.1mf 100v 5%	100339
1	.22mf 100v 5%	100349
2	.68mf 100v 5%	100343
1	1.0mf 100v 5%	100373
1	4.7mf 16v	100351
1	10mf 16v	100350
4	33mf 16v	100326

BAG 4

4	220ohm ¼w 5%	101925
4	330ohm ¼w 5%	101926
5	1k ¼w 5%	101928
1	5.6k ¼w 5%	102091
1	6.8k ¼w 5%	101931
7	10k ¼w 5%	101932
2	15k ¼w 5%	102083
1	16k ¼w 5%	101942
2	20k ¼w 5%	101940
[illegible]	39k ¼w 5%	101967

BAG 5

2	IN914	100705
2	#6-32 x 3/8" Screw	100925
2	#6-32 Nut	100933
2	#6 Lock Washer	100942
4	#4-40 x 3/8" Screw	100908
4	#4-40 Nut	100932
4	#4 Lock Washer	100941
1	3ft. 18 Pair Cable	103066
1	37 Pin Adapter Bracket	101795

BAG 6

6	Buss Strips	101805
2	100 Pin Edge Connector	101864
1	DC37S Connector	102114
2	10 Pin Right Angle Wafer	101798
2	20 Pin Right Angle Wafer	101788
2	10 Pin Connector	101720
2	20 Pin Connector	101789
70	Terminal Pins	101723
4	Polarizing Keys	101791
2	Fastwrap	103037
1	Heat Sink Grease	
2	Heat Sink (large)	101870

MISC:

1	Controller PC Board 1	100173
1	Controller PC Board 2	100174
1	Assy, Theory	101531
1	Software + Documentation For DOS	

BAG 1
93L66 sub. for 74L193
9L00 sub. for 74L00
9L04 same as 74L04
74367 or 8097
sub. for 8T97

88-DISC
PARTS LIST
JANUARY, 1976

BAG 1

1	74L30	101082
2	7805	101074
1	7824	101079
4	8T97	101040
1	8T98	101045
1	9601	101033

BAG 2

4	.1mf 12v	100348
3	.1mf 50v	100312
3	33mf 50v	100311
1	500mf 15-25v	100310
1	1000mf 25v	100365
1	2200mf 50v	100376
1	3300mf 16v	100315

BAG 3

3	150 ohm ½w	101915
17	330 ohm ½w	101926
1	1k ½w	101928
1	39k ½w	101967
1	7.5 ohm 5w	101987
1	33 ohm ½w	101921

BAG 4

17	220 ohm ½w	101925
3	RL21	100702
2	VJ048	100711
2	IN4004	100718
1	TIP 145 or 146	102820
1	IN914	100705

BAG 5

1	12ft. 18 Pair Cable	103066
2	6ft. #20 Black	103062
3	2ft. #20 Orange	103063
2	3ft. #26 White	103060

BAG 6

8	#4-40 x 5/16" Screw	100912
2	#4-40 x ½" Screw Flat Head	100938
2	#4-40 x 1" Screw	100913
10	#4-40 Nut	100932
8	#4 Lock Washer	100941
4	#4 Flat Washer	100940
6	#6-32 x 3/8" Pan Head Screw	100925
6	#6-32 x ½" Pan Head Screw	100918
4	#6-32 x 5/8" Pan Head Screw	100916
2	#6-32 x 3/4" Pan Head Screw	100935
4	#6-32 x 1" Pan Head Screw	100919
4	#6-32 x 2" Flat Head Screw	100937
27	#6-32 Nut	100933
35	#6 Lock Washer	100942
1	#6 Ground Lug	101801
2	.15" Spacer	101823
6	5/16" Spacer	101829
2	.6" Spacer	101824
4	#6 Flat Washer	100943
2	#6-32 x ¼" Screw	100917

BAG 7

1	Heat Sink	101775
1	Heat Sink Spacer 5½"	101835
1	Disk Drive Spacer 9"	101841
1	Right Angle Bracket	101717
1	Strain Relief	101719
1	Terminal Block	101868
30	Insulated Terminals	101803
1	Fuse Holder	101813
2	DC37S Connector	102114
2	DC37P Connector	102115
2	DC37 Connector Cover	101799
1	Toggle Switch STI-1C	102566
1	44 Pin Edge Conn., & Key Pin	101800
15	Fastwrap	103037
1	Heat Sink Grease	
1	Fuse 2ASB 3AG	101762

MISC:

1	Power Cord 3 Wire	101742
1	Disk Mechanism (Pertec)	FD-400
1	Case	100511
1	Disk Rail	101862
1	Fan Filter	101757
1	Fan and (4) clips	101869
1	P-8388 Transformer	102612
1	Programmer Transformer	102609
1	Diskette	101712
1	Power Supply PC Board	100171
1	Buffer PC Board	100172
1	"ALTAIR DISK" Nameplace	101808
1	Serial Number Sticker	101833
1	Assy, Theory, OP Manual	101561

POWER SUPPLY

T1. TRANSFORMER [illegible] -2.8 [illegible] C.T. STANCOR P-8388

T2. TRANSFORMER - 8 V. - 12 + [illegible] V. C.T. 400 MA [880-208 (B4240-1167)]

C1 2000 ΩF 50V. ELECTROLYTIC
C2 .1 MF. 50 V. CER. DISC
C3 35 MF. 50V. ELECTROLYTIC

C4 3300 MF. 16 V. ELECTROLYTIC
C5 .1 MF. 50V. CER. DISC
C6 35 MF. 50 V. ELECTROLYTIC

C7 1000 MF. 25 V. ELECTROLYTIC
C8 .1 MF. 50 V. CER. DISC.
C9 35 MF. 50 V. ELECTROLYTIC

BR1 DIODE BRIDGE (VJ04H?)
BR2 DIODE BRIDGE (" ")

D1 1N4004 DIODE ———————— OK
D2 1N4004 DIODE ———————— OK

Q1 PWR. TRANSISTOR TIP 145

X1 REGULATOR 7824 + HEATSINK
X2 " " 7805
X3 " " 7805

R1 33 Ω 1/2 W.
X R2 7.5 Ω 5 W.

VECTOR BOARD

TERMINAL BLOCK CINCH 2 X 10 TERMS, STRAPPED
44 PIN SOCKET 22 × 2 AMP 582017-1 74-49

CONTROLLER BD. #2

ICS.

- 4EA. - 74123 — OK
- 3EA. - 74L73
- [illegible] - 93L16
- 4EA - 74L04
- [illegible] - 74L00 — OK
- 1 EA - 74[illegible]10 — O
- [illegible] - 74L02 — OK
- 3EA - 74L75
- 1EA. - 74166 — OK
- 1EA - 8T98
- 1EA. - 74L74
- 1EA. - 8T97

RESISTORS

3EA 10000 Ω BR-BK-O [illegible] ½ [illegible]
1EA 27000 Ω O [illegible] " "
2EA 15000 Ω BR-GR-O "
1EA 220 Ω R-R-BR
1EA 33[illegible] Ω O-O-[illegible]
1EA 680 [illegible] BLU-[illegible]-BR
1EA 16000 Ω BR [illegible] O "
3EA 1000 Ω BR B[illegible] R "

CAPACITORS

2 EA .001 μf
1 EA 1.0 μf 3/8" T. x 9/16" H. x 3/4" LONG 1TT
1 EA .22 μf 3/8" H x 3/16 T. x 1/2" LONG 1TT
1 EA 4.7 μf electrolytic
1 EA 10. μf electrolytic
23 EA .1 μf 3/16" T. x 1/2" L. x 3/8"
2 EA 35 μf electrolytic

DIODES

2 EA. 1N914

REGULATOR

1 EA 7805

CONTROLLER BL[illegible]

ICS

4 EA	74L03	OK
1 EA	74L02	OK
1 EA	74L20	O
2 EA	74L10	O
1 EA	93L16	
1 EA	74L74	
4 EA	74L73	
1 EA	74L11	
4 EA	74L04	
3 EA	74L00	OK
1 EA	74L30	OK
1 EA	74164	
2 EA	74L75	
1 EA	7493	OK
4 EA	8T97	

RESISTORS

3 EA	330 Ω	[illegible] ¼ W
3 EA	220 Ω	R-R-BR
4 EA	10000 Ω	BR-BL-O
3 EA	20000 Ω	R-BL-O
1 EA	5600 Ω	GRN-BLU-R
2 EA	1000 Ω	BR-BL-R

CAPACITORS

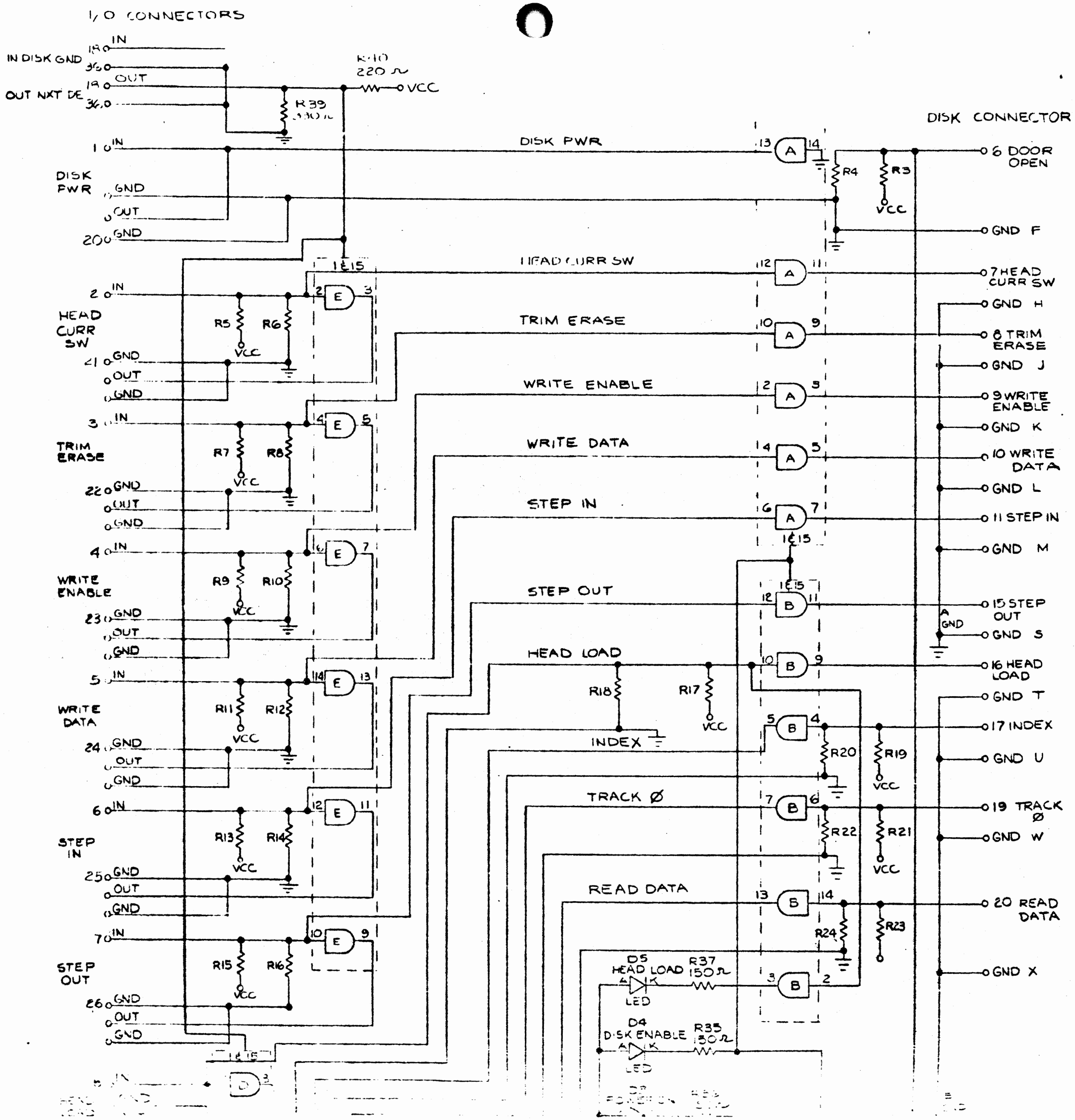
16 EA	.1 μf	
2 EA	.68 μf	3/8" THICK × 9/16" H. × 3/4" LONG. ITT
2 EA	.047 μf	3/16" T × 3/8" H × 1/2" LONG
1 EA	430 PF	500 V. 5%
1 EA	910 PF	
1 EA	33 μf electro[illegible] 16V.	SPRAGUE 30D TE-1159 3/4" × 5/16"
1 EA	35 μf electr[illegible]	
1 EA	.01 μF, CER. DISC .01 μ 50V.	

