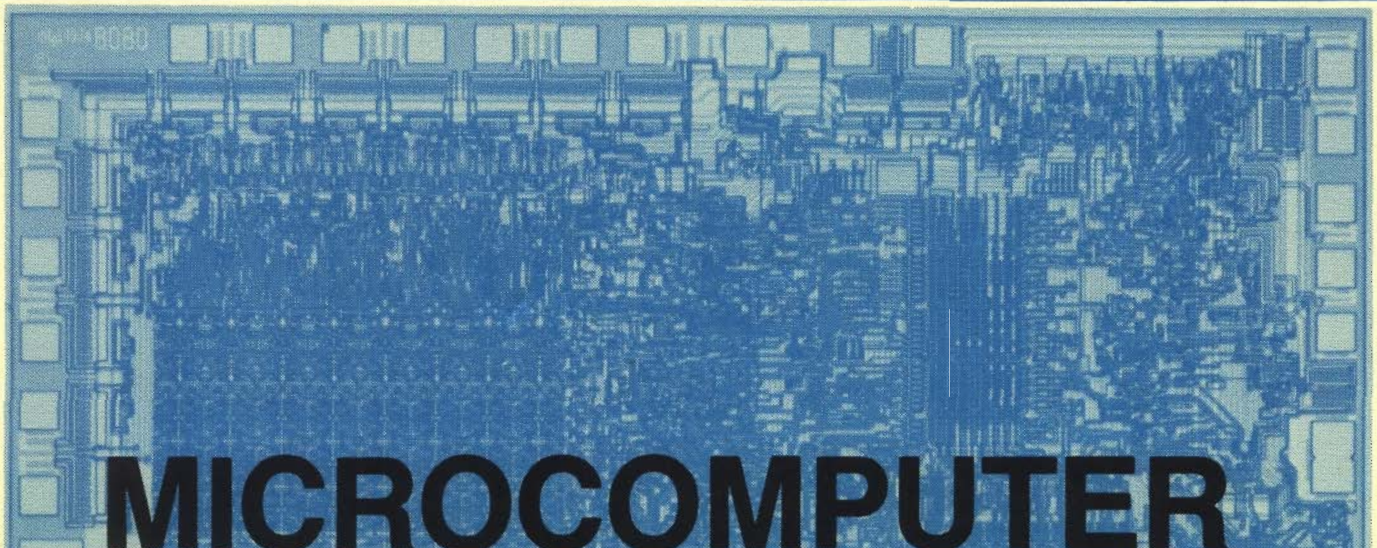
IMS ASSOCIATES, INC.

USER MANUAL

IMSAI® 8080



MICROCOMPUTER SYSTEM

USER MANUAL

IMSAI®

INTRODUCTION

This chapter contains the following sections:

1. Kit Unpacking Instructions
2. Construction Hints - general notes on how to build your kit.
3. Recommended Overall Order of Assembly (includes cross-reference to chapters where specific assembly instructions for the various submodules will be found).
4. Mainframe Assembly - assembly instructions for integrating Chassis with Power Supply, Mother board and Front Panel, and instructions for testing the Power Supply.
5. System Functional Test - how to check out your overall system.

KIT UNPACKING INSTRUCTIONS

1. Remove all packages from the outer box. For a standard IMSAI 8080 kit, these will consist of:

    a. Documentation Set (Manual plus two books)
    b. Cabinet Base Plate
    c. Table Top Cover (or Rackmount cover and Rackmount painted pieces)
    d. Two large inner boxes
    e. Two small inner boxes.

2. Largest inner box contains flat parts such as pc boards, small sheet metal parts, two plastic panels and a mailing tube containing the front panel mask and paper backing sheet (latter is deleted if an OEM machine has been ordered).

3. The next smaller size inner box contains plastic sacks of components. (There will be a plastic sack with a parts list corresponding to each pc board, plus sacks for the chassis and rackmount hardware and a sack containing the paper tape for the IMSAI Self-Contained System software.

4. One of the two small boxes contains the large components for the Power Supply (transformer, capacitors, etc.).

5. The second small box is either empty (serving as a spacer box for packaging purposes) or contains overflow from the sack parts box.

6. Unpack plastic sacks only when you are ready to begin assembly of that particular module. If any parts are missing, contact IMSAI Customer Service for immediate replacement.

7. Be careful in handling the painted sheet metal parts, the plastic parts and the film negative to avoid scratching. PC boards should not be stacked without protective material between to avoid destroying or shorting traces.

CONSTRUCTION HINTS

GENERAL

The IMSAI 8080 microcomputer is a complex piece of electronic equipment. This section covers a number of items, each of which must be followed to insure a working system at the completion of assembly. This entire section must be read completely before beginning assembly, and the builder must refer back to the notes in this section often enough to insure that no components are installed incorrectly. While each assembly step is easy to do correctly, there are many steps and it is also easy to do one or more incorrectly, and much more time will be spent solving a problem than would have been needed to prevent it.

There may be items about which you are not completely sure during assembly. Should this occur, DO NOT CONTINUE. Study the manual to see if you can resolve your question, or seek the help of someone more knowledgeable in digital electronics. If you feel your question is not resolved by further study or asking whoever is available to you, call IMSAI. This will enable you to do a better construction job, and it will enable us to revise the manual so that it will be of more assistance to you. We recognize that some builders will have had very little experience in assembling electronic kits, and it is our intention to continually revise the manual based on comments by users, so that even the most inexperienced builder can achieve the best unit available with a minimum of effort. No question is too simple to call about if you're not sure about it.

TOOLS AND WORKPLACE

It is next to impossible for even an experienced builder to produce a good machine unless proper tools and an adequate workspace are available. The kit does not require much space to work in, but enough table surface should be available for the piece being worked on, all the tools needed for that piece, and an orderly arrangement of the components which will be used in assembling that piece. The work area should be very well-lit, with no shadows. If the entire room is not well-lit by ceiling or window light, then at least two bright lamps should be used, preferably one on either side and slightly behind the chair to help eliminate shadows. You may want to protect the table surface with cardboard or newspaper.

The most important single item in assembly is the soldering iron. It is critical enough that a separate part of this section is devoted to it. Other tools which are absolutely necessary to do an adequate assembly job are screwdrivers to fit the screws used in the kit (both straight slot and phillips), a small pair of diagonal cutters (preferably a 4" pair, flush-cutting), small needle-nosed pliers, and a wire stripper. A 1/4" nut driver will make cabinet assembly very much easier, as the sheet-metal screws used are designed primarily to use a nut-driver. A voltmeter should be available for testing. Any inexpensive meter (VOM) with DC voltage scales between 5 and 30 volts should do. Do not attempt to assemble the kit until you have the tools necessary; damaged parts cannot be replaced under warranty.

SOLDERING

Almost every problem with an assembled kit is a soldering problem. If you have never soldered before, or if you have done some soldering but do not yet have facility in making good soldering joints both quickly and every time, practice before beginning assembly on the IMSAI 8080 boards. Obtain some extra #20 hook-up wire and solder locally and solder pieces together until you feel comfortably able to quickly make a good joint. The importance of good solder joints is just too great to convey adequately here; but don't be scared off, because once you get the hang of it, they're very easy to do.

Soldering Irons

There are a great many tools available with the name "soldering iron". Two thirds of these are not appropriate to small electronics assembly and if used are almost certain to damage both parts and boards. The problem with most of these are that they are too big and too hot. Note that most every soldering "gun" is in the too big, too hot class. Proper soldering irons are easily available at any local hobbyist electronics outlet, and they are not expensive. Use a 30-40 watt iron with a small tip, such as an Ungar 776 with a 7155 tip. If you wish to invest in a top-quality tool, a temperature-controlled tip model such as the Weller W-TCP with a small 700^{o}F tip is well worth the extra cost. Many irons are available with either unplated copper tips or plated tips. Though slightly more expensive, the plated tips last very much longer and give superior service.

Solder

Using the proper solder is as important as using the proper iron, and there are many solders to choose among. In normal electronics assembly, separate paste or liquid flux is not used. Rather, a solder with a "core" of rosin (or resin) base flux is used. This flux (contained in the hollow center of the solder) should be sufficient. Absolutely avoid any solders using an acid flux. (Or any cans of acid flux - unless a can of flux says"rosin" you may safely assume it is an acid flux. Acid fluxes are used for mechanical soldering where the surfaces are not as clean as those in electronic assembly. They are corrosive and will typically damage a printed circuit.

Also very important is the ratio of tin to lead used in the solder. Best to use is 63% tin, 37% lead, called 63/37 or eutectic. Much more common is 60/40, which is still a very good solder. Avoid using 50/50 or 40/60, even though they're a little cheaper. The higher-lead ratios solidify gradually, while the 63/37 solidifies almost instantaneously, making "cold solder joints" very much less likely.

Also important is the gauge (or diameter) of the solder. For fine electronics work a fine gauge should be used, such as #20 (from #19 to #22 is OK). Again, the correct solder is easy to obtain from any local hobbyist electronics outlet or TV repair shop. ERSIN Multicore or KESTER are two brands you can count on for good results. The solder included in the kit should be sufficient. If for some reason it is not, and you cannot obtain the proper solder locally, DO NOT USE any substitutes. More solder of the proper type can be obtained from IMSAI.

Soldering Technique

For a joint to solder correctly, enough heat must be applied so that both pieces of metal get hot enough to melt the solder. The tip of the iron should be applied so that it touches both the wire and the foil pad on the board. The end of the solder should then be touched to the junction of the iron, lead, and pad, so that a small amount melts and "wets" the joint (flows smoothly on both the lead and pad). As soon as the joint has wet, the iron can be removed, and the joint inspected immediately. Careful inspection of each joint is the key to successful soldering. While the solder is being applied, watch the joint carefully. You should be able to see the solder flow onto the two surfaces. It should flow around the lead, and if you see that the solder has flowed only on one side of the lead, the iron should be re-applied (while watching the joint) to heat the joint enough for the solder to flow. (The typical reason for solder to flow only half-way around a lead is that not enough heat was applied.) For the normal joint, only a small amount of solder is needed (approximately 1/8" of 20 gauge solder wire) for it to flow all the way around the lead. Also, for the normal joint, only 2 to 4 seconds of heat applied from the iron is necessary. More heat and solder will be needed for some joints with larger leads and holes or large foil areas, but if more heat or solder is needed on typical component leads (like IC's), it is an indication that something is not right.

Since nearly all the holes in IMSAI printed circuit boards are plated-through (the inside walls of the hole have a metal surface, connecting the pads on the opposite sides and providing greater area for solder to adhere to) some solder will typically wick through and be visible on the top side of the board. This is normal. If small drops of solder appear on the top side, it is an indication that too much solder is being applied, along with more than sufficient heat. These balls of solder can easily short to neighboring pins and must be avoided. If the correct amount of heat or less than the correct amount was used along with too much solder, the solder remains on the bottom of the board (the side the solder is always applied from) and forms a blob which can easily short to neighboring pads or traces. If one of the small gaps between foil pads or traces has been shorted with too much solder, it can often be unshorted by running the hot iron lightly down the shorted trace, re-melting the solder at the shorted point and pulling it away with the iron. Do not leave iron on traces or pads too long when soldering or fixing a short, as overheated traces easily come off the board. As a result, very special care must be exercised for any component removal operation.

The tip of the iron must be kept clean to work well. Most stores that carry irons also carry small sponges in holders designed for cleaning hot tips. The tip is simply wiped on the wet sponge quickly. A damp rag will serve as well though less convenient. The tip must be kept adequately tinned at all times to avoid an oxide coating forming. It should appear bright and shiny. A small amount of solder should be melted onto the tip each time it is cleaned unless a joint is to be made immediately. If a tip becomes oxidized, dipping it in a can of rosin flux is usually sufficient to enable solder to flow on it again. They may be cleaned of oxide by fine steel wool or other abrasive, but a plated tip should never be filed.