REGULATOR

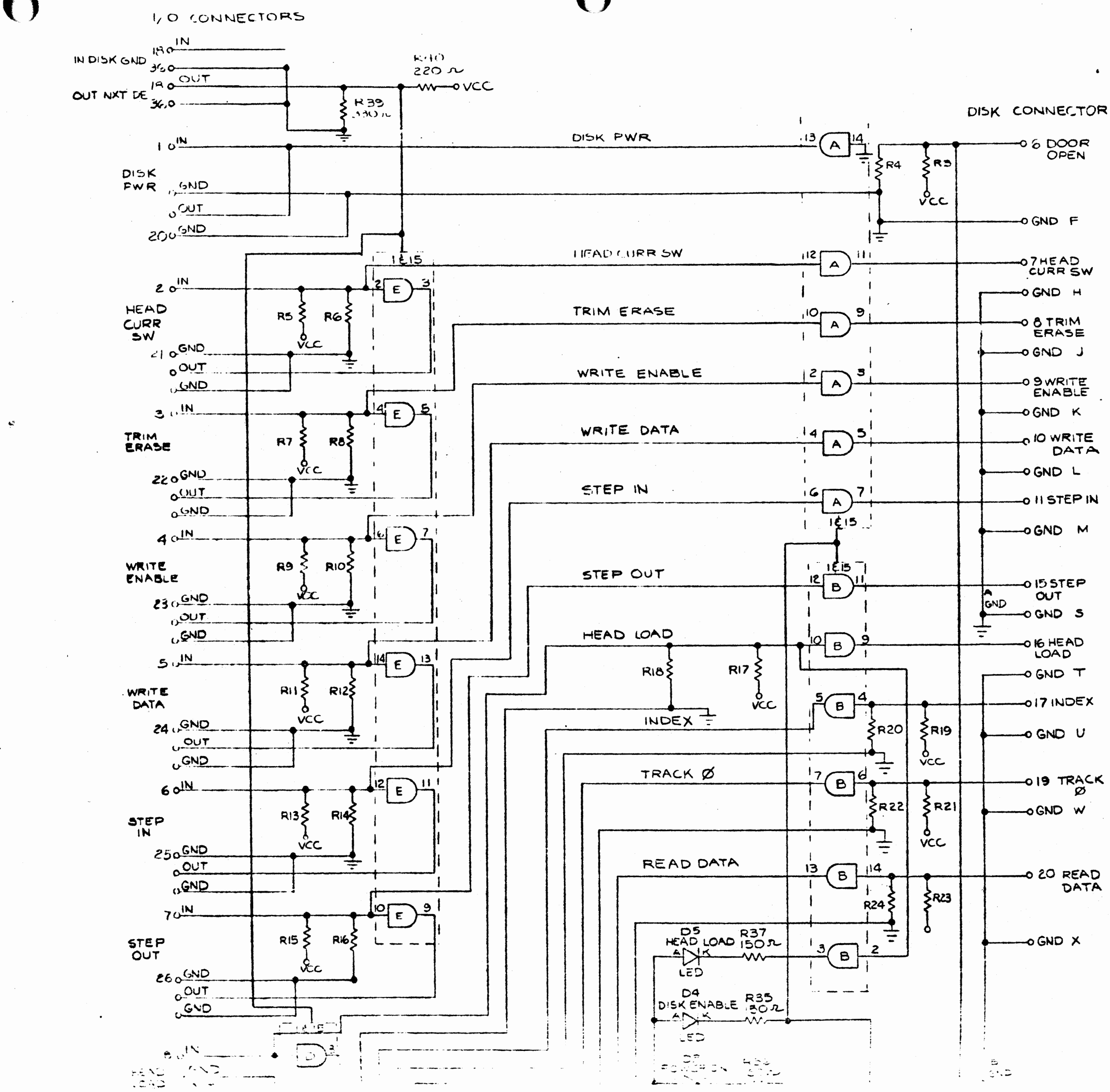
1 EA	7805

MISC

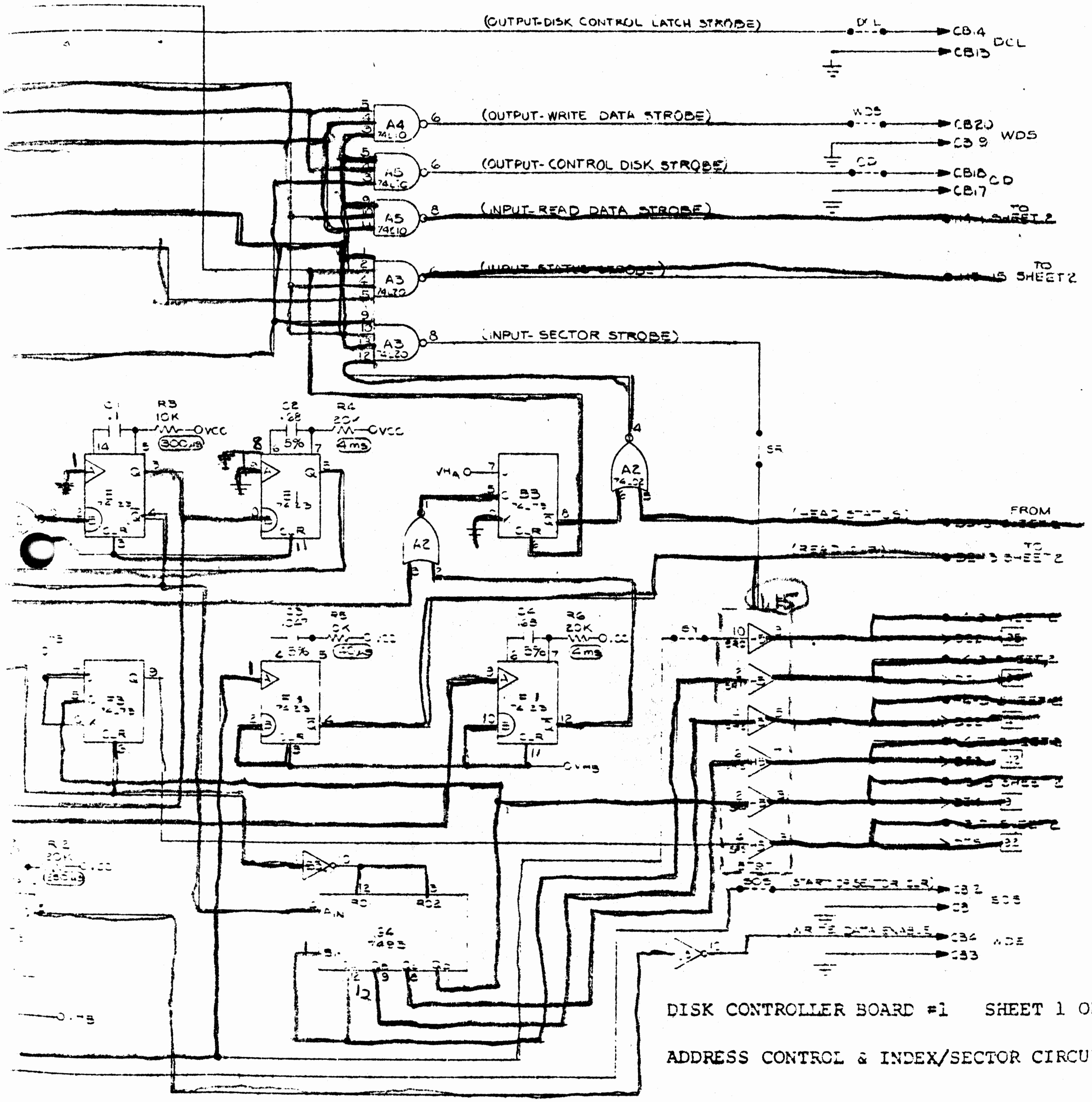
X 1. 5" RACK MOUNT CABI[illegible]
2. FAN 3" OR 5" QUI[illegible] (WHISPER[illegible])
X 3. DISK DRIVE PERTEC FD[illegible] U2 P/N 2[illegible]003-01C
X 4. POWER CORD & STRAIN RELIEF
5. DISK RAIL
X 6. FAN FILTER
X 7. 2EA CANNON 37 PIN PLUGS — ITT. 7615 DC-37P (CRAMER)
X 8. 2EA SHELLS A
9. 2EA CANNON 37 PIN SOCKETS — ITT. 7615 DC-37S (CRAMER)
10. PANEL FUSEHOLDER
11. FUSE 2A. S.B.
12. 1EA. SWITCH MIN. AMERICAN [illegible] 6A. 125 SPDT.
13. SPADE TERMS, 30 EA
14. TIE-WRAPS 15 EA.
15. HEAT SINK GREASE
16. DISKETTE DYSAN 101 [illegible]



I/O CONNECTORS
IN DISK GND
OUT NXT DE
R40
220 Ω
VCC
R39
330 Ω
DISK PWR
HEAD CURR SW
TRIM ERASE
WRITE ENABLE
WRITE DATA
STEP IN
STEP OUT
HEAD LOAD
INDEX
TRACK Ø
READ DATA
DISK CONNECTOR
6 DOOR OPEN
GND F
7 HEAD CURR SW
GND H
8 TRIM ERASE
GND J
9 WRITE ENABLE
GND K
10 WRITE DATA
GND L
11 STEP IN
GND M
15 STEP OUT
GND S
16 HEAD LOAD
GND T
17 INDEX
GND U
19 TRACK Ø
GND W
20 READ DATA
GND X
D5
HEAD LOAD
LED
R37
150 Ω
D4
DISK ENABLE
LED
R35
150 Ω



I/O CONNECTORS
IN DISK GND
OUT NXT DE
R40
220 Ω
VCC
R39
DISK CONNECTOR
DISK PWR
6 DOOR OPEN
R4
R5
GND F
HEAD CURR SW
7 HEAD CURR SW
GND H
TRIM ERASE
8 TRIM ERASE
GND J
WRITE ENABLE
9 WRITE ENABLE
GND K
WRITE DATA
10 WRITE DATA
GND L
STEP IN
11 STEP IN
GND M
STEP OUT
15 STEP OUT
GND S
HEAD LOAD
16 HEAD LOAD
GND T
INDEX
17 INDEX
GND U
TRACK Ø
19 TRACK Ø
GND W
READ DATA
20 READ DATA
GND X
R17
R18
R19
R20
R21
R22
R23
R24
D5
HEAD LOAD
LED
R37
150Ω
D4
DISK ENABLE
R35
150Ω
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16