The tip of the iron should never have enough solder on it that it could drip off. If you find that solder tends to drip off the tip, you are undoubtedly using too much solder. A solder drip on a P.C. board is often extremely difficult to see, since it is the same color as the traces, and it is sure to short several traces and cause trouble or damage components when the board is operated. Inspect your boards very carefully for any such solder drips, shorts near soldered leads, incompletely soldered leads, and unsoldered leads. A 100% inspection of soldering should catch 99% of all problems before the board is even turned on. When soldering components with long leads (resistors, etc) we suggest clipping the leads after soldering so that lead clipping gives you an easy and positive way to check all the joints on those components. A completed unit will typically run when first turned on if the soldering was done correctly.

MOS IC HANDLING

Some of the chips in the kit are MOS type chips (such as the 8080A, 8111 and 8251). MOS chips are sensitive to static electricity and other large transient voltages. In order to prevent damaging these, some precautions should be followed. They all relate to avoiding the discharge of static through the pins on one of these chips.

Avoid working in a room with very low humidity. Wearing cotton fabric or other non-static forming fabrics will help. Air directly from a heater vent is typically extremely low in humidity and should be avoided in the work area. Keeping everything involved (chip, board, iron, tools, boxes, chip containers, work surfaces and you) at the same potential is required, and the biggest step in achieving this is continuous physical contact between them. For example, before removing a chip from a box and setting it on the table, the box should be set on the table, you should touch the table, and only then pick up the chip to place it on the table. Try to handle the chip from the ends rather than the pins as much as possible, and always touch the chip's container or surface which it is touching before picking up the chip. Also touch a surface or container before placing the chip back in it. Touch a PC board before inserting the chip. Touch the soldering iron to the work surface or to a small piece of metal foil on the work surface before touching it to the PC board for soldering. In general, make sure the chip is not the path for any static discharge. Save MOS IC insertion as the last steps in assembly to avoid unnecessary exposure.

POLARITY

Many electronic components will not work if they are connected backwards. Any component which it is important to insert one way only will have a mark of some sort to indicate which way is which. The board where they go will have some sort of corresponding mark at each place, or an indication that all such components go the same way as a marked "typical" one.

I.C.'s

All I.C.'s must be inserted with Pin 1 in the correct location to avoid damaging the I.C. Pin 1 is indicated on the chips by several different marks. The most common is a rounded or square notch in the center of the end near Pin 1. Another common one is a slightly depressed or raised dot in the corner of the chip next to Pin 1. One or both of these will always be present to indicate Pin 1. Sometimes there are other circular markings on the centerline of the chip, usually towards one or both ends; these should be ignored. Often there is some kind of Pin 1 mark on the bottom of the chip also. (Note: Many I.C.'s have a code for date of manufacture which is a 4 digit code. e.g. 7425 would indicate manufacture in the 25th week of 1974. Do not confuse these with the device number. The code will be alone, the device number will have manufacturer-dependent suffixes and prefixes. e.g., SN7404N is a 7404 type chip. On the PC board, some Pin 1 indication will be found, such as a square pad, a dot, an arrow showing Pin 1 direction with the note "typical" (indicating all chips on the board face the same way), or similar mark.

The board or the chip is very likely to be damaged if there is a need to unsolder a chip that was soldered in with Pin 1 in the wrong direction. Unless you are completely sure you are capable of unsoldering an integrated circuit without damage to the circuit or the board, you should send the board back to the factory to have the work done for you. Remember that on these boards with plated-through holes, pins are not only soldered on the top where you see the visible bead of solder, but is soldered inside the hole which makes it much more difficult to remove.

Diodes

Diodes will typically have a band around the body, next to the cathode end. This corresponds to the bar on the typical diode symbol. The same is true for Zener diodes. A diode symbol should be found on the board or assembly diagram to indicate the proper mounting direction.

Capacitors

Some capacitors have a plus and minus lead; among them the tantalum and power supply electrolytic capacitors. Some mark on the body of the capacitor will indicate the plus lead, typically a + sign near it. There will be a mark (typically a + sign) on the board or assembly diagram to indicate the proper direction to mount the capacitors. A capacitor of this type is usually destroyed very quickly if power is applied to it in the reverse direction, so check your assembly carefully.

Transistors

Most transistors have a flat side or a small tab to indicate the lead orientation. If this indication is oriented according to the assembly diagram the leads should fit in the holes with little bending and no crossing.

MOUNTING COMPONENTS

Integrated Circuit Chips (IC's)

Some of the chips come in a little plastic rectangle with an open bottom and top. These can be used as inserters by setting the carrier with the chips on a piece of felt or similar material on a table top and pushing lightly with a pencil eraser or small object that will fit in the top of the carrier, until the chip has slid down with the leads resting against the table. Now, because of the material, the leads will be sticking out beyond the carrier a little bit. If you then pick up the carrier and the chip and set it on the board, you can line up the little protruding tips of the IC's ends into the holes into which they are supposed to go, and while you are holding the carrier, push the chip the rest of the way into the board again with a pencil eraser or with an object that will fit inside of the carrier.

For the chips that do not come in a carrier, after you insert the ones that did come in a carrier, you could use those carriers to insert the others also, by turning the carrier upside down and setting one of the other chips on the carrier and pushing it into the carrier and then just continuing the same process described above, to insert it in its location.

For chips with no such inserter aid available, the pins should be bent inwards far enough to line up with the holes in the board. Bend the pins on each side equally. The whole row of pins on one side can be bent in uniformly if they are all pressed against a flat surface to bend them. After putting the chip in the board, two diagonally opposite pins can be bent slightly to hold the I.C. in the board while soldering.

Take special care on each and every chip to observe the following points:

1. That Pin one is in the correct direction. Refer to marking on the board or assembly instructions to determine which direction Pin one belongs.

2. After inserting the chip and before soldering, check that every pin went through the hole properly. Sometimes a pin will catch on edge of a hole and bend under the chip instead of going through. Care should be taken to avoid this happening and to check before soldering to make sure it has not happened.

After inserting one or two chips, get a feel for how much pressure is needed to push it out of the carrier. Any chips that seem to take more pressure indicate that perhaps one or more pins are not lined up with the holes properly. Most chips after insertion, will stay in the board securely due to the fact that the leads are normally bent outward somewhat and will hold the chip by pushing outward against the holes. Some chips, however, will be loose after inserted. Extra care should be taken to see that these are properly against the board when they are soldered. The board can either be set flat against the table or other surface that will hold the chips against the board or two diagonally opposite ends may be bent slightly to prevent the chip from dropping out.

Power Regulators

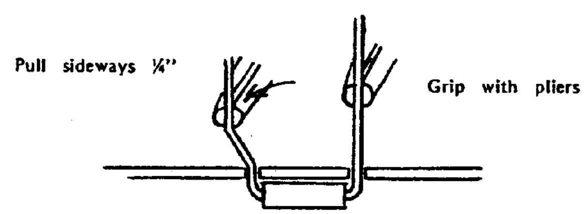
The 7805 regulators for the +5 volts are supplied with a heat sink and mounting hardware. The three leads must be bent down at the proper lengths to match the solder pads, and this should be done with the needle-nose pliers. The lead should come straight out and bend sharply down, rather than slope gradually towards the hole. After the leads are bent, the regulator can be fastened to the board along with the heat sink, using the short 6-32 screw down from the top, with the lockwasher and nut on the back. The regulator should be held to prevent turning while the nut is tightened firmly. The nut should be tight enough to insure good heat conductivity between the regulator and heat sink and board. Heat sink grease may be used if desired.

Discrete Components

Resistors and diodes can be installed most neatly using a lead bender to bend the leads consistantly. Most pads for this sort of component are .5" apart.

Disc ceramic capacitors often have the dipped insulation extending down the leads a short distance, preventing these from being pulled down all the way to the board. This insulation may be broken off by squeezing it in the pliers. Take it off until the bare wire comes up to the level of the bottom of the capacitor.

All discrete components should be held in their desired final position while being soldered. Normally this means holding them against the board by putting a slight bend in the lead behind the board so the component cannot lift from the board. (See the sketch for a way of bending the leads we find works better than simply finger-bending them slightly.) Components not held in place look sloppy and it is much harder to move them once they are soldered. In some cases, a little extra lead is needed, such as to lay the disc capacitors down on top of the chips on the front panel board. In these cases the solution is again to hold them in their final position during the soldering operation. This insures that the leads are left the proper length.



RECOMMENDED ORDER OF ASSEMBLY

Step	Description	Described In
1	Assemble MPU and RAM boards. Check carefully.	MPU Chapter RAM Chapter
2	Assemble CP-A including switches and flat cable. Check carefully.	CP-A Chapter
3	Assemble electronic components on Power Supply. Check carefully.	PS-C Chapter
4	Assemble Mother board(s). Check carefully	Mother Board Chapter
5	Assemble Chassis sheet metal: a. Install required number of card guides on card frames. b. Install fan (if supplied) on back frame. Install line cord through grommet. c. Bolt together sheet metal parts. Install rubber feet.	Cabinet Assembly Chapter
6	Install Power Supply Board in chassis. a. Bolt board in place. b. Bolt transformer in place. c. Cut wires to length and crimp on (or solder on) lugs. d. Connect up Power Supply except for wires to Mother board(s).	Mainframe Assembly Section
7	(Connect Mother boards together and) install Mother Board(s) in chassis.	Mainframe Assembly Section
8	Connect wires to Mother board. Check carefully.	Mainframe Assembly Section
9	Prepare front plastic panel assembly.	CP-A Chapter
10	Plug CP-A board into Mother board. Connect wires to CP-A board. Install front panel assembly. Hold CP-A DIP cable out of way.	Mainframe Assembly
11	Check complete assembly carefully before applying power. Plug in machine and turn on. Test Power Supply voltages.	General Assembly and Test Instructions
12	Plug in MPU board and RAM board(s) and test system.	General Assembly and Test Instructions
13	Assemble other individual boards. Check carefully.	(Individual board chapters)
14	Install individual boards.	
15	Install required cables. Install Cable Clamp.	Cabinet Assembly Section
16	Install Switch escutcheon and cover and/or Rackmount parts.	Cabinet Assembly Section

MAINFRAME ASSEMBLY

Assembly of the mainframe consists of the following steps:

- ·Power supply installation
- ·Mother board installation
- ·Connection between power supply and mother board
- ·Installation of CP-A panel.
- ·Connection of power supply and front panel

POWER SUPPLY INSTALLATION

Remove #8 hardware from transformer on Power Supply p.c. board. Take care to not let the transformer damage the p.c. board. Put the five #8 screws in the cabinet bottom and secure with the 8-32 threaded spacers. Install the four ¼"-20 nuts and spacers for the transformer similarly. Carefully lower the Power Supply Assembly onto the mounting screws so all the screws extend through the board. Fasten with washers and nuts. See Figure 1. Complete the power supply by attaching the capacitor brace plate to the bases of the large capacitors with the adhesive backed foam tape on one side of the brace plate.

MOTHER BOARD INSTALLATION

Attach the Mother board to the cabinet base with the hardware supplied with the Mother board as shown in Figure 2. The front 100-pin connector should be located in front of the sheet metal front frame to accomodate the CP-A assembly.

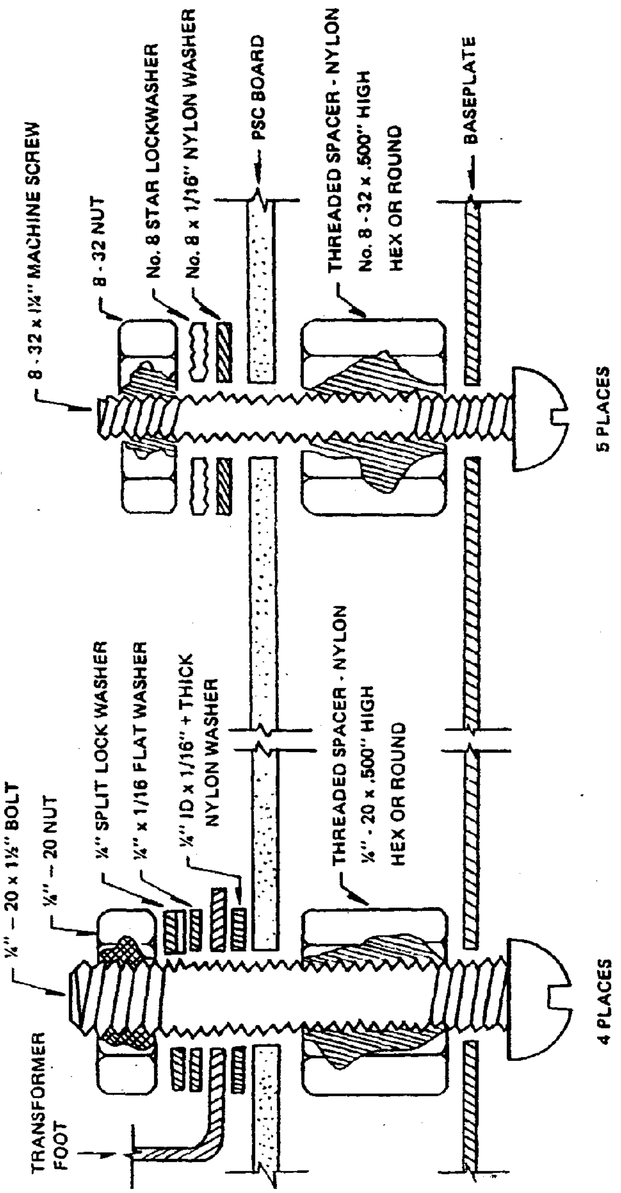
CONNECTION BETWEEN POWER SUPPLY AND MOTHER BOARD

See the wiring drawing in the Power Supply chapter. Connect the following wires between the Power Supply and the system:

a) 1 or 2 #18 gauge wire from holes at edge of -16 volt plane to -16 volt trace on Mother board.

b) 1 or 2 #18 gauge wire from holes at edge of +16 volt plane to +16 volt trace on Mother board.

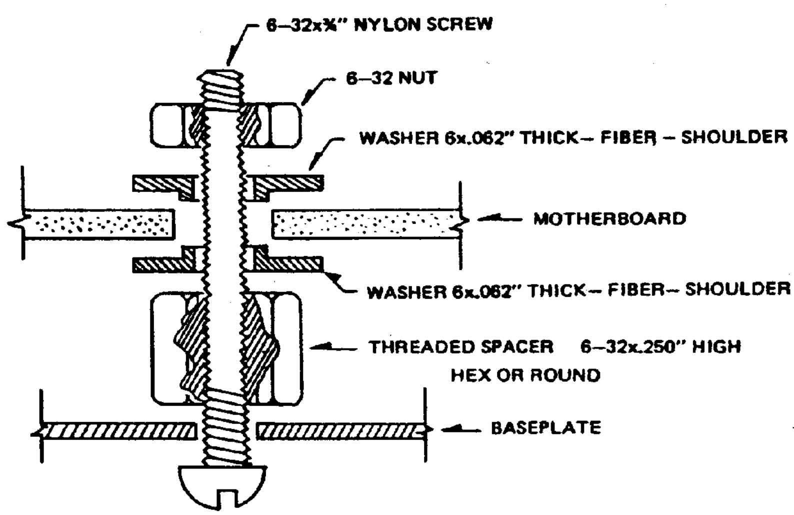
c) 2 or 3 #14 or #12 gauge wire from +8 volt plane to +8 volt bus on Mother board.



PS-C BOARD MOUNTING SYSTEMS

MAINFRAME ASSEMBLY
FIGURE 1

MOTHERBOARD MOUNTING SYSTEM



4 PER EXP–4

8 PER EXP–6

24 PER EXP–22

MAINFRAME ASSEMBLY
FIGURE 2

d) 2 or 3 #14 or #12 gauge wires from ground plane to ground bus on Mother board.

e) 2 #18 gauge wires from External Switch pads to power switch on CP-A or on back panel.

f) 2 wires (#18 or #20 gauge) from switched AC pads to fan (if fan installed) install insulated tubing over fan terminals.

g) 3 wires from power cord to terminals W, G, and B on PS-C. Make sure the power cord wire colors match the label on the panel

INSTALLATION OF CP-A PANEL AND CONNECTION TO POWER SUPPLY

Plug the completed CP-A panel into the front 100 pin connector on the Mother board. Install the eight Allen head screws into the PEM nuts on the sheet metal front frame. Solder the two #18 gauge wires from the External Switch pads on the Power Supply assembly to the power switch pads on the CP-A panel. Provide as much clearance as possible between the connections on the CP-A board and the sheet metal front frame. Be careful not to damage the acrylic panels with the soldering iron.

CHECK OUT OF POWER SUPPLY

Before plugging in circuit boards except the CP-A board, the unit should be plugged into the AC power supply and the power supply turned on by depressing the front panel rocker switch. The voltages at the outputs should then be measured (any DC volt meter with a full scale voltage of 20 to 50 volts will do) and the voltages should read approximately 18 volts on the +18 and -18 volt outputs, and 10 volts on the +8 volt output. If the voltage does not come to these values, a check should be made that the positive and negative terminals of the capacitors are connected properly and the diodes are mounted properly. If there is a problem with any of these items a wiring error has probably been made and the wiring should be re-checked carefully. If the wiring is checked and no error is found, assistance should be sought from a person know-ledgeable in electronics or from the factory.

When the voltage of the capacitors has been checked out to be satisfactory, the unit may be turned off. A 10 minute wait will permit the capacitors to discharge. While there is considerable energy stored in the power supply filter capacitors when they are fully charged, the voltage levels are not high enough to present a danger. Some care should be taken, however, not to discharge the capacitors by shorting them with a tool or other metallic object.

With the Power Supply checked out and operating properly, the rest of the system is ready to be tested. The MPU board should be inserted in the slot behind the front panel with the flat cable inserted into the socket in the upper right hand corner of the MPU board before the board is fully seated.

The memory board should then be inserted in the third slot. While it is not necessary that the first memory board be addressed beginning at position 0, it is normally expected and the rest of this section will assume that the memory board jumpers were wired according to the directions in the User Guide section of the RAM-4A board for addressing the board at 0.

The slots in the Mother board are not unique and if a larger version (e.g., 22 slot) was ordered with more edge connectors, the boards need not be plugged into the second and third slot as directed but may be plugged into any slots.

SYSTEM FUNCTIONAL TEST

When the boards are installed, the machine is ready to test. Turn the power on with the front panel rocker switch and depress the RUN/STOP switch momentarily to STOP position and release. The WAIT light should be on and the RUN and HOLD lights should be off, with the other lights in various states at this time. Raise the RESET switch momentarily to the RESET position and release. All the lights on the bottom row in the ADDRESS BUS section should be indicating that the program counter is set to location 0. The WAIT light should still be on with the RUN and HOLD lights off. The DATA BUS lights may show various random bits on and the STATUS byte should have three lights on: MEMR, M1, and $\overline{\text{WO}}$. With all 16 ADDRESS switches in the down or 0 position, the EXAMINE/EXAMINE NEXT switch should be raised momentarily to the EXAMINE position and released. Check that the lights after this operation are exactly the same as described for after the RESET switch was operated.

The machine is now ready to enter a small test program. For complete description of program operation in computers, read An Introduction To Microputers. For the initial machine test, the following program should be entered:

TEST PROGRAM 1

ADDRESS	HEX	BINARY	OCTAL	
0	DB	1101 1011	333	INPUT
1	FF	1111 1111	377	ADDRESS
2	D3	1101 0011	323	OUTPUT
3	FF	1111 1111	377	ADDRESS
4	C3	1100 0011	303	JUMP
5	00	0000 0000	000	LOW ADDRESS
6	00	0000 0000	000	HIGH ADDRESS

TEST PROGRAM 2

ADDRESS	HEX	BINARY	OCTAL	
0	DB	1101 1011	333	INPUT
1	FF	1111 1111	377	ADDRESS
2	2F	0010 1111	057	COMPLEMENT DATA
3	D3	1101 0011	323	OUTPUT
4	FF	1111 1111	377	ADDRESS
5	C3	1100 0011	303	JUMP
6	00	0000 0000	000	LOW ADDRESS
7	00	0000 0000	000	HIGH ADDRESS

The address is now at 0 as indicated by the lights labelled ADDRESS BUS. Into position 0 we wish to put an input instruction.

The bit pattern for the input instruction must be set in the center group of switches labelled ADDRESS-DATA. Switches 7, 6, 4, 3, 1 and 0 should be placed in the up position. Compare these switch positions with the binary representation of the input instruction listed on the first line of test program 1. We wish now to deposit this bit pattern in memory position 0. Raise the DEPOSIT/DEPOSIT NEXT switch up momentarily to the DEPOSIT position and release. The address bus should still show 0 (no lights lit) and the data bus should now show the bit pattern set in the switches (bits 7, 6, 4, 3, 1 and 0 lit and bits 5 and 2 off).

Next, the bit pattern for the address of the input port should be written in position 1. This can be done by setting all eight ADDRESS-DATA switches up, corresponding with the address listed on line 2 of Test Program One, and the DEPOSIT/DEPOSIT NEXT switch depressed momentarily to the DEPOSIT NEXT position and released.

Now the address bus light should show position 1 (address bus light 0 on and all other address bus lights off). The data bus should show all eight lights lit corresponding to the bit pattern written here. Similarly, the next five lines of Test Program One should be set into the ADDRESS-DATA switches and deposited by operating the DEPOSIT NEXT switch, each time checking to make sure that the data bus lights correspond with the settings of the ADDRESS-DATA switches and that the address is correct indicating that no steps have been skipped or done twice.

When the last byte has been deposited in address position 6, then all 16 address switches should be returned to the 0 position (down) and the EXAMINE switch operated. This should reset the address bus lights to 0, and display the contents of the bottom word in memory on the data bus lights. (This should still be the binary pattern listed in line 1 of the Test Program). The EXAMINE NEXT switch can then be operated and the address bus lights should indicate address 1 (bit 0 on and all other bits off). The Data Bus should show the contents now of memory location one which should correspond to the second line of Test Program One listing (all ones).

The EXAMINE NEXT switch can be repeatedly operated, each time checking that the data located in the consecutive memory location corresponds exactly to the listing for Test Program One.

The EXAMINE switch can again be raised momentarily with the address switches all down, to return the machine to position 0, once it has been determined that all lines listed in Test Program One are stored correctly in the memory.

Now we can single-step through this program and watch the operation of the machine. With the machine sitting at 0 with the correct instruction on the data bus, and the MEMR, M1 and $\overline{\text{WO}}$ lights lit in the status byte, the processor is reading the first instruction out of memory into the processor for execution. If the SINGLE-STEP switch is either depressed or raised once, it will permit the processor to complete its cycle and begin the next cycle. The address bus lights will show position 1, the data bus will show all ones corresponding to the bit pattern in the Test Program, and the status byte will show MEMORY READ and $\overline{\text{WO}}$. The lack of an MI light in a status byte indicates that the processor is no longer fetching an instruction to execute, but rather this cycle it is fetching the address for the instruction which it has already stored internally.