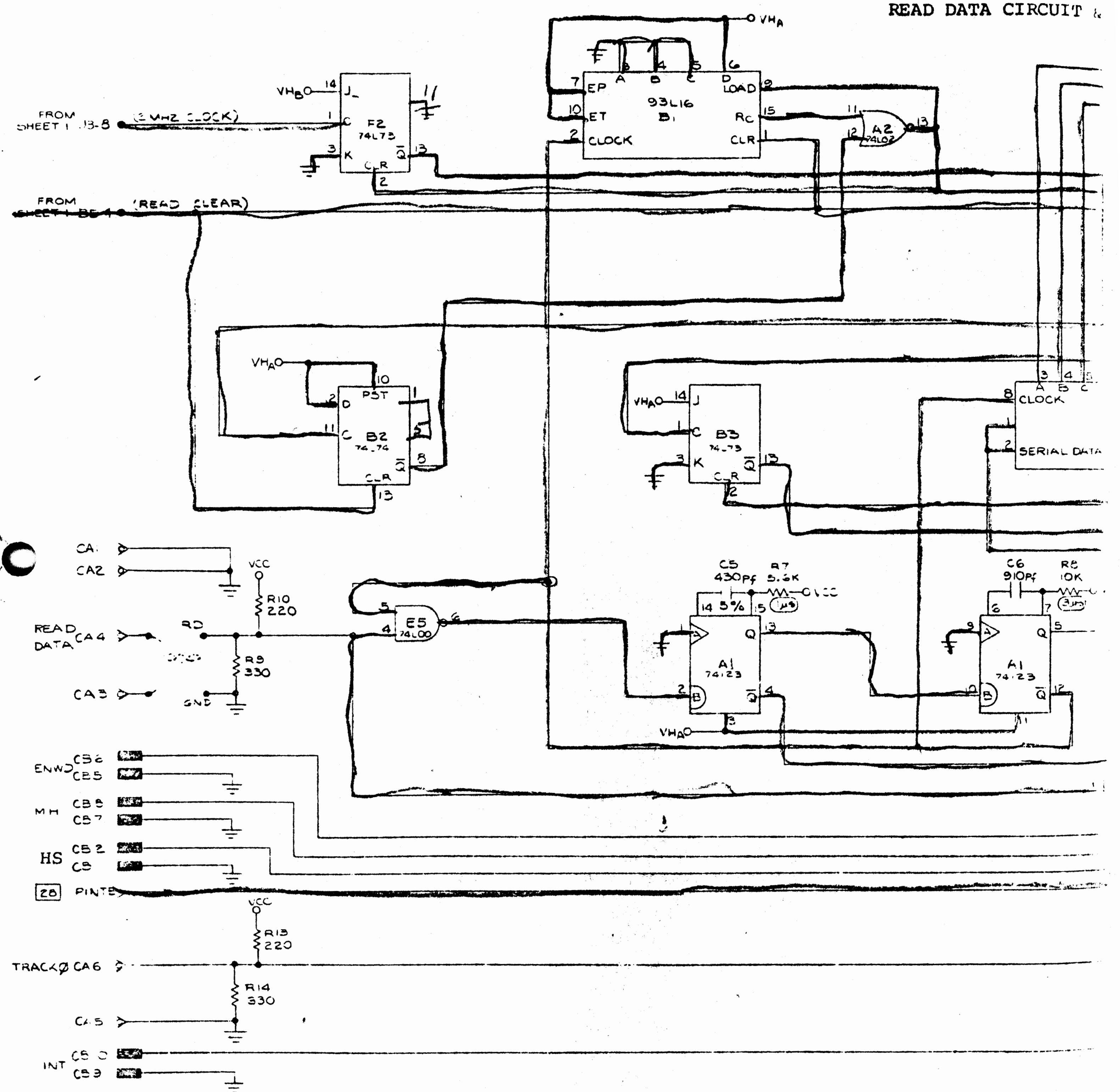


DISK CONTROLLER BOARD #1 SHEET 1 OF 3

ADDRESS CONTROL & INDEX/SECTOR CIRCUIT

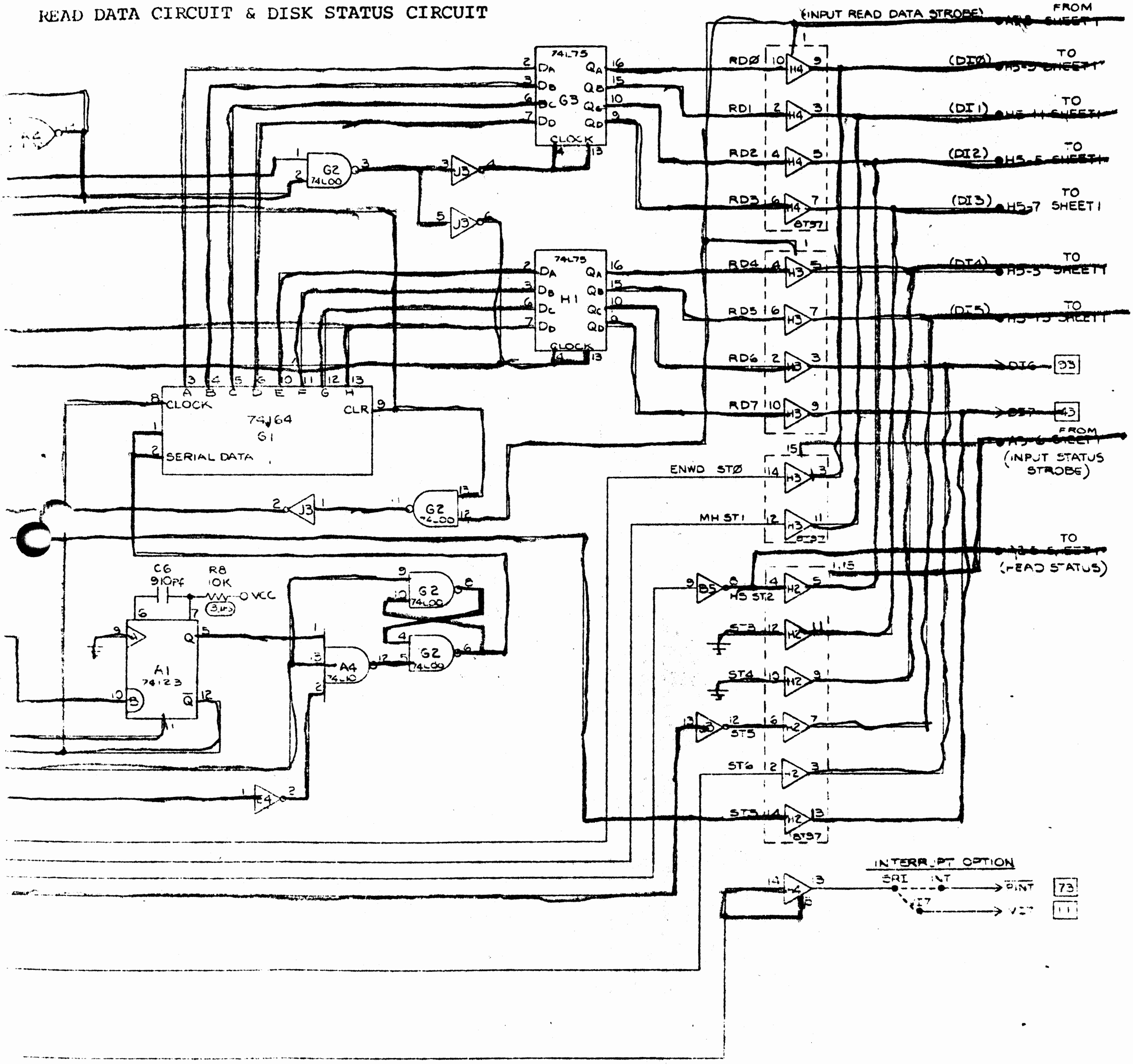
DISK CONTROLLER BO...

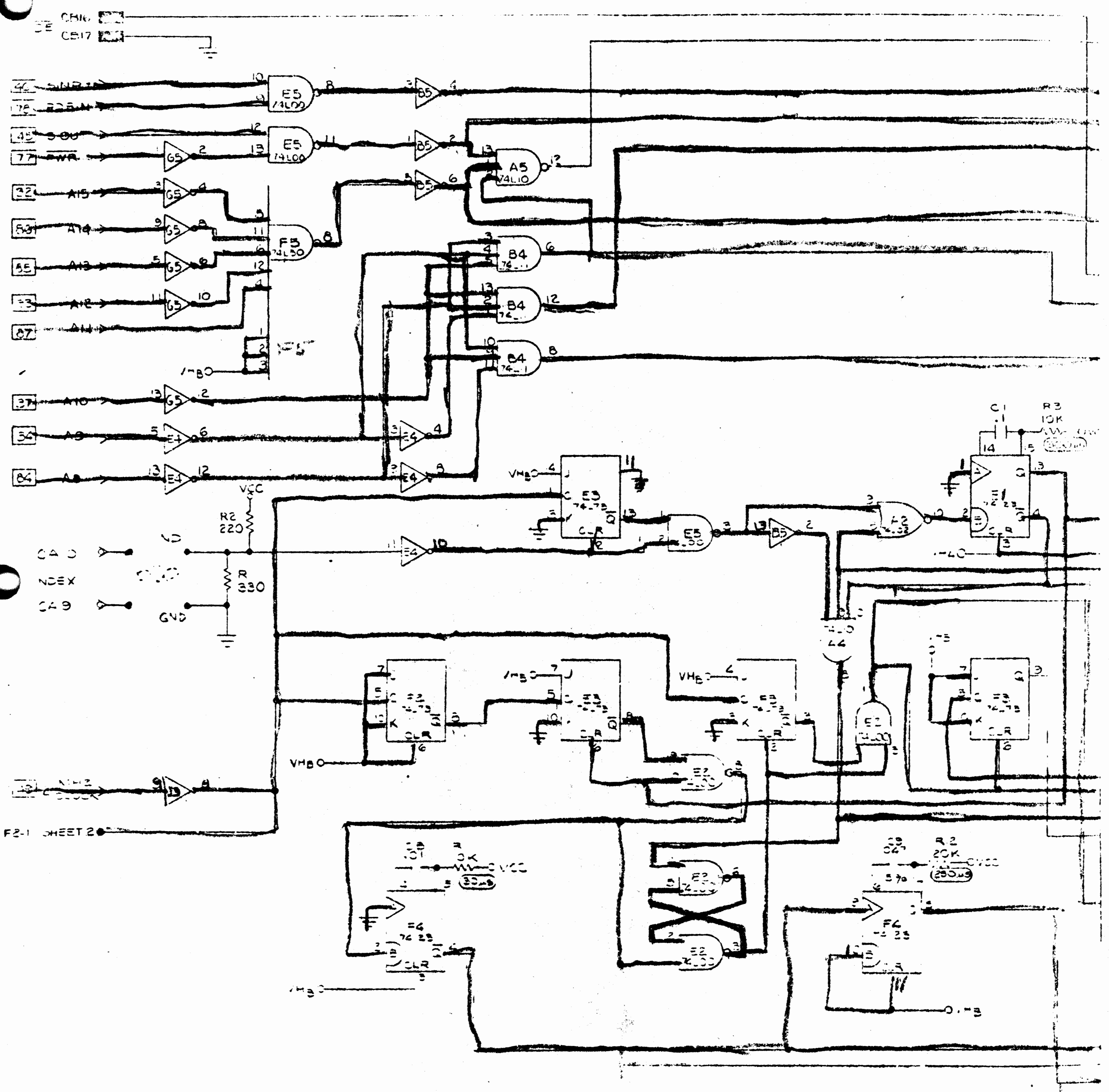
READ DATA CIRCUIT &...



DISK CONTROLLER BOARD #1   SHEET 2 OF 3

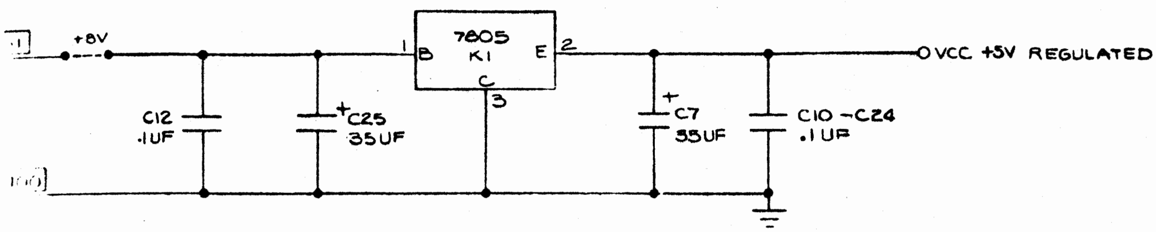
READ DATA CIRCUIT & DISK STATUS CIRCUIT



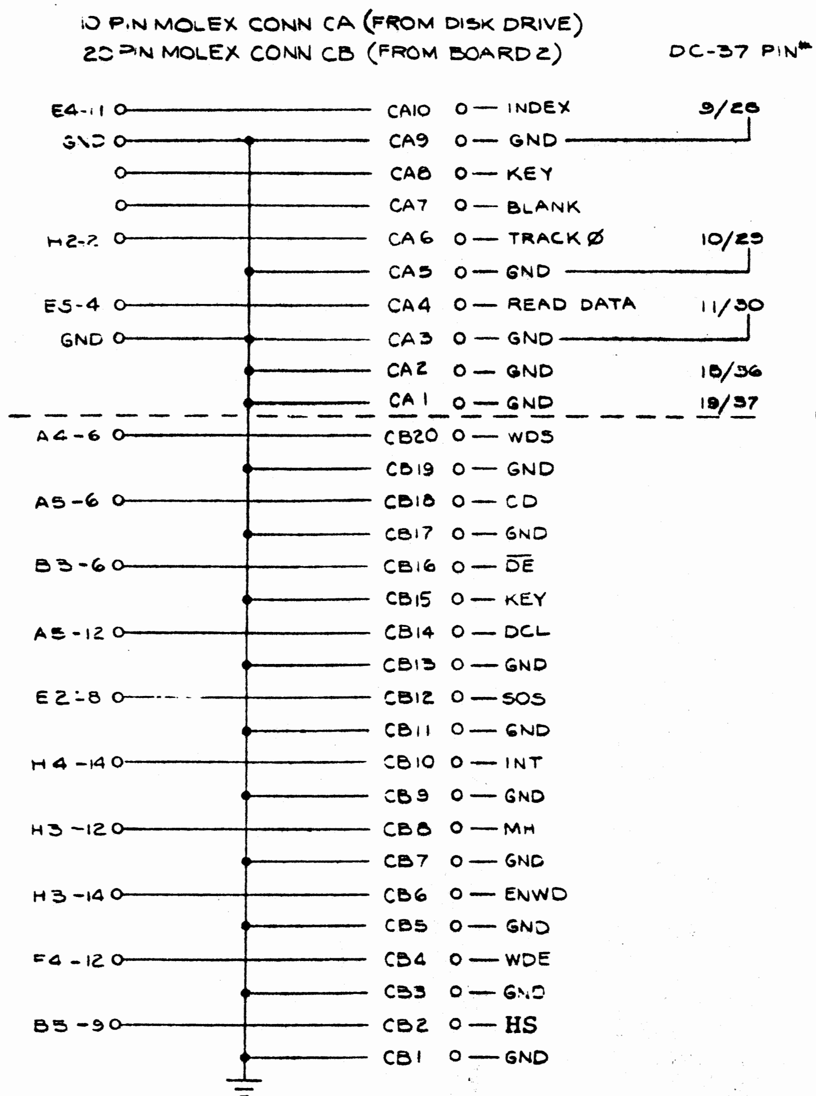


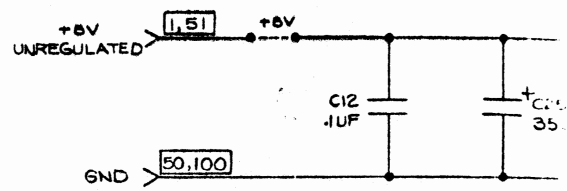
CB16
CB17
E5
74L00
B5
A5
74L10
G5
F5
74L30
B4
E4
VHB
VCC
R2
220
R
330
GND
INDEX
E3
74L73
A2
E1
C1
R3
10K
E2
74L00
F4
74123
F2-1 SHEET 2