If the SINGLE-STEP switch is operated once again, the address bus lights will all be lit. The status byte will show INP and $\overline{\text{WO}}$ and the data bus will at first show no lights on. If one or more switches in the left hand group of eight switches is now raised or lowered, the corresponding light on the data bus indicators will turn on or off. The processor is now executing the first instruction which was an input data from address FF hex (377 octal) which is the address for the programmed input port on the front panel. By means of this instruction with this address the processor is able to read the position of the eight switches in the left hand group. (The address being read is indicated by the lights in the address bus and, on input or output instructions, the address appears in both groups of eight lights on the address bus. Thus, for this address, all the lights in the address bus are lit.)

The switches in the left hand group should be left in the position of some up and some down to provide a recognizable pattern before continuing. With the pattern left in the left hand group of switches, the single step switch can be operated once more permitting the processor to complete the execution of the input instruction, and begin the next cycle. Having completed the input instruction, the next cycle will be a fetch cycle during which the processor reads the next instruction to be executed, which it will find in memory address position 2. The address bus lights should now show positon 2 (bit 1 on and all others off), and the data bus should indicate the bit pattern listed on line 3 of Test Program 1 for address position 2. This is the output instruction.

The Status Byte will again have MEMR,M1, and $\overline{WO}$ lights lit and the others off. When the single step switch is operated once again, the processor is permitted to complete the cycle during which it reads in the output instruction and begin the next cycle during which it will read the address of the output device. Since it is reading this address from the next memory position, (memory position 3), the address bus will have bits one and 0 on and the others off. The Data Bus will have all lights on indicating the bit pattern we stored in memory position 3. The status bit will show MEMORY READ and WRITE OUT lights on, and the M1 light is off at this time, indicating that this is not an instruction fetch cycle, but rather it is one of the cycles required to execute the last instruction fetched-in this case, reading the address to which the data will be output. When the SINGLE STEP switch is operated once again, the processor is permitted to complete the cycle of reading the output address in and begin the next cycle which is the output operation.

The output operation looks similar to the input in that the address of the output device appears in both the upper and lower half of the Address Bus, (again in this case lighting all the lights), and the data being output appears in the Data Bus, which should show the pattern previously set in the left hand group of switches. Since the data is being output from the accumulator in the processor where it was previously stored in the input instruction, it will not be affected by moving the switches in the left hand group at this time. The Status Byte shows the MEMR light off at this time and shows the out light on indicating that the processor is executing an output instruction. The $\overline{WO}$ light is off indicating that the processor's WRITE strobe is active. If the SINGLE STEP switch is operated once more, it will permit the processor to complete the WRITE operation and begin the next cycle. At this time, the PROGRAMMED OUTPUT lights at the top left of the panel, should be lit according to the complement of the pattern that was set in the switches. That is, for each switch that was set in the up position, the light will be out, and each switch that was set in the down position, the corresponding light will be on.

Since the processor has completed the output instruction the next cycle is used to fetch the next instruction to be executed, which it will read from memory position 4. In memory position 4 we had stored the jump instruction which should now appear on the lights on the data bus indicators. As the SINGLE STEP switch is operated again, permitting the processor to complete the fetch of the jump instruction, and start the next cycle of executing that jump instruction, we find that the processor is reading the low half of the address from memory position 5. The status byte shows the MEMR and $\overline{\text{WO}}$ lights lit, and the M1 light is off at this time.

If the SINGLE STEP switch is operated once again, it will be seen that the processor is reading the high address byte previously stored in memory location 6.

The next operation of the SINGLE STEP switch permits the processor to complete the execution of that jump instruction, which is instructing the processor to take its next instruction to be executed not from memory position 7 but from memory position 0 as was stored in the two bytes following the jump instruction.

The Address Bus lights should now be all off indicating that the processor is indeed fetching the next instruction from memory location 0. The Data Bus should show the pattern that we wrote in memory position 0 as the input instruction. We have now completed one cycle of the loop in Test Program 1. Further operations of the SINGLE STEP switch will let the processor step through the execution of the loop additional times and each time through the loop it is possible to set a different pattern in the left hand group of switches to be read in and later to be written out to the PROGRAMMED OUTPUT light
The RUN/STOP switch can be momentarily raised to the RUN position and released. This will permit the processor to run at the full clock speed which will result in the loop being executed roughly 50,000 times every second. Thus, as any of the switches in the left hand group of eight are moved while the program is running, the machine reads the new position essentially instantly and displays it in the PROGRAMMED OUTPUT port above.

It may have been puzzling that the lights in the PROGRAMME OUTPUT port seem to indicate the opposite of what might have been expected when a bit was read in as a 1 and outpu to the PROGRAMMED OUTPUT port. This will serve as an example of the way logic design has been affected by the appearance of large scale integration and micro-processors. While it would have been entirely possible and easy to provide a circuit modification such that when the data was put out as a 1 the light would be lit rather that turned off, such as addition to the circuit would have cost you more that the cost for byte of memory. The same function as the added circuit can be accomplished by adding one instruction to the loop which complements the data, that is, changes all ones to 0's and all 0's to 1's. Test Program 2 is exactly the same as Test Program 1 with the addition of one instruction between the input instruction and the output instruction, which will complement the data read in from the switches before it is output. If the machine is stopped and reset, Test Program 2 may be entered exactly the same way as Test Program 1 was and checked and then run through one or more cycles with the operation of the machine and to double-check that the program truly has been entered correctly. Then the RUN switch may be actuated to permit the loop to run at high speed.

With this change in the program, the PROGRAMMED OUTPUT port will show a light lit when the switch is positioned up to enter a 1 bit. Not only is this a less expensive way to achieve the function of causing the lights to turn on when the bit is entered as a 1, but it is a much more versatile solution since the operator can change his mind at a later date and either remove the complement instruction or change it to yet another instruction for a different result.

When single stepping through Test Program 2, the compliment data instruction is seen to use up only one cycle of the processor. We are able to see it being fetched to be executed, and when the SINGLE STEP switch is operated again, we are immediately fetching the next instruction. This will be true of any instructions which operate only on data which is already stored within the processor. Additional cycles are only necessary if additional information must be read in or out of the program processor itself.

After either loop is running, the RUN/STOP switch may be depressed to STOP at any time and the operation processor will stop during the fetch of the next instruction. Due to the speed at which the processor operates, it is impossible to tell beforehand at what point in the loop the processor will be at the exact instant that the RUN/STOP switch is moved to STOP, so that the processor will stop at different places in the loop for different times when the switch is actuated.

The switch may be raised to the RUN position starting at any point in the loop and the processor will continue to run at high speed beginning at the point. The flip-flop set by the RUN/STOP switch simply instructs the processor to wait at each cycle for a pulse which is generated by the SINGLE STEP switch to be received before executing the next cycle, and apart from waiting for this pulse, the processor executes exactly the same whether it is in the single run mode or stop mode.

The definition of a computer involves both the ability to execute in sequence of instructions which is stored inside the machine, also the ability to make a decision between on the value of data and as a result of that decision, choose between alternate possible paths of program step sequences to execute. Test programs 1 and 2 involve only the execution of a sequence of stored program steps and do not involve any decisions. Program 3 will illustrate the use of decisions in a computer program and should provide some interesting entertainment as well. It is a game program using the INPUT switches and the PROGRAMMED OUTPUT lights on the IMSAI 8080 front panel.

A pattern of lights in the PROGRAMMED OUTPUT ports is moved to the left one bit at a time, and the left hand bit which is "pushed off" the end of the programmed I/O register re-appears at the right end of the register. The rate at which the bit pattern is shifted to the left can be chosen by the binary number set in the front panel switches when the program is first started or when the machine is reset to start again. When a higher binary number is entered in these switches and program restarted, the bit pattern will shift to the left at a higher rate of speed. Initially, switches should be set for 2, that is all switches down except PROGRAMMED INPUT switch bit 1 on, in order that the bit pattern will be shifted slowly enough to easily see what the game program is doing. Once the program has been started, the rate at which the bit pattern is shifted to the left is not affected by any further movement of the front panel switches. From this time on, any time any one of the eight switches in the PROGRAMMED I/O group is <u>changed</u>, then the bit in the PROGRAMMED OUTPUT port which is directly above that switch at the moment is was moved, will change. If it was off before, it will turn on; and if it was on before it will turn off. The direction of travel of the switch is not significant--only that its position was changed. After a switch change is detected, and the light above it turned on or off as appropriate, no further switch movements will affect the condition of any of the lights until the next shift to the left has occurred. This was done to give the switches time to stop bouncing and stay closed as the processor in this machine is quite fast enough to see the slight bouncing of the switch contact when it initially closes.

By waiting for the next data shift before recognizing any more switch changes, we are prevented from falsely interpreting a bouncing contact as a switch which was repeatedly opened and closed. The object of the game can be either to turn out all the lights in the shifting bit pattern by moving a switch when the bits are passing directly over it, or alternately to turn on all the bits in the shifting bit pattern by moving a switch when a bit which is off is directly over it. Any time the shifting bit pattern is all 0's or all 1's, no movement will be seen in the PROGRAMMED OUTPUT port but by moving any switch, one of the lights will be changed so that the motion is again apparent.

Players can compete for the shortest time to go from all 0's to all 1's, or the other way - from all 1's to all 0's. When the game has been mastered at one rotation speed, the switches can be set for a higher binary number and the system reset to cause the processor to go back to memory location 0 and begin execution of the program again, and a new switch setting will be read to result in a higher rate of rotation, which makes it harder to move a switch at the exact instant the bit desired to be changed is directly above it. If there were only a single light on, circulating across the output port, and the player, (in attempting to turn it off by moving the switch when the bit was directly over that switch) was too slow, then the bit will have shifted away so that it is now over the next switch to the left, not only will that bit not be turned off, but the bit behind will be turned on so that now there are two bits circulating across the register and the player is further away from achieving all bits turned off.

Knowledge of some of the internal structure of the 8080 processor will be necessary to understand the game program. The Intel data book contains complete information and functional specifications on the internal structure of the 8080 processor, but only the basic aspects of the structure need be known to understand the program operation.

Figure 1 shows the structural blocks in the processor which are important to the programmer. Central to the processor's operation is the register named the ACCUMULATOR. This register and all the others is like one eight bit position in memory or a small "blackboard" with room for only eight bits of either 1's or 0's to be written. When the input instruction was executed during programs 1 and 2, the pattern from the switches on the front panel was read into the ACCUMULATOR register, and when the OUTPUT instruction was given it was again the contents of the ACCUMULATOR which was output to the PROGRAMMED OUTPUT port on the front panel. All arithmetic is done in the ACCUMULATOR and, except for special instructions, (to permit other registers to be read to or from memory) all programmed input/output from either memory or input/out interfaces goes to and from the ACCUMULATOR. The INSTRUCTION register is another "blackboard" with room to store the address where it last read a program byte from memory so that when it finished the execution of that step, it can increment that address by one and use it to determine where to get the next instruction.

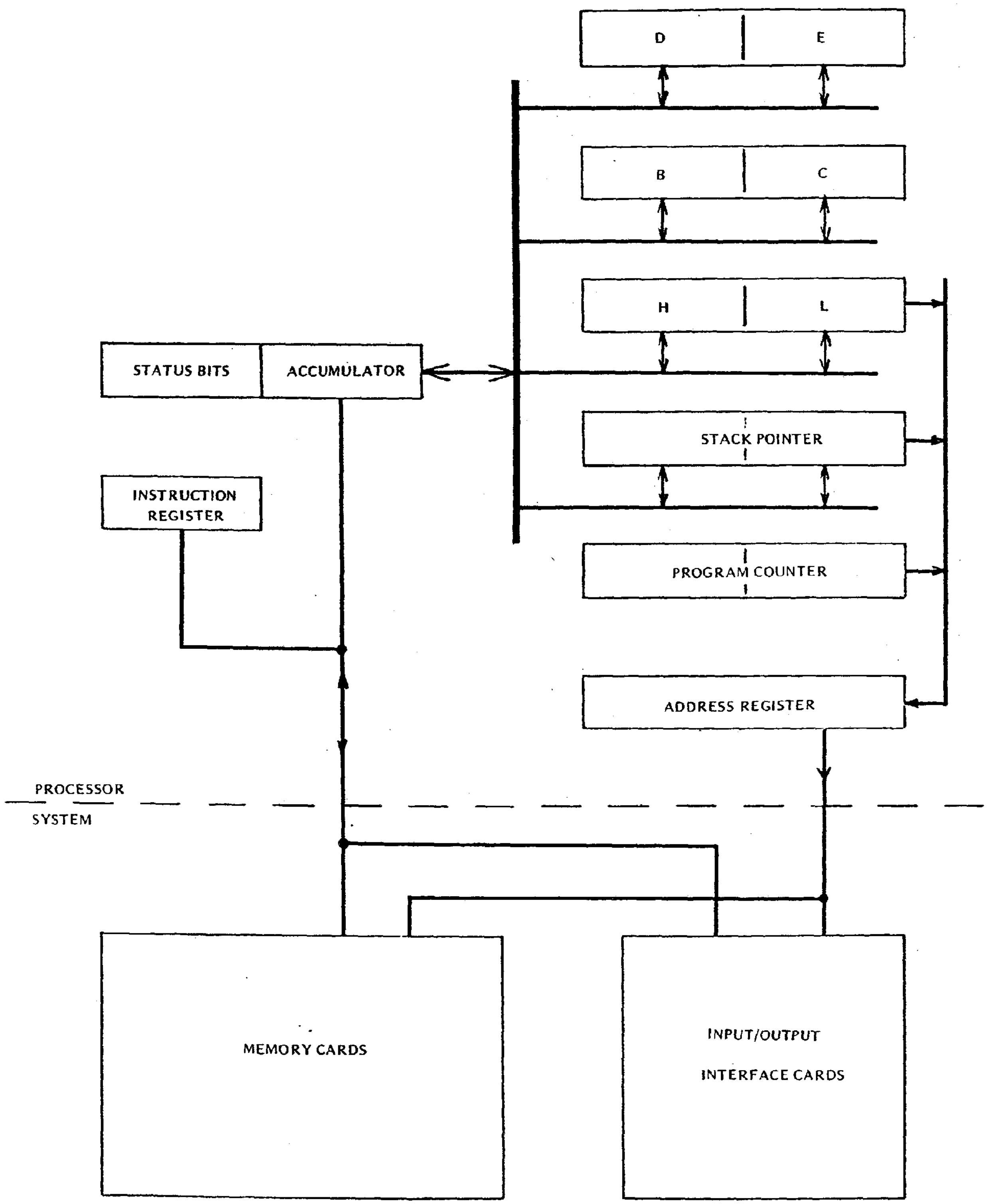
The STATUS BITS are 5 bits that are set to 1 or 0, according to the results of the last data operation performed in the ACCUMULATOR. One of the STATUS BITS or condition flags is the Z bit (zero bit) which is turned on when the last operation in the ACCUMULATOR resulted in the ACCUMULATOR being left all 0's. Otherwise, this bit is turned off. The second condition flag is the sign bit. If the most significant bit of the result of the last operation in the ACCUMULATOR has the value 1, this flag is set to 1, otherwise it is reset to 0. Three other condition flags are the sign parity and the auxilary carry, and their functions are described in the Intel Data Book on page 4-2. The fifth condition flag is a carry flag which is turned on if the last arithmetic operation produced an overflow. An overflow is produced, for example, when two numbers are added together and their sum is too large to be contained in the register into which it is put. For instance, if the ACCUMULATOR contained eight 1's and another number was added which contained the value 6, the correct answer would be the combination of the value 5 and a bit turned on in the 9th position. Since the ACCUMULATOR has only eight positions, the carry bit would be turned on.

Some of the STATUS BITS are affected by the operations in other registers than the ACCUMULATOR. For instance the carry bit is affected by additions made in the H and L registers by using the double add instructions. Use is made of this in the game program. There are five other registers in the processor, each of which is 16 bits long, and some of which are divided in half so that operations may be done with only 1/2 at a time. The ADDRESS REGISTER is a 16 bit register over which the programmer has no control. It is simply used to output either the memory address or the input/output address necessary to execute the next cycle. The other four 16 bit registers can all be used by the programmer. There are many instructions in the 8080A processor's instruction set whose function is to move data from any register to any other register, to permit arithmetic operations between a register and the ACCUMULATOR (with the result always being left in the ACCUMULATOR), and some special instructions to permit direct transfer of data from memory to a register, or vice versa.

The B, C, D, and E half registers are all general purpose registers. The H and L register pair and the STACK POINTER register pair both have special functions in addition to being usable for general purposes. The game program does not make use of these special functions.

With the basic structure of the processor in mind, we can now look at the operation of the game program. Larger programs cannot be readily understood or written by working directly on the list of machine instructions, such as we did for Test Programs 1 and 2. A flow diagram is essential to quickly follow the sequence of the instructions and understand how they work together to achieve the desired result.

Figure 2 shows a flow diagram for Program 2. Each program function is briefly described in a separate box, and the flow of the executive of the program is indicated by the lines. Test Program 2 was a simple loop with no decisions so that after executing the short sequence of instructions, the flow of the program is back to the beginning of the loop to begin again. Figure 3 shows the flow diagram for the game program. Although it need not be understood to execute the game program, a thorough understanding of how this flow diagram achieves the operation of the game will be a useful step towards writing your own programs.



8080 PROCESSOR
FIG. 1

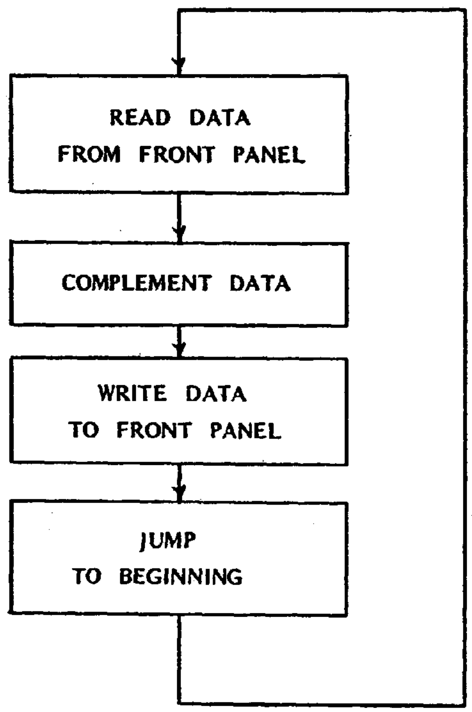

READ DATA
FROM FRONT PANEL
COMPLEMENT DATA
WRITE DATA
TO FRONT PANEL
JUMP
TO BEGINNING

FIG. 2

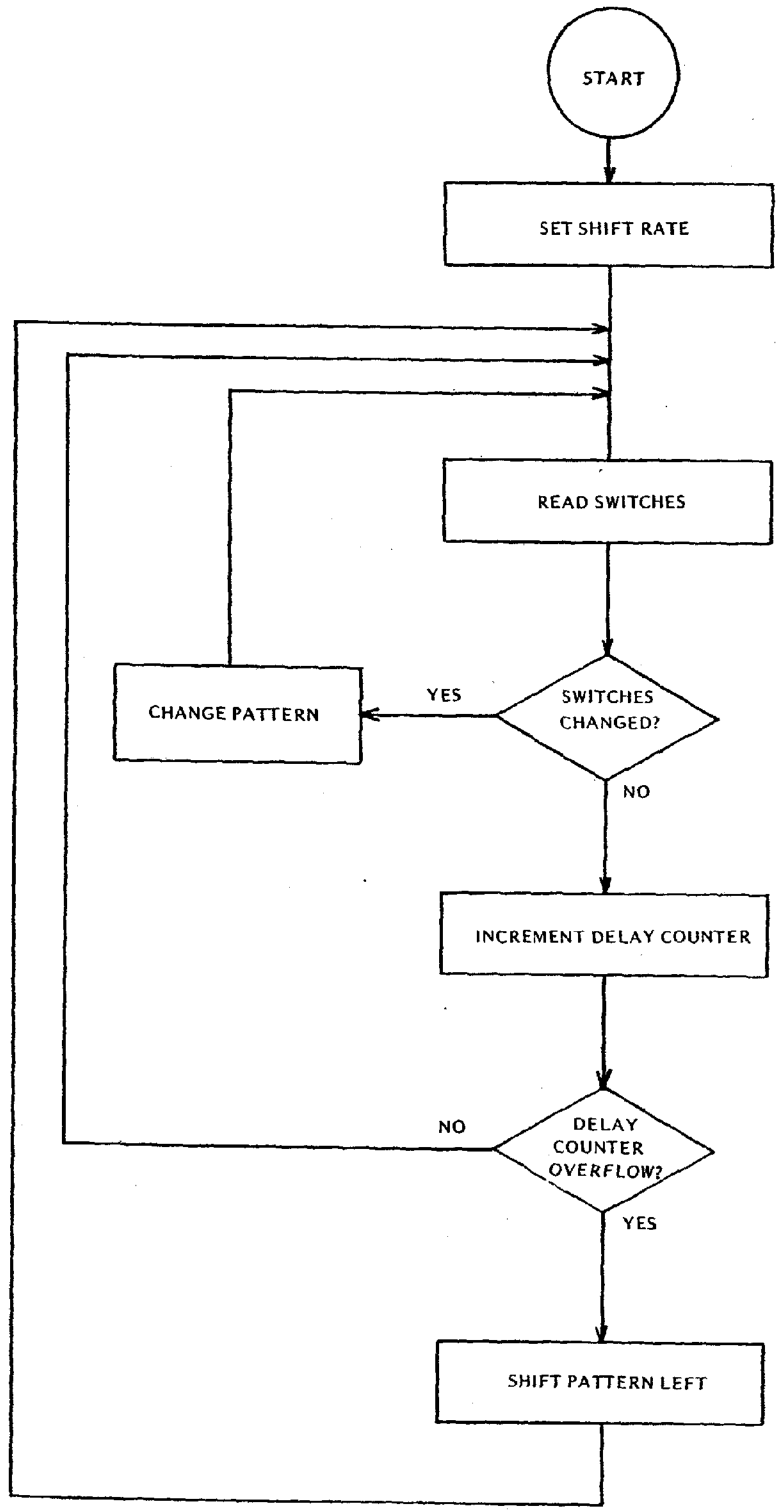


FIG. 3

GAME PROGRAM BASIC FLOWCHART

GAME PROGRAM LISTING

OCTAL		HEX			
ADD.	INST.	ADD.	INST.	MNEMONIC	DESCRIPTION
000 000	257	0000	AF	XRA, A	Exclusive OR A to itself (put zero in A)
001	147	01	67	MOV H, A	Move A to H (put zero in H)
002	333	02	DB	INP	Input data
003	377	03	FF		from front panel switches
004	157	04	6F	MOV L, A	Move A to L
005	371	05	F9	SPHL	Put H&L reg. into SP
006	257	06	AF	XRA, A	Exclusive OR A to itself (put zero in A)
007	201	07	81	ADD C	Put C in A, affecting flag bits
010	302	08	C2	JNZ	Jump if not zero
011	023	09	13		(skip switch test for debounce after a switch change)
012	000	0A	00		
					IF NORMAL, CONTINUE
013	123	0B	53	MOV D, E	Move E to D
014	333	0C	DB	INP	Input data
015	377	0D	FF		from front panel switches
016	137	0E	5F	MOV E, A	Move A to E
017	252	0F	AA	XRA, D	Exclusive OR D to A
020	302	10	C2	JNZ	Jump if result not all O's
021	041	11	21		(change display if switch position changed from last time)
022	000	12	00		
					IF SWITCHES UNCHANGED, CONTINUE
023	071	13	39	DAD SP	Add SP to HL
024	322	14	D2	JNC	Jump if no carry results
025	006	15	06		(return to read switch loop if no carry yet)
026	000	16	00		
					IF CARRY, CONTINUE
027	170	17	78	MOV A, B	Move B to A
030	007	18	07	RLC	Rotate left 1
031	107	19	47	MOV B, A	Store A in B
032	323	1A	D3	OUT	Output A
033	377	1B	FF		in front panel lights
034	257	1C	AF	XRA, A	Exclusive OR A to itself (put zero in A)
035	117	1D	4F	MOV C, A	Move A to C (Reset debounce indicator)

GAME PROGRAM LISTING (CONT.)