DISK CONTROLLER BOARD #1 SHEET 3 OF 3



POWER SUPPLY CONNECTIONS			
REF / SPARE	TYPE	VCC	GND
[redacted]	74L04	[redacted]	[redacted]
[redacted]	74L00	[redacted]	[redacted]
[redacted]	74L11	[redacted]	[redacted]
[redacted]	74L10	[redacted]	[redacted]
[redacted]	74L20	[redacted]	[redacted]
[redacted]	74L30	[redacted]	[redacted]
[redacted]	74L02	[redacted]	[redacted]
[redacted]	8T97	[redacted]	[redacted]
[redacted]	74L73	[redacted]	[redacted]
[redacted]	74123	[redacted]	[redacted]
[redacted]	7493	[redacted]	[redacted]
[redacted]	93_6	[redacted]	[redacted]
[redacted] 12	74L74	14	7
[redacted]	74L75	5	12
[redacted]	74.64	14	7
K1	7805	2	3

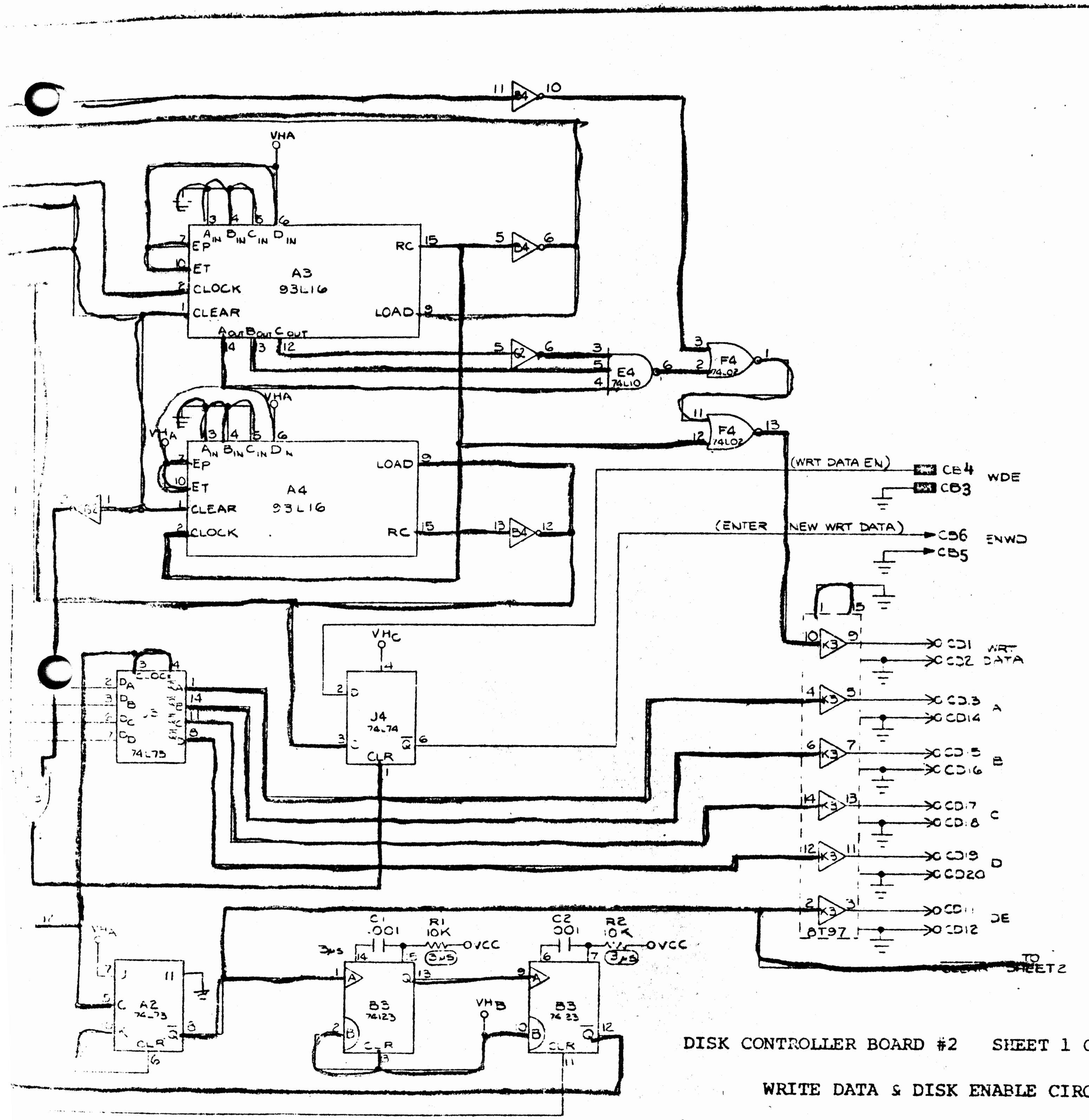




POWER
REF / SPARE
B5, E4, C5, J3
E2, E5, G2
B4
A4, A5
A3
F5
A2
H2, H3, H4, H
B3, E3, F2, F3
A1, E1, F1, F4
G4
B1
B2 ½
G3, H1
G1
K1

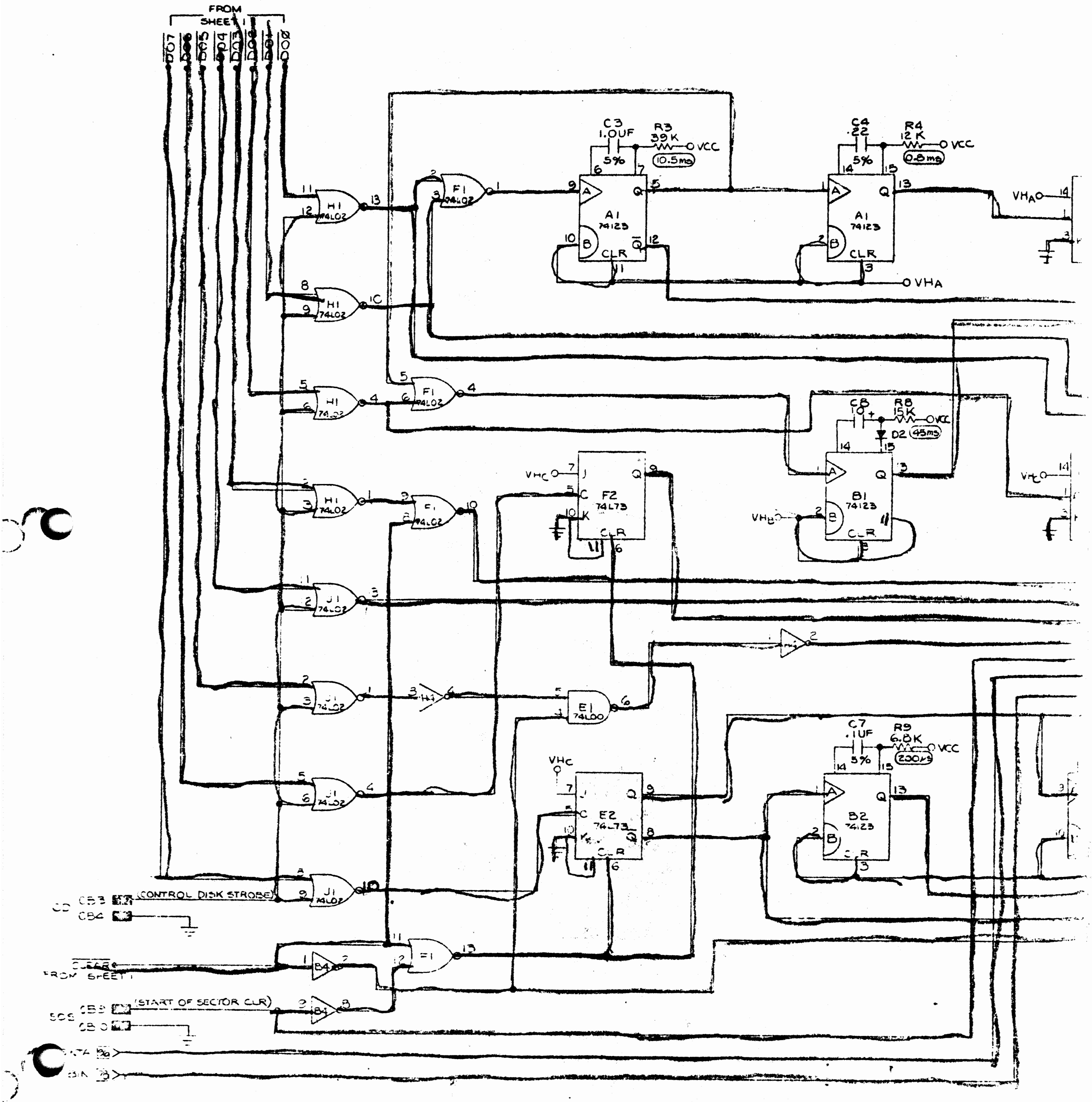
NOTES:

1. ALL DIODES IN914 UNLESS OTHERWISE SPECIFIED.
2. ALL RESISTORS IN OHMS, ½ W UNLESS OTHERWISE SPECIFIED.
3. ALL CAPACITORS IN µF UNLESS OTHERWISE SPECIFIED.
4.
   - A. ALTAIR BUSS INPUT TO BD.
   - B. ALTAIR BUSS OUTPUT FROM BD.
   - C. DISK CONNECTOR INPUT TO BD.
   - D. DISK CONNECTOR OUTPUT FROM BD.
   - E. INTERBOARD OUTPUT.
   - F. INTERBOARD INPUT.
   - G. ON BOARD CONNECTION
   - H. [41] ALTAIR BUSS #
   - I. (250ns) ONE SHOT TIME CONSTANT ±10%.
   - J. JUMPER
5. A. VCC — R15 1K — VHA
   B. VCC — R16 1K — VHB