OCTAL ADD.	OCTAL INST.	HEX ADD.	HEX INST.	MNEMONIC	DESCRIPTION
036	303	1E	C3	JMP	Jump
037	006	1F	06		(to read loop)
040	000	20	00		
					CHANGE DISPLAY IF SWITCH DIFFERENT
041	250	21	A8	XRA, B	Exclusive OR B with A
042	107	22	47	MOV B, A	Store A in B
043	323	23	D3	OUT	Output A
044	377	24	FF		in front panel lights
045	257	25	AF	XRA, A	Exclusive OR a with itself A (put zero in A)
046	147	26	67	MOV H, A	Move A to H (set counter to insure enough delay for debounce)
047	057	27	2F	CMA	Complement A (to all l's)
050	117	28	4F	MOV C, A	Move A to C (set C to debounce)
051	303	29	C3	JMP	Jump
052	006	2A	06		(to read loop)
053	000	2B	00		

NOTE:

Exclusive OR of two switch patterns results in l's in positions which were changed, with all 0's elsewhere.

B= DISPLAY BYTE STORAGE

C=SWITCH DEBOUNCE INDICATOR
l=DEBOUNCE 0=NORMAL OPERATION

D=LAST SWITCH SETTINGS

E=CURRENT SWITCH SETTINGS

H,L=DELAY COUNTER

SP=INCREMENT FOR DELAY COUNTER

CABINET ASSEMBLY INSTRUCTIONS

Begin by installing the correct number of plastic card guides on the chassis part C's. The card guides should be placed from the front backwards, an equal number on each piece C, taking care that the wedge - shaped opening of the slot is positioned upwards. Note that the two ends of piece C are not symmetrical. The end with the wider space between the last small hole for mounting the card guide and the end flange is placed toward the back of the cabinet, so that the guides will line up with the connectors on the Mother board.

The card guides should be assembled starting from the front end (with card guide mounting holes placed closer to the end flange). Make sure you place the card guide so as to form a left hand and a right hand mounting-rail piece. If this is not done, then the card guides will be upside down on one of the two piece 'C's when they are mounted into the cabinet. The card guides are most easily mounted using a small press and placing the tab of each card guide in position started into the hole and pressing them into place until the mounting tabs snap through. A drill press with a large flat - headed screw mounted in the chuck works well with this operation. (Alternately, the card guides may be installed very carefully using needle nosed pliers). Care should be taken that the tabs are started into the hole when beginning to press the guide into place, otherwise one or both may be bent out flat and broken off. One end of the guide at a time should be inserted rather than trying to press both ends in simultaneously.

If a fan is to be installed in the chassis, it should be assembled on the back frame piece A1 at this time using the hardware in the fan kit. The fan terminals should be towards the top and towards the Mother board side of the chassis.

Next, the power cord should be inserted using the special grommet in the hole provided on the back panel. 4 to 6 inches of the power cord should be left on the inside of the cabinet. If the power cord grommet is squeezed together with a pair of pliers before insertion into the cabinet back, it will ease the job of inserting this tight fitting grommet. To insert the grommet, the power cord should be pulled through the hole nearly to the point where the grommet has been placed around the power cord, then the outer edge of the grommet can be grasped with a pair of pliers and squeezed slightly and inserted in the hole and worked in while slight tension is also being put on the cord from the back side to assist. Working this grommet in by rocking it back and forth works better than just pushing harder.

The front and back frames can now be screwed to the base plate using 6-32x5/16" machine screws. Note that the back frame fits <u>under</u> the base plate and the front frame fits on <u>top</u> of the base plate, set back about 1" from the front edge of the base plate. Next, install the two card frames between the front and back frames. Use two 6-32x5/16" machine screws at each end of each card frame. The front and back frames have slotted holes allowing the card frames to be adjusted slightly when the Mother board is installed on the base plate and boards are inserted in the card frames.

The self-adhesive rubber feet can then be separated from each other, the protective backing removed, and placed on the bottom of the cabinet spaced 3 inches along the left hand and right hand edge of the bottom, to support the cabinet weight.

BASE PLATE HOLE IDENTIFICATION

The base plate currently being shipped is a universal base plate, with extra holes for accomodating two styles of mother board mounting systems and two styles of power supplies. For the power supply and mother board systems shipped with your kit:

1. Place the Power Supply p.c. board in the base plate cavity and line up the holes in the p.c. board with the corresponding holes in the base plate and mark (e.g., with a felt-tip pen) which holes are to be used.

2. The mother board mounting system uses the two rows of 12 holes each on the left side of the base plate.

SWITCH ESCUTCHEON INSTALLATION

When the CP-A Front Panel Assembly has been mounted, the Switch Escutcheon (piece A2B) can be installed on the base plate at the front of the computer using four 6-32x5/16" Phillips pan head machine screws. Note that the Escutcheon should fit <u>under</u> the base plate.

CABLE CLAMP INSTALLATION

Cables that do not fit the connector holes on the back frame of the chassis may be clamped for strain relief at the top of the back frame using the L - shaped aluminum bar, piece K. Install using two 6-32x½" Phillips pan head machine screws. Depending on the thickness of the cables being clamped, either of the two sides of the angle may be used.

TABLE TOP COVER INSTALLATION

To install the table top cover, slide the cover carefully over the chassis frame and hold in place with four 6-32x½" Phillips pan head machine screws.

Refer to Appendices for an exploded view of the chassis cabinet.

RACK MOUNT SYSTEM ASSEMBLY INSTRUCTIONS

For the rack mount system, begin by installing the rack mount cover on the chassis. Use five 6-32x5/16" Phillips pan head machine screws. Next install the left and right side plates to the chassis with the front flanges pointing outwards. Use four 6-32x½" Phillips pan head machine screws on each side plate. The forward holes in each pattern on the side plates should be used.

Next mount the assembly in the rack using two screws on each side of the front flanges. Hardware requirements for mounting the assembly into the rack will vary according to the individual rack. It is suggested that the rear of the assembly also be supported in the rack. Finally, mount the front face panel onto the side plate flanges using four #10 round head screws and clips.

NOTE: for installations without slides where easy removal of the computer is desired, the side plates can be mounted directly in the rack and the computer can be slid on its rubber feet on the bottom flanges of the side plates. The rear of the side plates in this case should be fastened securely to the back of the rack cabinet.

8080 Rack Mount
Parts List

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION
	93-3010008	1	Rack Mount Front Panel Rev. C
	93-3070001	1	Rack Mount Left Slide Rev. C
	93-3070002	1	Rack Mount Right Slide Rev. O
	93-3010012	1	Rack Mount Cover Rev. B
Screw	20-3302001	5	6-32x5/16" Phillips Pan Head Machine Screw
Screw	20-3502001	8	6-32x½" Phillips Pan Head Machine Screw
Screw	20-5707001	4	#10x3/4" Flat Head Type B Self-Tapping Sheet Metal Screw
Screw	20-5708001	4	#10x3/4" Button Head Type B Self-Tapping Sheet Metal Screw
Nut	21-5650001	8	Speed Nut, Tinnerman C 9031-10Z-1

8080 Chassis
Parts List

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Back Frame	93-3010003	1	Back Frame "A1"
Front Piece	93-3010010	1	Painted Front Piece "A2B"
Cabinet Base	93-4010004	1	Cabinet Base "A3"
Front Frame	93-3010001	1	Front Frame "B"
Card Frame	93-3010002	2	Card Frames "C"
Cable Clamp	93-3010013	1	Cable Clamp "K"
"DANGER" Label		1	"DANGER" Label on right card frame
Rubber Feet	28-0400001	6	Adhesive backed rubber feet
Screw	20-3302001	25	6-32x5/16" Phillips pan head machine screws
Screw	20-3502001	2	6-32x¼" Phillips pan head machine screws
Capacitor Brace	91-0300001	2	Capacitor Braces with foam backing
Cover	93-4010005	1	Table Top Cabinet Top
Screws	20-3502001	4	6-32¼" Phillips Pan Head Machine

IMSAI

MOTHERBOARD

MOTHER BOARD

FUNCTIONAL DESCRIPTION

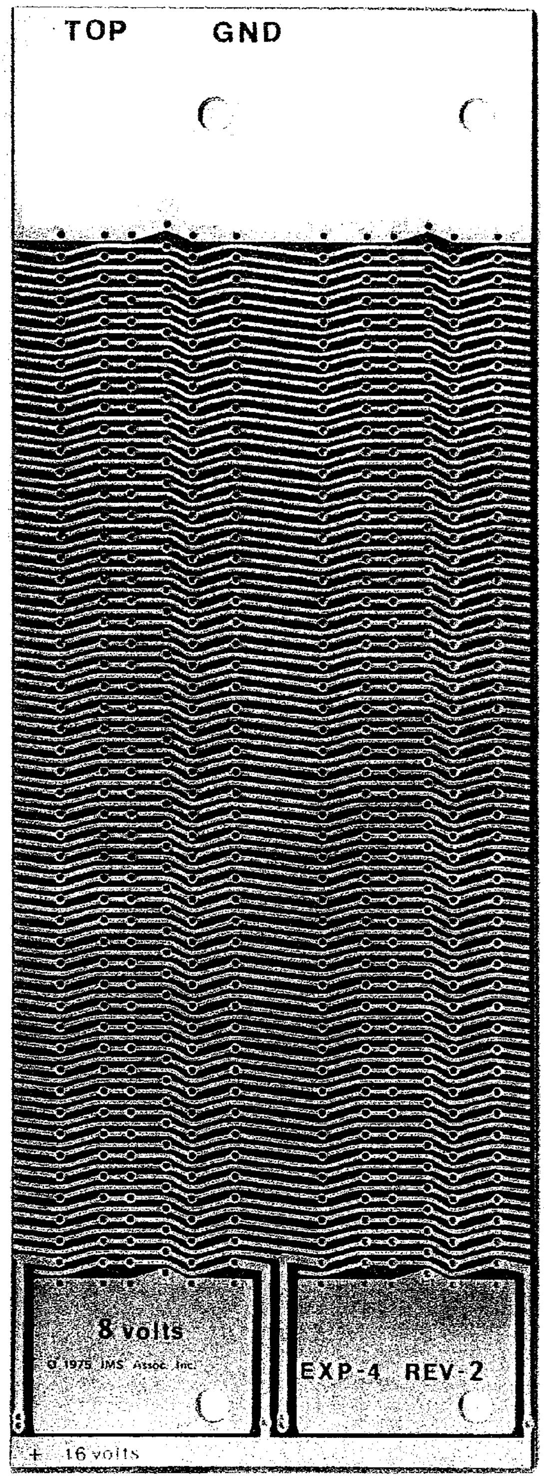
The IMSAI 8080 system Mother boards are available in three different length sections varying from a minimum of 4 printed circuit card connector positions. The basic system includes a Mother board with six connector positions on it. One is used for the front panel and the other five are available for the MPU and any combination of memory or I/O cards.