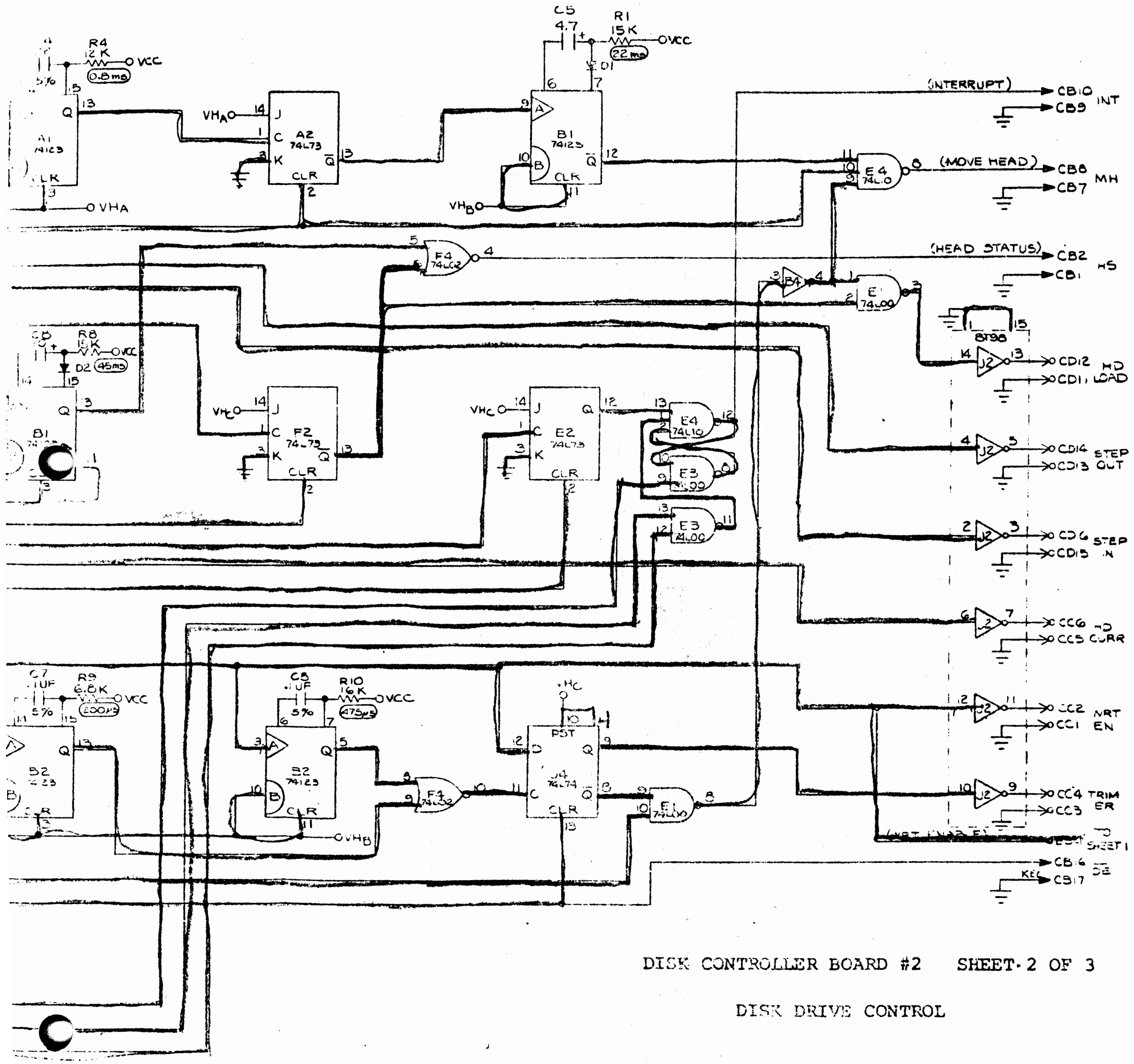


DISK CONTROLLER BOARD #2 SHEET 1 OF 3

WRITE DATA & DISK ENABLE CIRCUITS

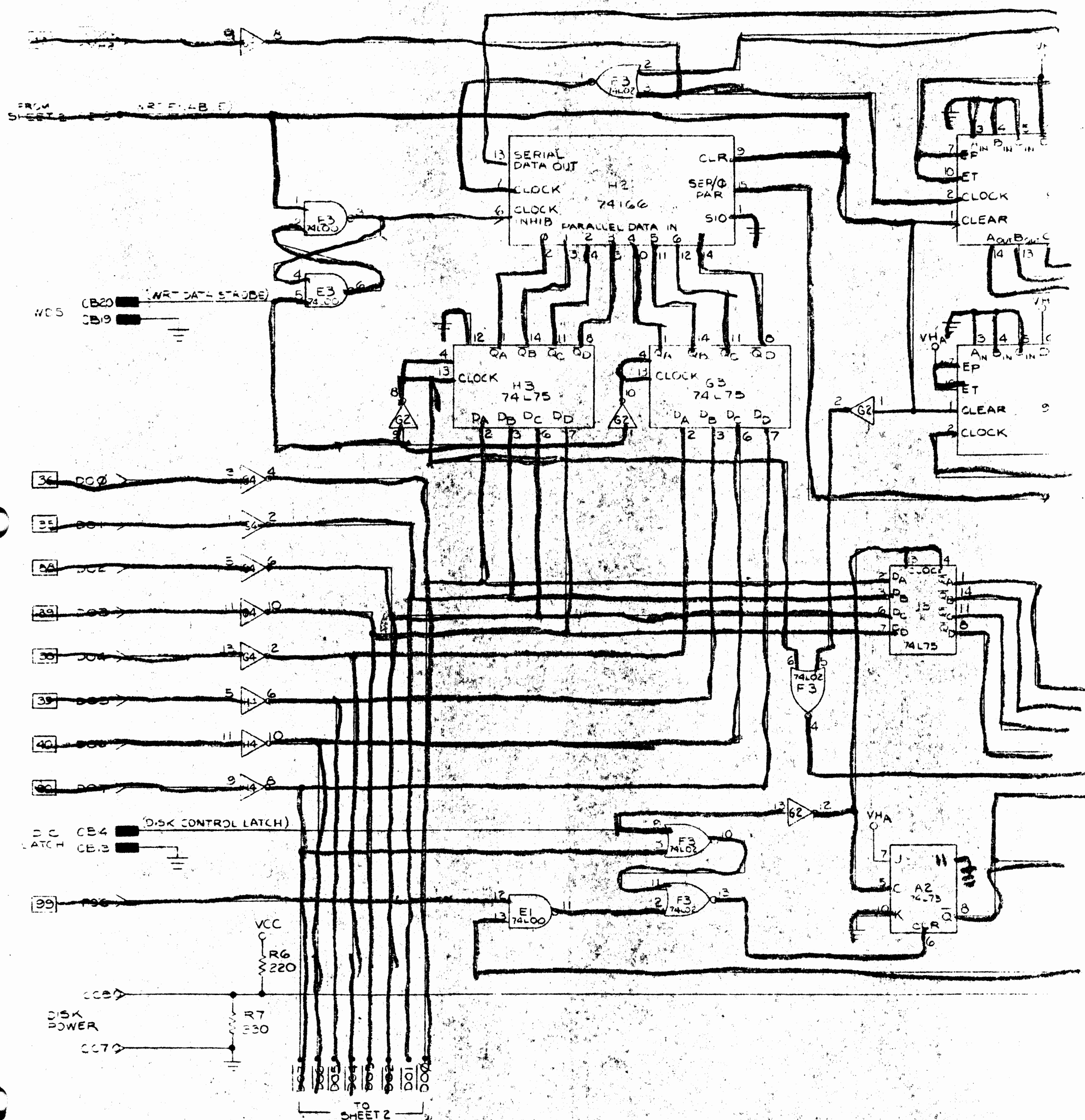


FROM SHEET 1
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DOØ
C3 1.0UF 5%
R3 39K VCC
10.5ms
C4 .22 5%
R4 12K VCC
0.8ms
A1 74123
H1 74LO2
F1 74LO2
VHA
C8 10
R8 15K VCC
D2 45ms
B1 74123
F2 74L73
VHC
VHB
J1 74LO2
E1 74LOO
C7 .1UF 5%
R9 6.8K VCC
200μs
E2 74L73
B2 74123
CB3 (CONTROL DISK STROBE)
CB4
CLEAR FROM SHEET 1
B4
SOS CB5 (START OF SECTOR CLR)
CB6

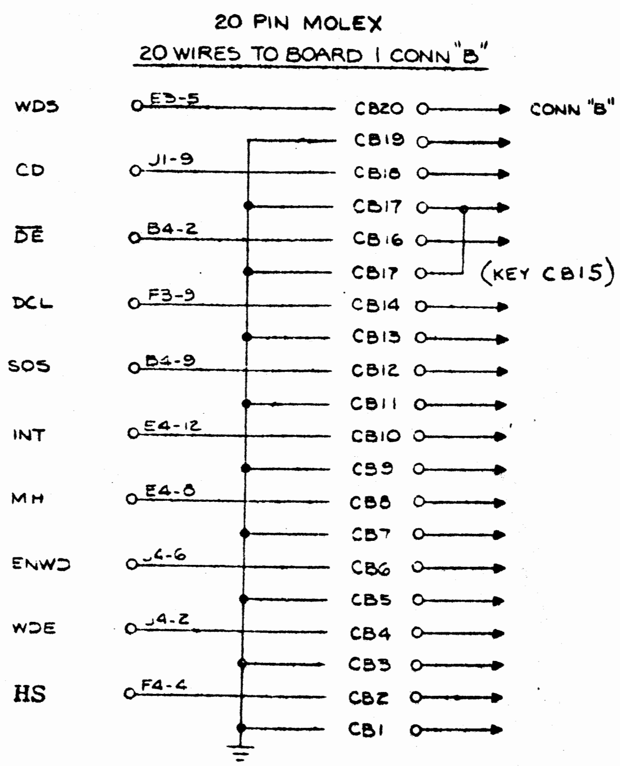


DISK CONTROLLER BOARD #2 SHEET 2 OF 3

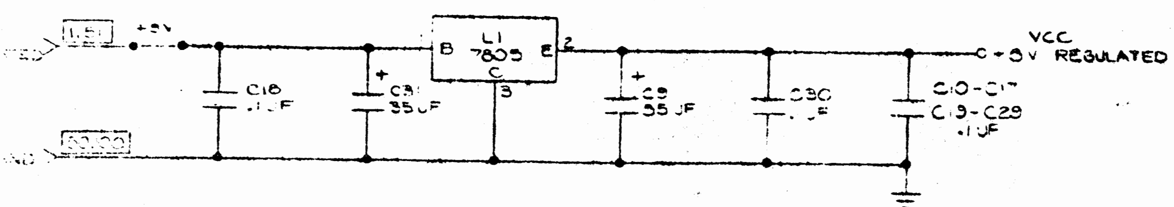
DISK DRIVE CONTROL


FROM SHEET 2
SERIAL DATA OUT
CLOCK
CLOCK INHIB
H2 74166
CLR
SER/Φ PAR
SIO
PARALLEL DATA IN
CLOCK
CLEAR
H3 74L75
G3 74L75
(WRT DATA STROBE)
WDS
CB20
CB19
DOØ
(DISK CONTROL LATCH)
DC LATCH
CB4
CB3
J3 74L75
A2 74L73
E1 74L00
F3 74L02
E3 74L00
VCC
R6 220
R7 330
DISK POWER
CC8
CC7
TO SHEET 2
VHA

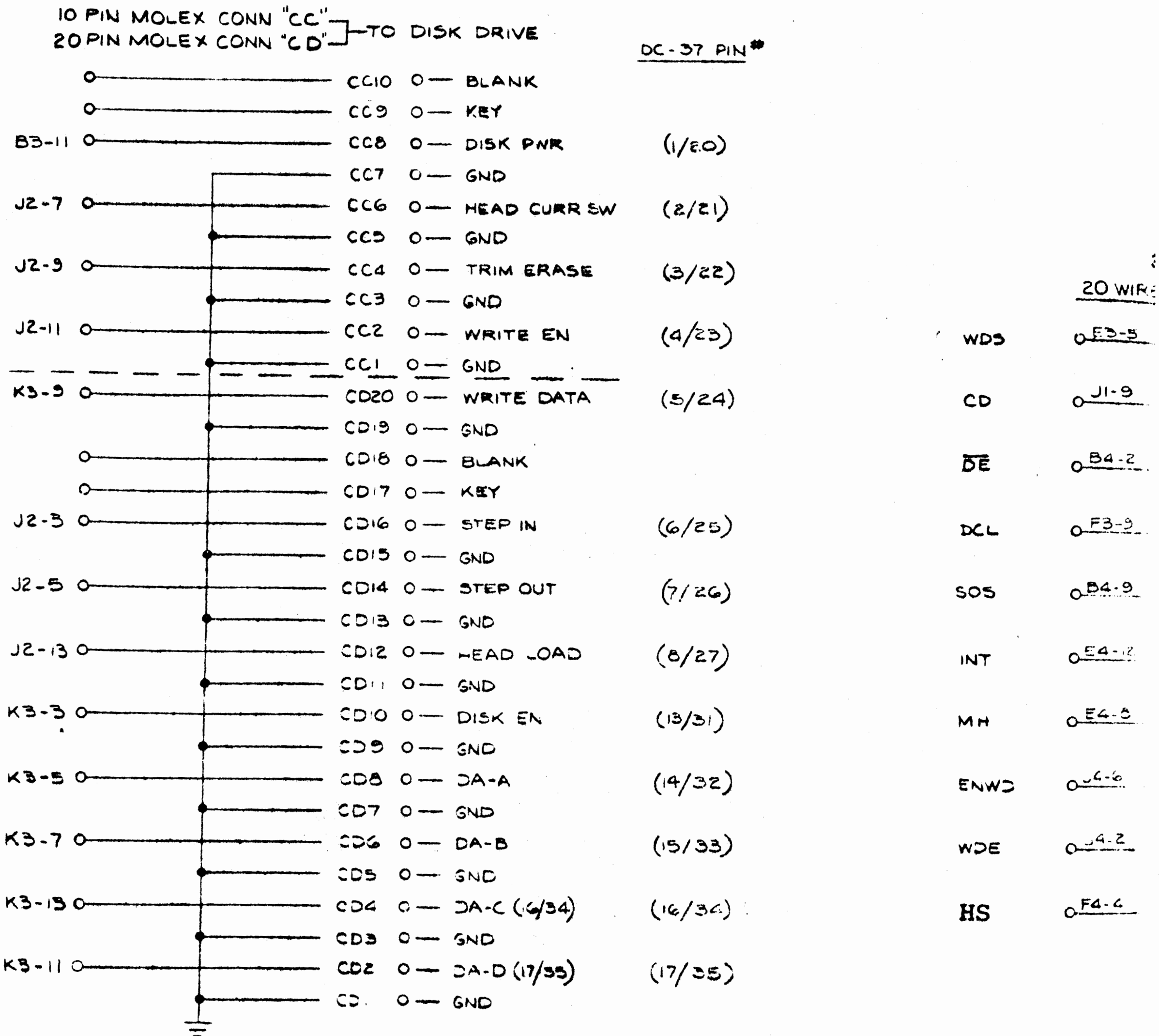
DISK 2-1



POWER CONNECTIONS			
REF	TYPE	VCC	GND
[redacted]	74L02	[redacted]	[redacted]
[redacted]	74L00	[redacted]	[redacted]
[redacted]	74L04	[redacted]	[redacted]
[redacted]	8T98	[redacted]	[redacted]
[redacted]	74L10	[redacted]	[redacted]
[redacted]	74166	[redacted]	[redacted]
[redacted]	74L75	[redacted]	[redacted]
[redacted]	93L16	[redacted]	[redacted]
[redacted]	74L74	[redacted]	[redacted]
[redacted]	74L73	4	11
[redacted]	74123	[redacted]	[redacted]
[redacted]	8T97	[redacted]	[redacted]
L1	7805	2	3

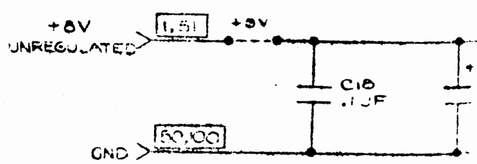


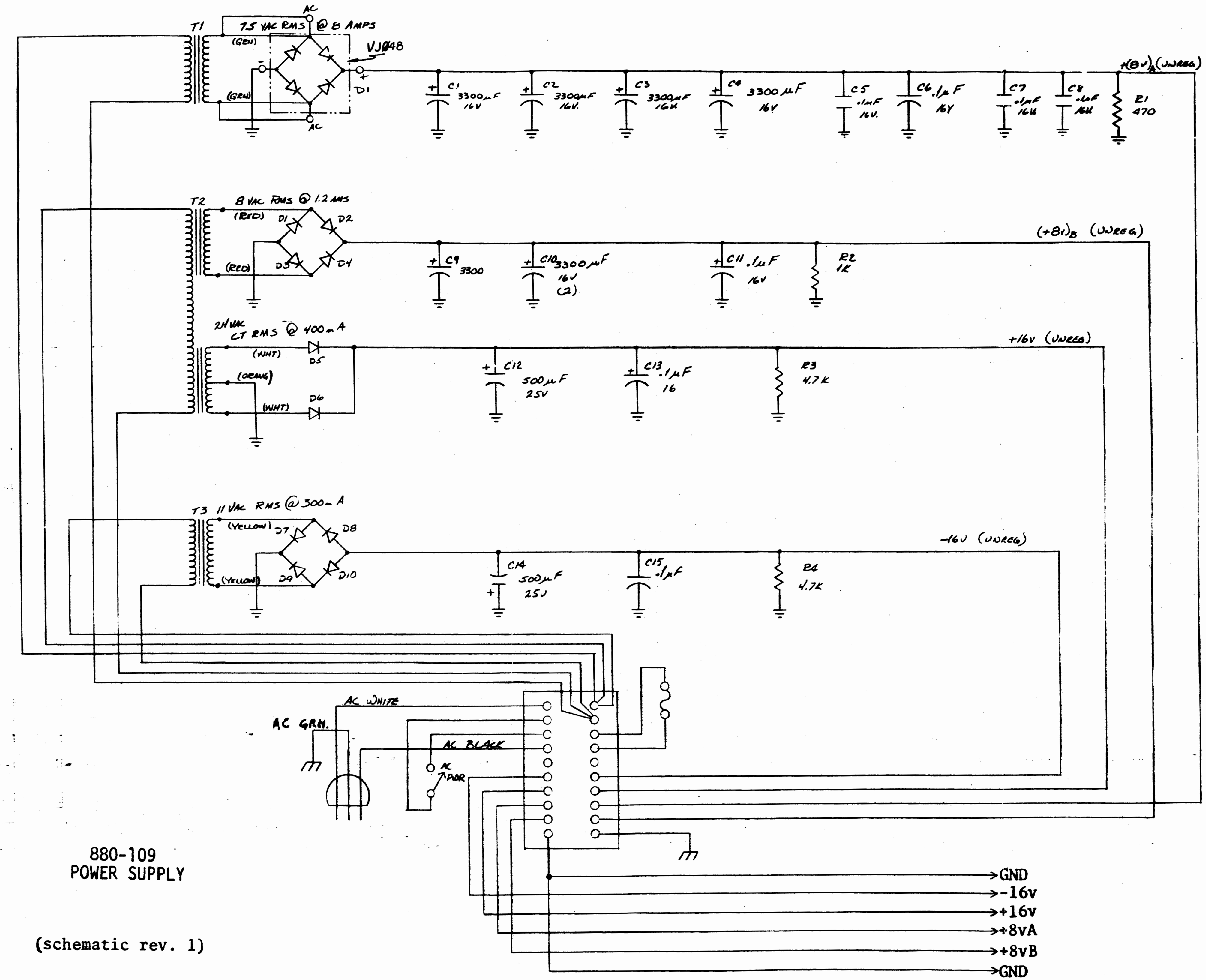
DISK CONTROLLER BOARD #2 SHEET 3 OF 3



NOTES:

1. ALL RESISTORS 1/2 W UNLESS SPECIFIED.
2. ALL CAPACITORS IN UF.
3. ALL DIODES 1N914.
4. A>— ALTAIR BUSS INPUT TO BD.
   B—> ALTAIR BUSS OUTPUT FROM BD.
   C>— DISK CONNECTOR INPUT TO BD.
   D—> DC 34 CONNECTOR OUTPUT FROM BD.
   E—> INTERBOARD OUTPUT.
   F—■ INTERBOARD INPUT.
   G ● ON BOARD CONNECTION.
   H [37] ALTAIR BUSS #
   I (40ms) ONE SHOT TIME CONSTANT ± 10%.
   J ●- - -● JUMPER.
5. VCC O—R11 1K—O VHA
   VCC O—R12 1K—O VHB
   VCC O—R13 1K—O VHC

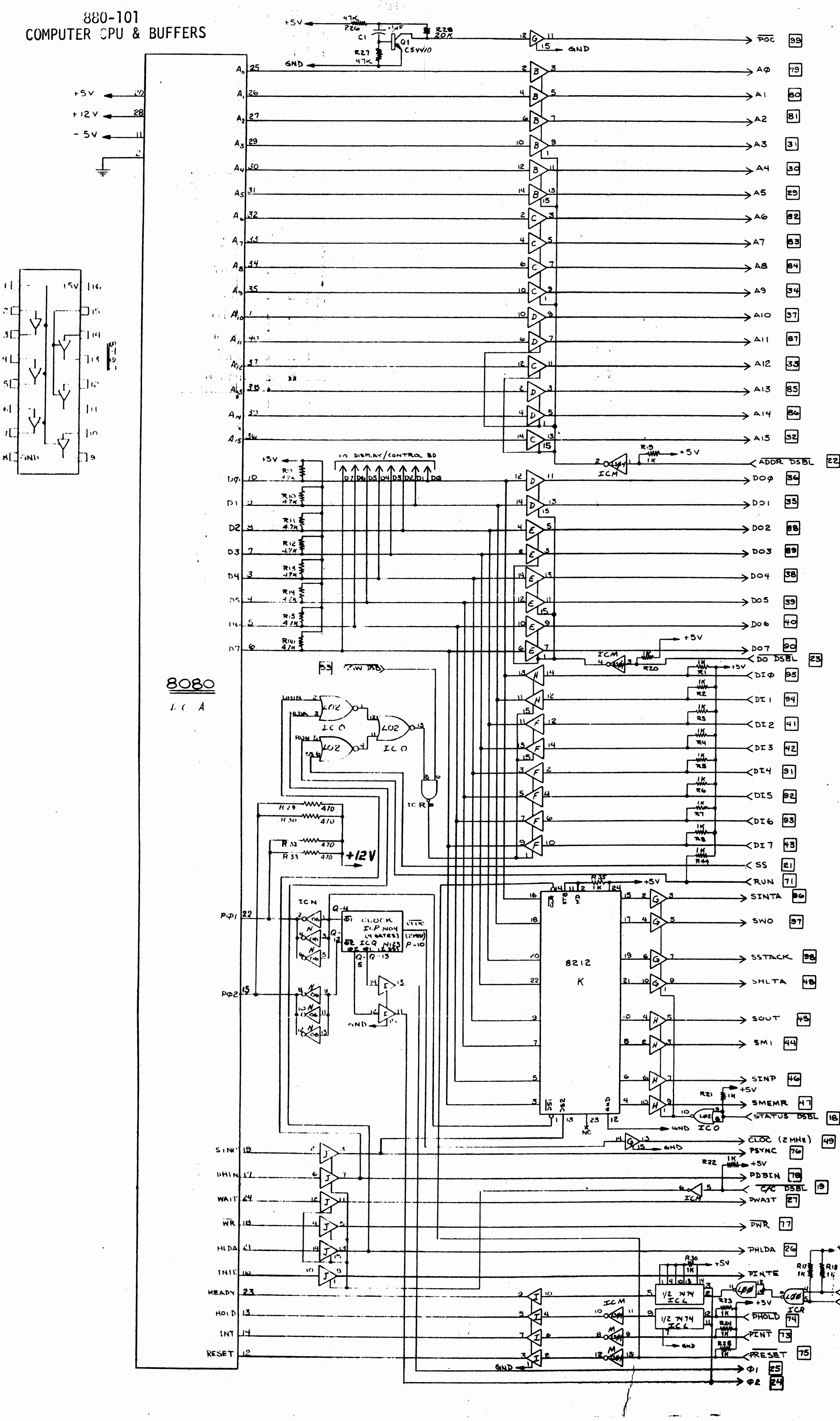


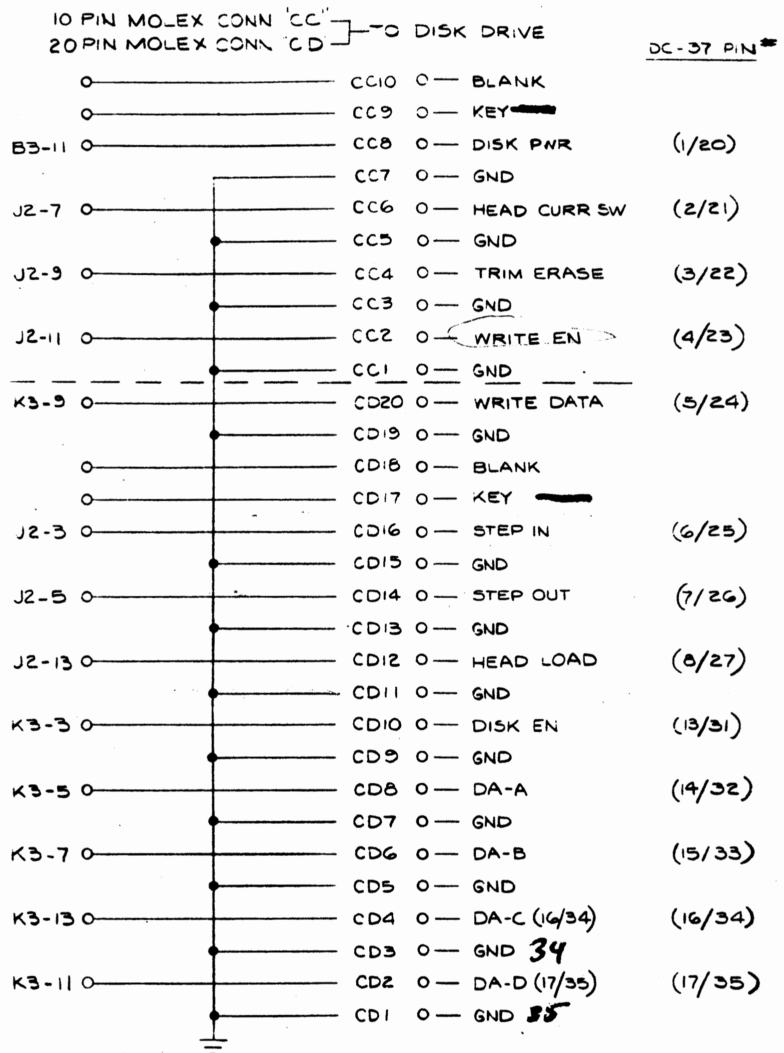


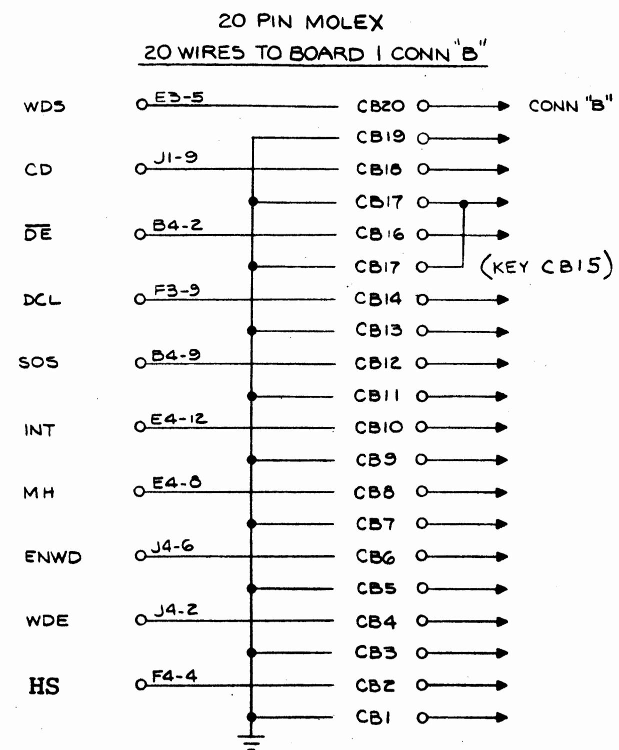
880-109
POWER SUPPLY

(schematic rev. 1)

880-101
COMPUTER CPU & BUFFERS

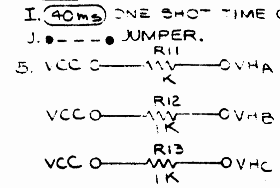


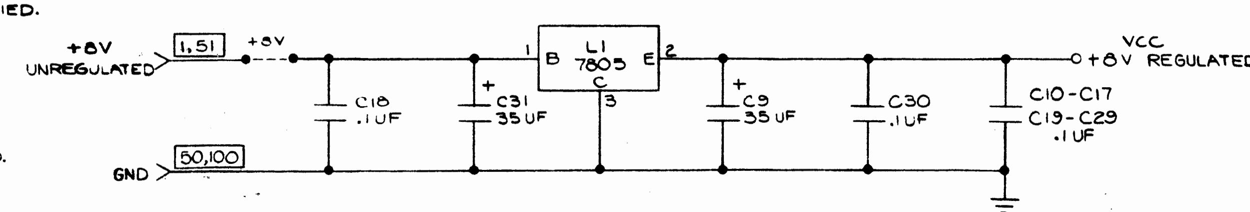




POWER CONNECTIONS			
REF	TYPE	VCC	GND
F1,F3,F4,H1,J1	74L02	14	7
E1,E3	74L00	14	7
B4,G2,G4,H4	74L04	14	7
J2	8T98	16	8
E4	74L10	14	7
H2	74166	16	8
G3,H3,J3	74L75	5	12
A3,A4	93L16	16	8
J4	74L74	14	7
A2,E2,F2	74L73	4	11
A1,B1,B2,B3	74123	16	8
K3	8T97	16	8
L1	7805	2	3