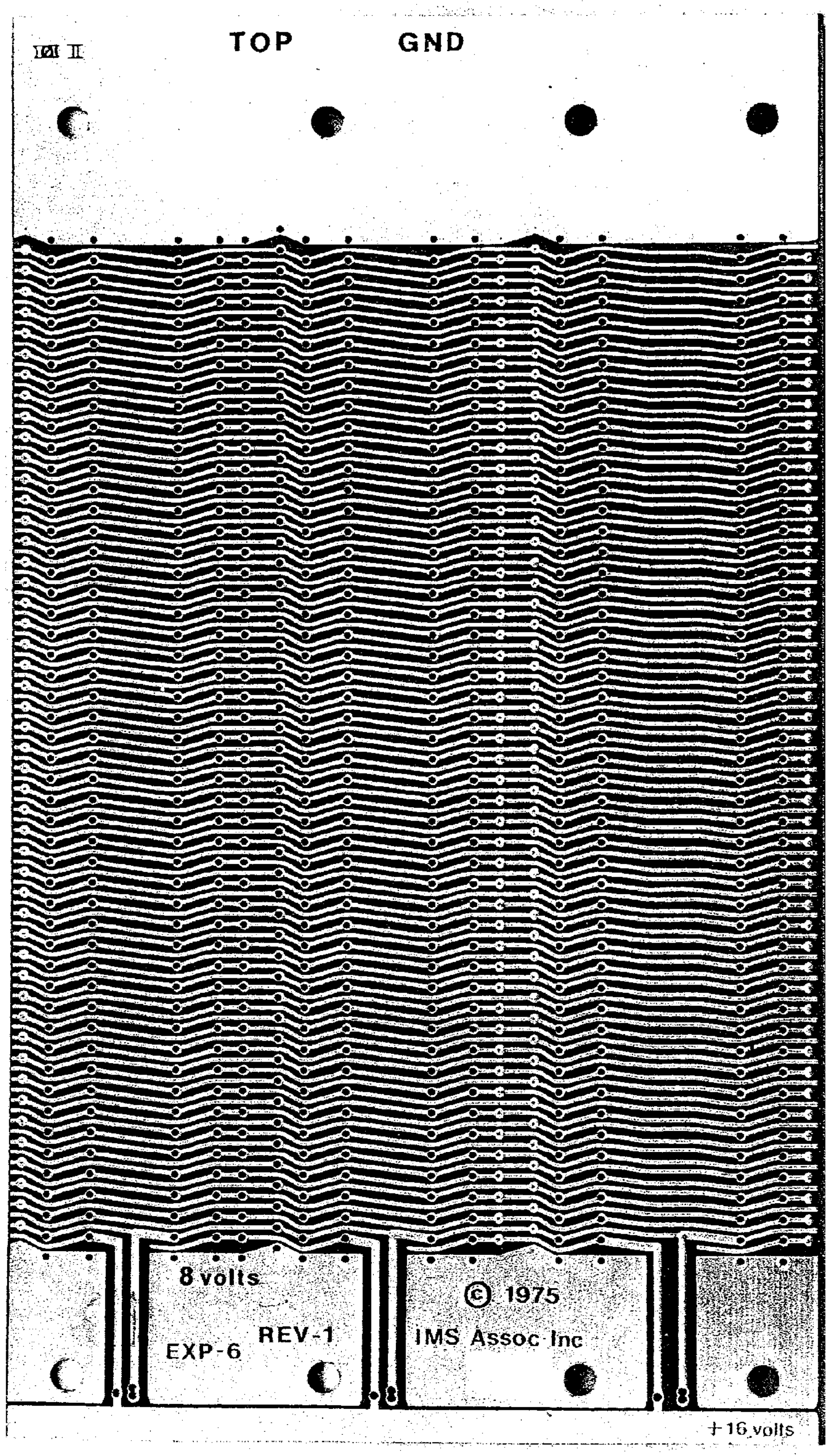
The card-to-card spacing on the Mother board is 3/4 inch except for the front position which is reserved for the front panel board or the parallel I/O board for the dedicated processor to accommodate mounting the card in the special front position in the cabinet.

Additional sections of Mother board are available with positions for 4 connectors. These may be added to the system at a later date, and connected to the previous Mother board sections by jumpers between the sections soldered into provided holes. No jumper wire soldering is required if the full-length board is purchased.

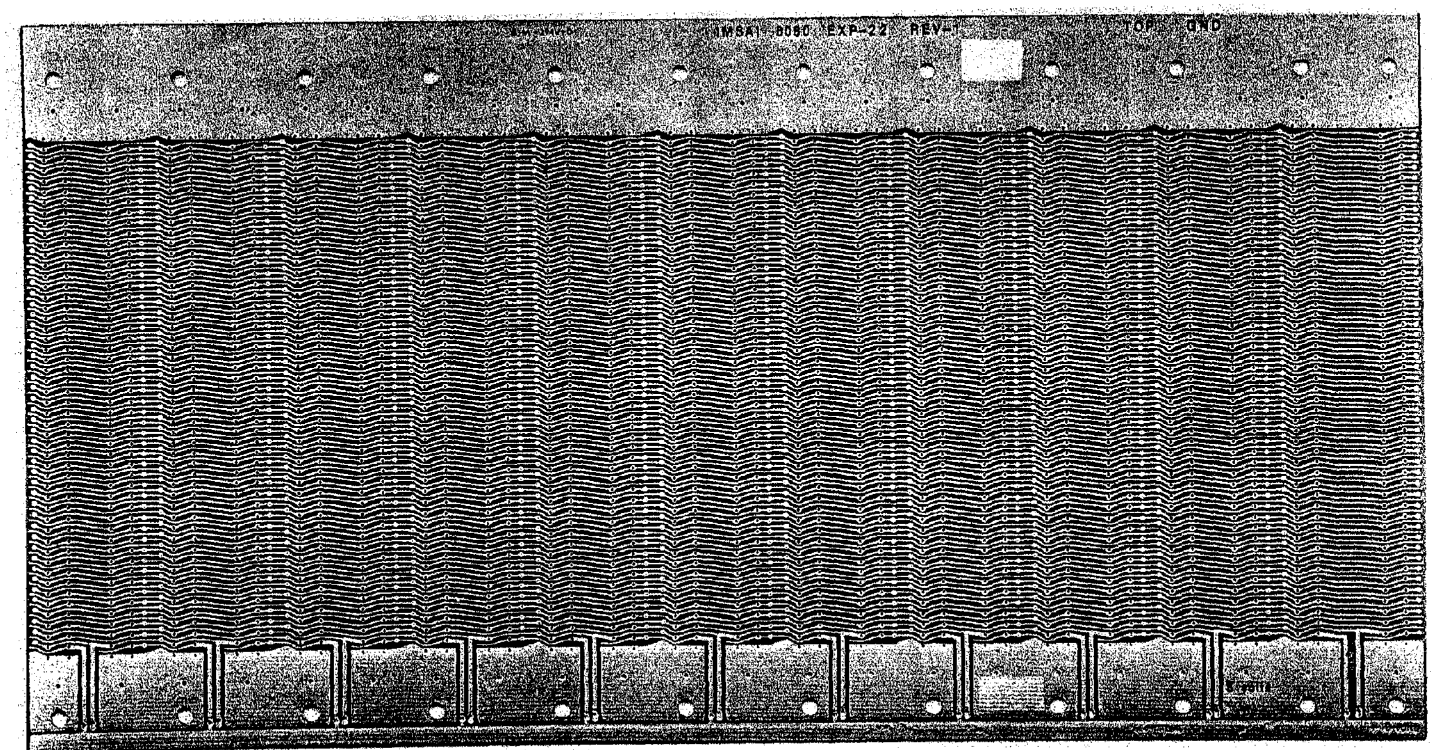
The Mother board is 1/16 inch printed circuit board with double-sided plated-through holes. Each of the connector pins is connected by traces on both sides of the board. Heavy power traces are provided to handle the very large currents involved in a fully-loaded back plane. The two connectors supplied with the IMSAI system are high-quality gold-plated-contact connectors, for reliable contacts and long life.

Trace spacing is tightly controlled on the board to avoid any close spots where shorts from solder bridges might tend to occur. The traces on Mother board are plated for better appearance and more reliable solder connections. A solder mask is provided on both sides of the Mother board.

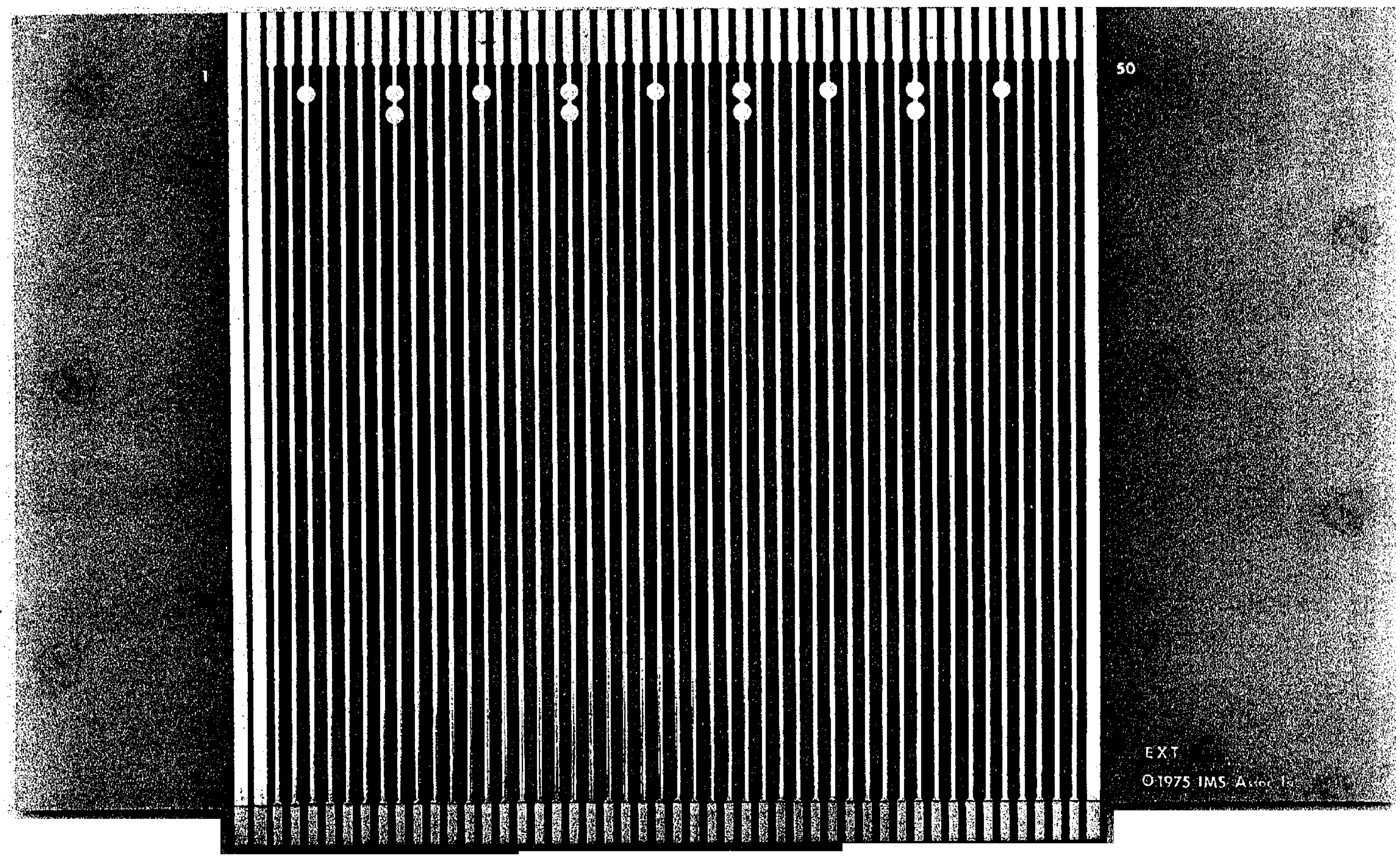

TOP
GND
8 volts
© 1975 IMS Assoc. Inc.
EXP-4 REV-2
+ 16 volts



EXP-6



EXP-22 Rev. 1



EXT (EXTENDER BOARD)