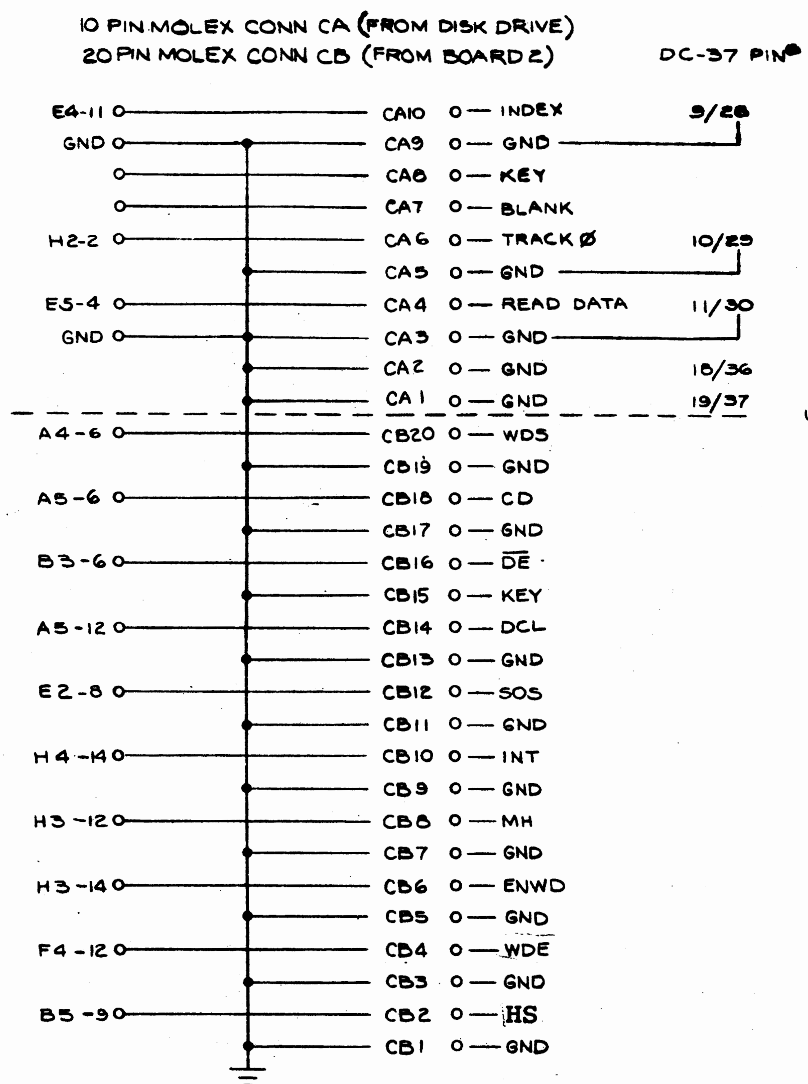
NOTES:
1. ALL RESISTORS 1/2 W UNLESS SPECIFIED.
2. ALL CAPACITORS IN UF.
3. ALL DIODES IN914.
4. A ALTAIR BUSS INPUT TO BD.
B ALTAIR BUSS OUTPUT FROM BD.
C DISK CONNECTOR INPUT TO BD.
D DISK CONNECTOR OUTPUT FROM BD.
E INTERBOARD OUTPUT.
F INTERBOARD INPUT.
G ● ON BOARD CONNECTION.
H. [37] ALTAIR BUSS #.
I. (40ms) ONE SHOT TIME CONSTANT ± 10%.
J. ●- - - -● JUMPER.
5.

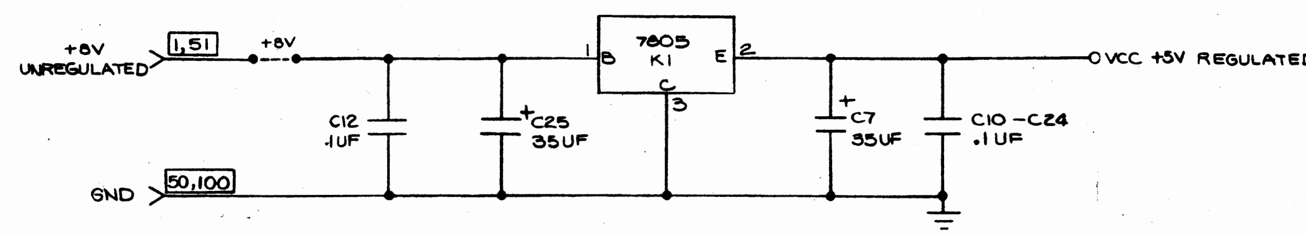




DISK CONTROLLER BOARD #2 SHEET 3 OF 3

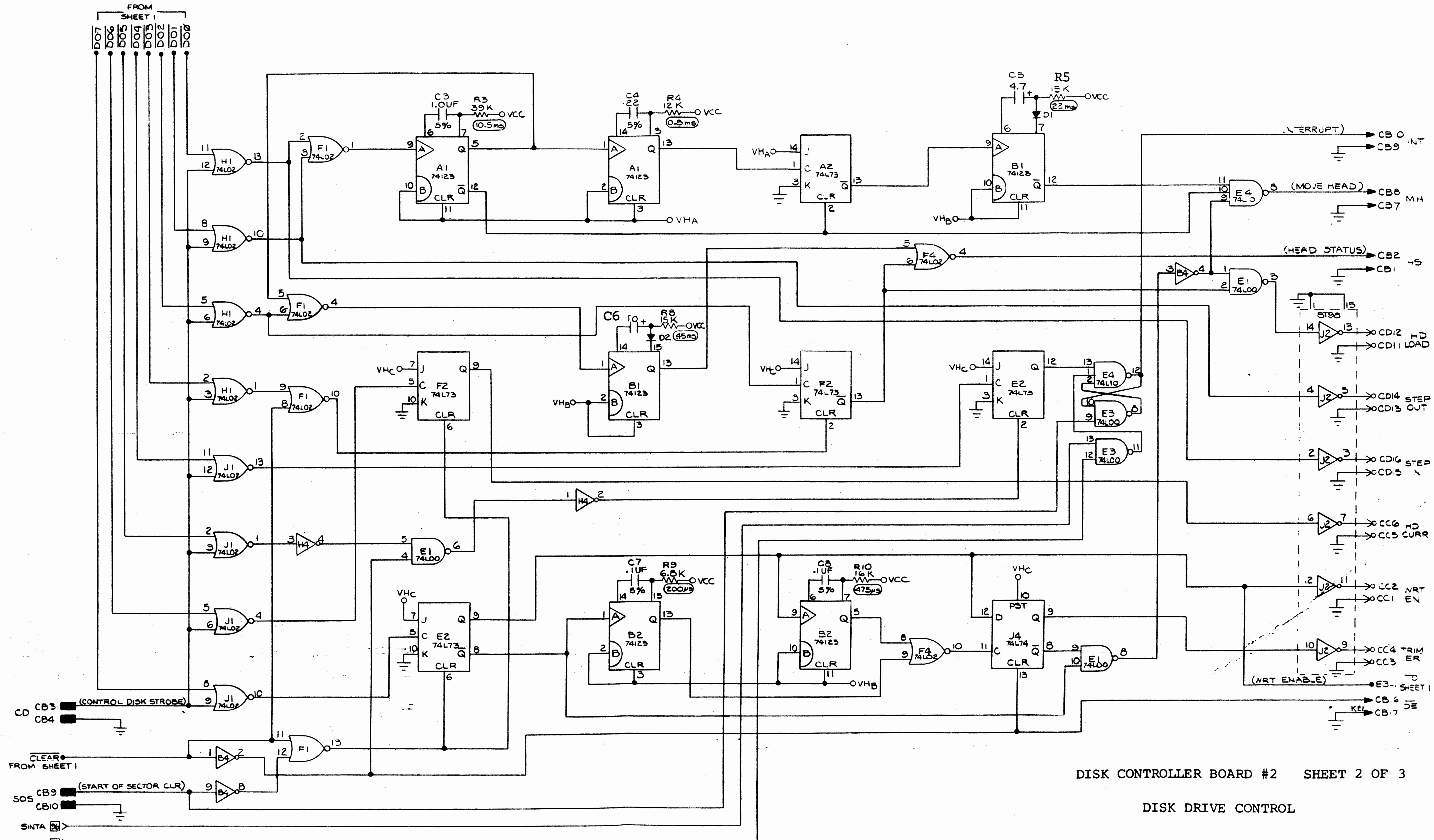
Disc 2-3





NOTES:
1. ALL DIODES IN914 UNLESS OTHERWISE SPECIFIED.
2. ALL RESISTORS IN OHMS, 1/2 W UNLESS OTHERWISE SPECIFIED.
3. ALL CAPACITORS IN µf UNLESS OTHERWISE SPECIFIED.
4.
A. ALTAIR BUSS INPUT TO BD.
B. ALTAIR BUSS OUTPUT FROM BD.
C. DISK CONNECTOR INPUT TO BD.
D. DISK CONNECTOR OUTPUT FROM BD.
E. INTERBOARD OUTPUT.
F. INTERBOARD INPUT.
G. ON BOARD CONNECTION
H. [41] ALTAIR BUSS #
I. (280µs) ONE SHOT TIME CONSTANT ± 10 %.
J. JUMPER
5. A. VCC R15 1K VHA
B. VCC R16 1K VHB

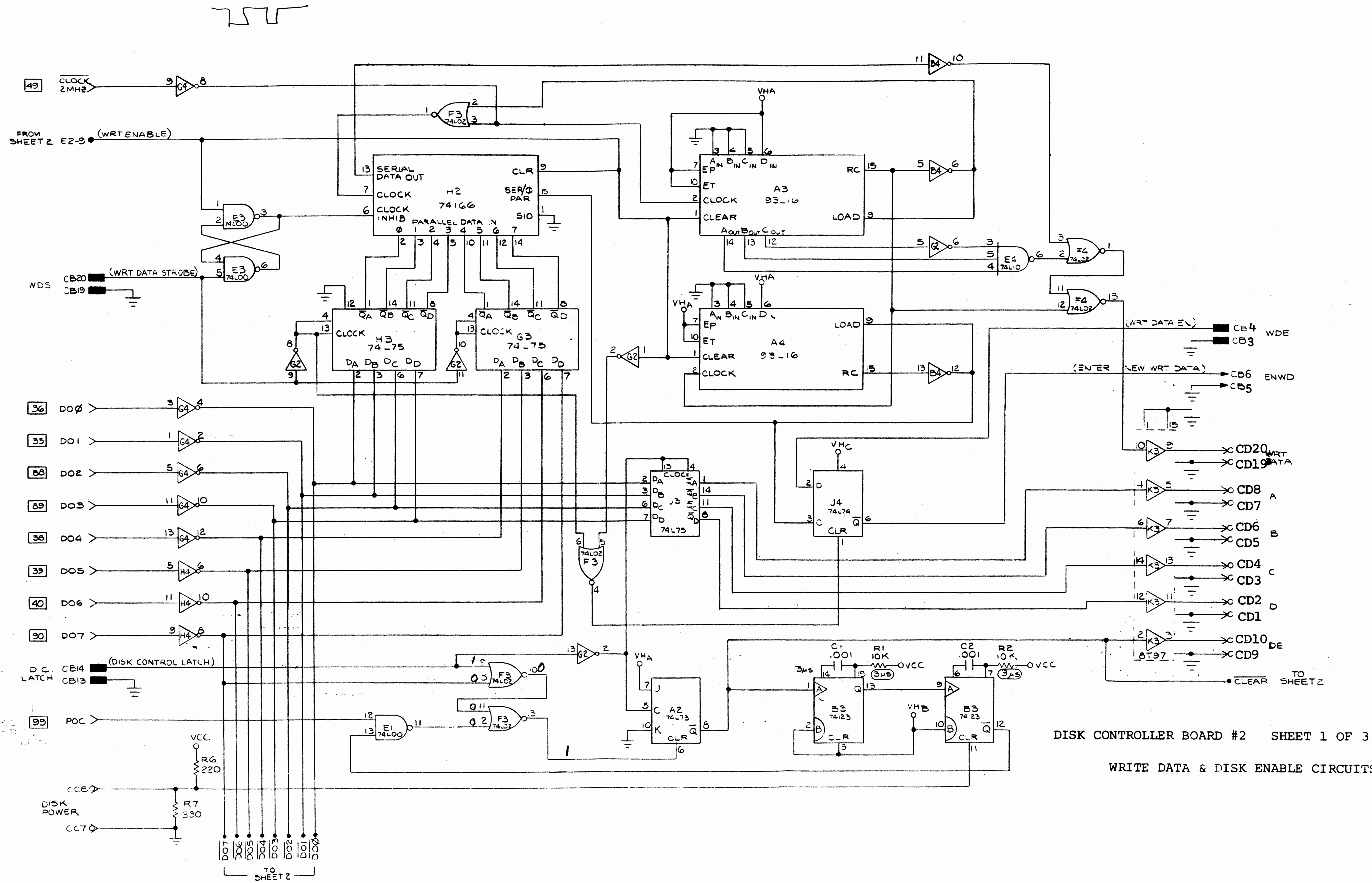
POWER SUPPLY CONNECTIONS			
REF / SPARE	TYPE	VCC	GND
B5,E4,G5,J3	74L04	14	7
E2,E5,G2	74L00	14	7
B4	74L11	14	7
A4, A5	74L10	14	7
A3	74L20	14	7
F5	74L30	14	7
A2	74L02	14	7
H2,H3, H4,H5	8T97	16	8
B3,E3,F2,F3	74L73	4	11
A1,E1,F1,F4	74123	16	8
G4	7493	5	10
B1	93L16	16	8
B2 1/2	74L74	14	7
G3,H1	74L75	5	12
G1	74164	14	7
K1	7805	2	3



DISK CONTROLLER BOARD #2 SHEET 2 OF 3

DISK DRIVE CONTROL

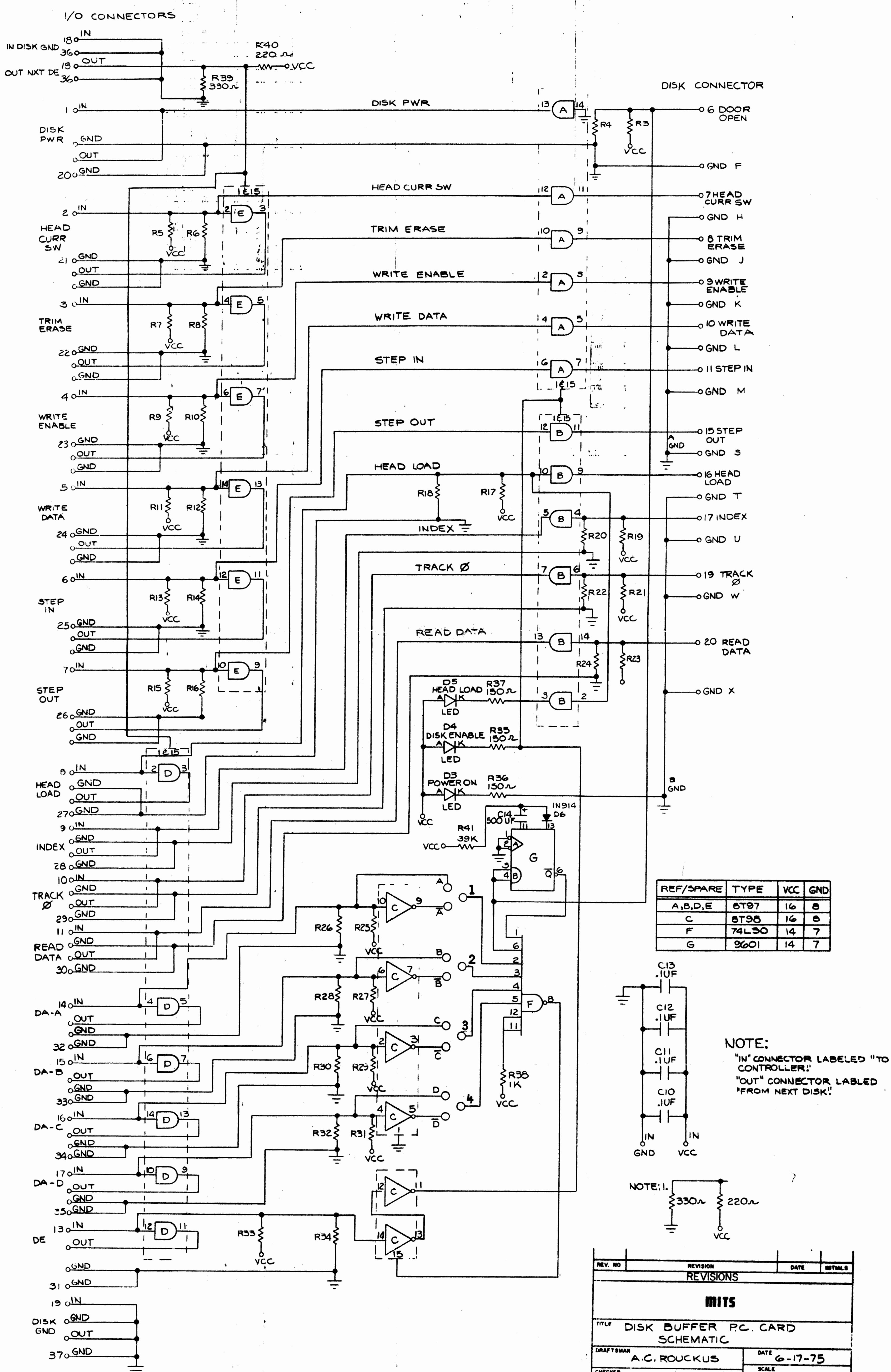
Disc 2-2



DISK CONTROLLER BOARD #2 SHEET 1 OF 3

WRITE DATA & DISK ENABLE CIRCUITS

Disc PC Card



REF/SPARE	TYPE	VCC	GND
A,B,D,E	8T97	16	8
C	8T98	16	8
F	74L50	14	7
G	9601	14	7

REVISIONS

MITS

TITLE: DISK BUFFER P.C. CARD SCHEMATIC

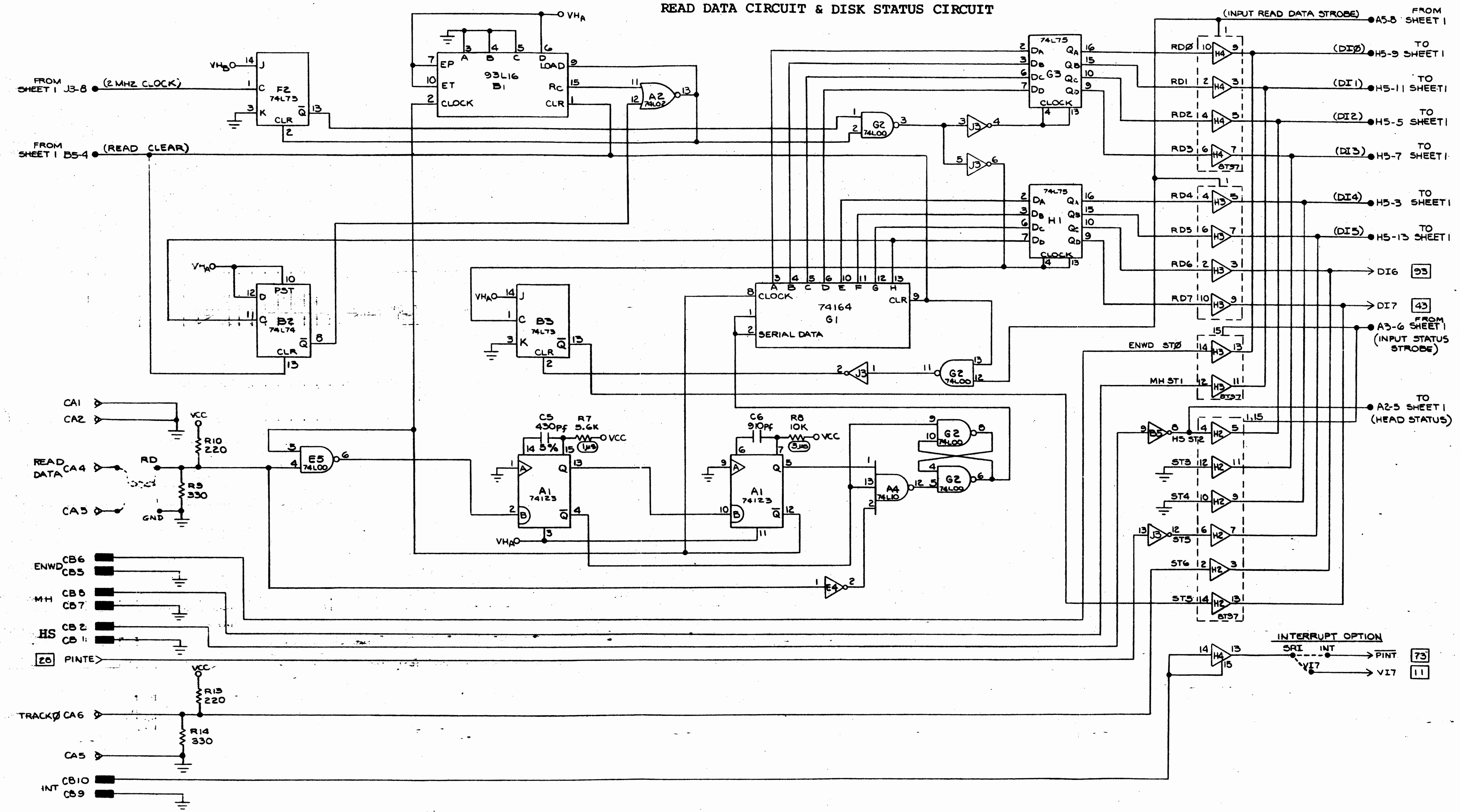
DRAFTSMAN: A.C. ROUCKUS

DATE: 6-17-75

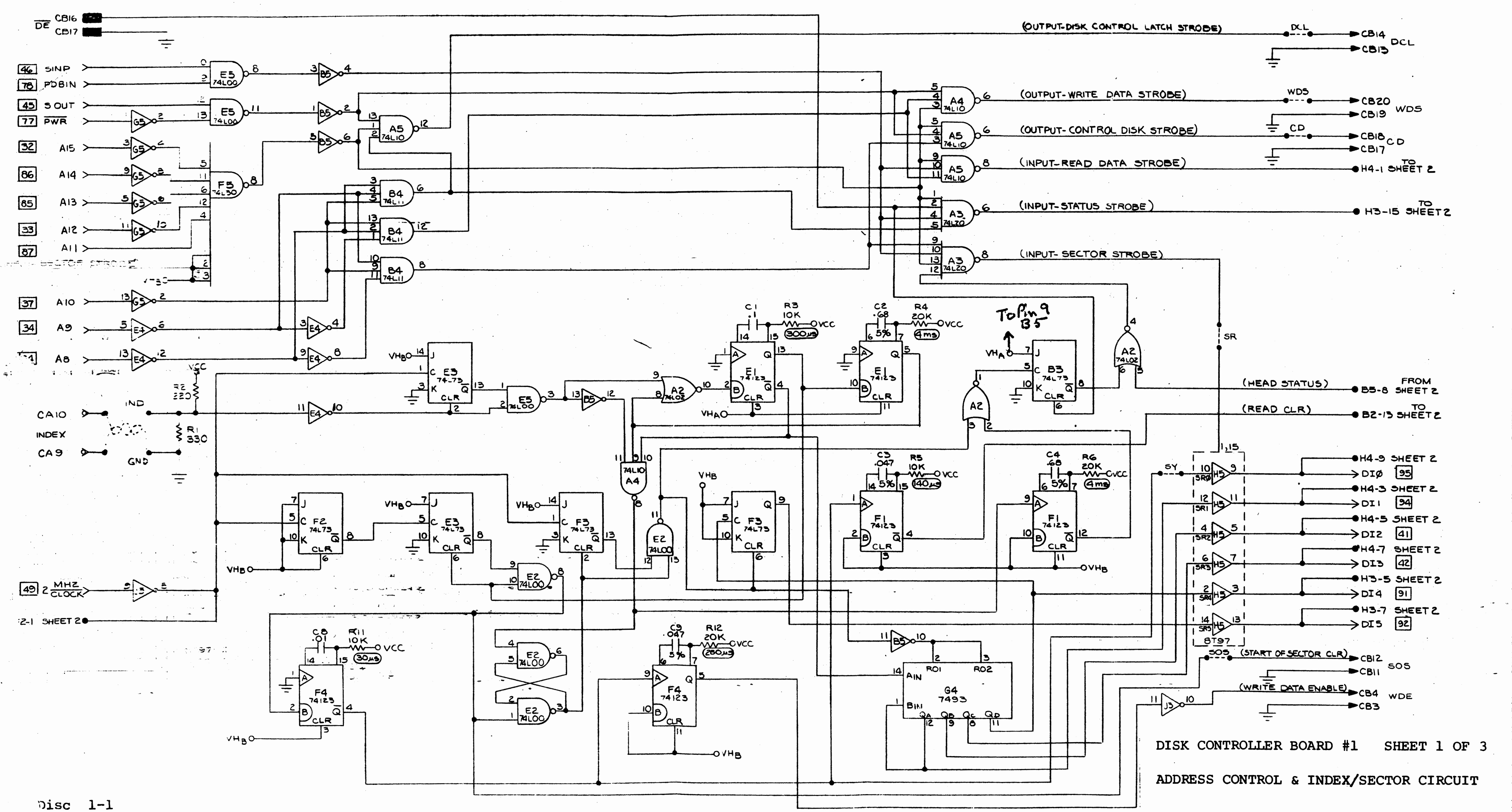
DRAWING NO. DD-101

DISK CONTROLLER BOARD #1 SHEET 2 OF 3

READ DATA CIRCUIT & DISK STATUS CIRCUIT



Disc 1-2



DISK CONTROLLER BOARD #1 SHEET 1 OF 3

ADDRESS CONTROL & INDEX/SECTOR CIRCUIT

Disc 1-1

FROM DISK