Mother Board
Parts List

EXP-4

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
PC Board	92-0000004	1	4-Slot Printed Circuit Board
Washer	21-3330001	8	#6 Sholder Fiber Washer
Spacer	21-4600001	4	6-32x¼" Threaded Spacer
Nut	21-3120001	4	6-32 Nut
Screw	20-3701002	4	6-32x3/4" Nylon Screw

EXP-6

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
PC Board	92-0000005	1	6-Slot Printed Circuit Board
Washer	21-3330001	16	#6 Shoulder Fiber Washer
Spacer	21-4600001	8	6-32x¼" Threaded Spacer
Nut	21-3120001	8	6-32 Nut
Screw	20-3701002	8	6-32x3/4" Nylon Screw

EXP-22

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
PC Board	92-0000006	1	22-Slot Printed Circuit Board
Wahser	21-3330001	48	#6 Shoulder Fiber Washer
Spacer	21-4600001	24	6-32x¼" Threaded spacer
Nut	21-3120001	24	6-32 Nut
Screw	20-3701002	24	6-32x3/4" Nylon screw

MOTHER BOARD

ASSEMBLY INSTRUCTIONS

The Mother board appears to be the simplest of all the boards to assemble. The solder mask minimizes the chances of shorting adjacent traces. However, it is imperative that extra care be taken during assembly to avoid excess solder shorting adjacent pins. Because a short on the Mother board is extremely hard to locate and correct when it is between the board and the connector, it is worth the builder's time to give special attention to making certain that no such shorts occur. Use only as much solder as required for a good joint. If too much solder is used, either the pool of solder can short to an adjacent pin on the top side or the solder can leak through and form a ball on the backside which can also short to an adjacent pin.

The board should be checked with an ohmmeter carefully both before and after assembly to insure that it will operate properly. While the chance that incomplete etching during manufacture left two traces shorted is extremely slight, the ohmmeter check before assembly is worth while simply because it would be so difficult to correct such a problem after a socket is soldered in place over it.

To test the board, either a simple ohmmeter or a battery connected to a buzzer or a light bulb and test leads are all that is required. Each pair of adjacent traces should be checked with the continuity tester to be sure that there is no connection between them. Should any adjacent traces be found to be electrically connected during this pre-assembly check, careful inspection of the board should reveal the short. Any imcompletely-etched copper or other metallic path between the two traces shold be removed with a sharp knife, such as an X-acto knife.

After each connector is soldered in, the continuity check should be made again to make certain that during assembly no shorts were created. If any are discovered, steps should be taken to remove them before further assembly. In most cases, this short will have been caused by too much solder having been applied and may be removed simply by removing the excess solder. If an Extender board is available, a simple tester may be made from it by temporarily connecting all the pins on the front side, except pin 1, together, connecting all the pins on the back side, except pin 100, together and then connecting the continuity tester between the two sides of the Extender board. If this extender board is inserted in the socket as it is being soldered, the continuity tester will indicate immediately any short between any two adjacent traces.

SOCKET INSERTION

The 100 pin edge connectors are symmetrical so that they may be inserted either way. The connector stands off the board slightly supported by raised feed at each end. Each connector should be checked during assembly to make sure that it is seated properly and that the Mother board near the center of the connector is neither pushed further toward the connector nor lifted away before the connector is soldered in place to prevent the Mother board from bowing.

The Mother board is not completely symmetrical and the connectors must be inserted from the top side. The top side is the side on which the +8 volt foil is broken every 2 connectors to allow the 2 traces for + and -16 volts to extend from the 16 volt bus at the end of the board into the connector pins. The back side of the board has both the very heavy ground bus and the 1 inch wide 8 volt foil area continuous for the full length of the board. The +16 volt trace is the .2 inch trace on the edge of the board alongside the +8 volt bus on the front side, that is, the side where the +8 is broken to allow for the pairs of +16 volt traces to extend into the pins. The -16 volt bus is the .2 inch wide trace along the edge of the board on the back side underneath the +16 volt bus. NOTE: Before mounting any connectors, locate the front of the Mother board. The connector for the front panel (CPA board) needs to be mounted in the first position at the front of the Mother board. Notice that the spacing between the first and second positions at the front of the Mother board is wider than the spacing between any two of the other connector positions.

The suggested procedure for inserting and soldering a connector is to insert the connector in place, seat the two ends firmly against the feet and solder the two pins on each end.

Next, the position of the center of the Mother board next to the connector should be checked and either pushed further toward the connector or pulled away so that the gap between the connector and the Mother board is uniform all the way across. Then the two pins in the center of the connector should be soldered.

One final check should be made to make sure that the gap is uniform all the way across the connector and the remaining pins in the connector should be soldered.

Care should be taken to check each connector after solder to make sure that every pin was soldered because it is easy to miss a pin and not see it during a quick inspection. After the last connector is soldered in place and the board checked you are ready to install the power connections and mount the board in the cabinet.

See MAINFRAME ASSEMBLY section for connecting the Mother board to the Power Supply and mounting the Mother board in the chassis.

Mother Board

USER GUIDE

With the proper care taken during assembly, the Mother board should be the most reliable board in the system. The only attention the user will typically put on the Mother board, is when he desires to add more card slot positions. Either 4 slot extension Mother Boards may be added to the original 6 slot Mother Board, or the 6 slot board may be replaced by a new 22 slot board.

If 4 slot extension(s) are used, the extension(s) should be assembled according to instructions for assembling the original Mother board. Then the original Mother board must be removed from the cabinet and jumpered to the new section by the use of short wire jumpers between the connection points provided in each trace.

The power buses should be connected with a much heavier wire. The two boards can then be reassembled into the cabinet.

Care should be taken when inserting jumpers that each goes between the corresponding lines on the two sections of Mother board.

IMSAI 8080 BUS SIGNAL LIST

1	+8v	51	+8v
2	+16v	52	-16v
3	XRDY	53	$\overline{\text{SSW DSB}}$
4	VI 0	54	$\overline{\text{EXT CLR}}$
5	VI 1	55	*
6	VI 2	56	
7	VI 3	57	
8	VI 4	58	
9	VI 5	59	
10	VI 6	60	
11	VI 7	61	
12		62	
13		63	
14		64	
15		65	
16		66	
17		67	
18	$\overline{\text{STATUS DSBL}}$	68	MWRITE
19	$\overline{\text{CCDSBL}}$	69	****
20	**	70	***
21	SS	71	RUN
22	$\overline{\text{ADDR DSBL}}$	72	PRDY
23	$\overline{\text{DO DSBL}}$	73	$\overline{\text{PINT}}$
24	φ2	74	$\overline{\text{PHOLD}}$
25	φ1	75	$\overline{\text{PRESET}}$
26	PHLDA	76	PSYNC
27	PWAIT	77	$\overline{\text{PWR}}$
28	PINTE	78	PDBIN
29	A 5	79	A 0
30	A 4	80	A 1
31	A 3	81	A 2
32	A 15	82	A 6
33	A 12	83	A 7
34	A 9	84	A 8
35	DO 1	85	A 13
36	DO 0	86	A 14
37	A 10	87	A 11
38	DO 4	88	DO 2
39	DO 5	89	DO 3
40	DO 6	90	DO 7
41	DI 2	91	DI 4
42	DI 3	92	DI 5
43	DI 7	93	DI 6
44	SMI	94	DI 1
45	SOUT	95	DI 0
46	SINP	96	SINTA
47	SMEMR	97	SWO
48	SHLTA	98	SSTACK
49	$\overline{\text{CLOCK}}$ (2 MHz)	99	$\overline{\text{POC}}$
50	GND	100	GND

* reserved for chassis ground
** reserved for memory unprotect
*** reserved for memory protect
**** reserved for protect status

BUS DEFINITION

Front Side

No.	SYMBOL	NAME	FUNCTION
5	V11	Vectored Interrupt Line # 1	
6	V12	Vectored Interrupt Line #2	
7	V13	Vectored Interrupt Line #3	
8	V14	Vectored Interrupt Line #4	
9	V15	Vectored Interrupt Line #5	
10	V16	Vectored Interrupt Line #6	
11	V17	Vectored Interrupt Line #7	
12 to 17	UNUSED		
18	$\overline{\text{STATUS DSBL}}$	STATUS DISABLE	Allows the buffers for the 8 status lines to be tri-stated
19	$\overline{\text{CC DSB}}$	COMMAND CONTROL DISABLE	Allows the buffers for the 6 output command/control lines to be tri-stated
20	UNPROT	UNPROTECT	Reserved for input to the memory protect flip-flop on a given memory board
21	SS	SINGLE STEP	Used by Front Panel to disable input buffer while panel drives bidirectional data bus

BUS DEFINITION

Front Side

No.	SYMBOL	NAME	FUNCTION
22	$\overline{\text{ADDR DSBL}}$	ADDRESS DISABLE	Allows the buffers for the 16 address lines to be tri-stated
23	$\overline{\text{DO DSBL}}$	DATA OUT DISABLE	Allows the bidirectional data bus drivers for the 8 data lines to be tri-stated for both input and output data buses
24	Ø2	Phase 2 Clock	
25	Ø1	Phase 1 Clock	
26	PHLDA	Hold Acknowledge	Processor control output signal which appears in response to the HOLD signal; indicates that the data and address bus will go to the high impedance state on the 8080. Note: $\overline{\text{ADDR DSBL}}$ and $\overline{\text{DO DSBL}}$ must be driven to tri-state the system bus
27	PWAIT	WAIT	Processor control output signal which acknowledges that the processor is in a WAIT state
28	PINTE	INTERRUPT ENABLE	Processor control output signal indicating interrupts are enabled: may be set or reset by EI and DI instruction and inhibits interrupts from being accepted by the CPU if it is reset

BUS DEFINITION

Front Side

No.	SYMBOL	NAME	FUNCTION
29	A5	Address Line #5	
30	A4	Address Line #4	
31	A3	Address Line #3	
32	A15	Address Line #15	
33	A12	Address Line #12	
34	A9	Address Line #9	
35	DO	Data Out Line #1	
36	DO0	Data Out Line #0	
37	A10	Address Line #10	
38	DO4	Data Out Line #4	
39	DO5	Data Out Line #5	
40	DO6	Data Out Line #6	
41	D12	Data In Line #2	
42	D13	Data In Line #3	
43	D17	Data In Line #7	
44	SM1	M1	Status output signal that indicates that the processor is in the fetch cycle for the first byte of an instruction
45	SOUT	OUT	Status output signal which indicates that the address bus contains the address of an output device and the data bus will contain the output data when $\overline{PWR}$ is active

BUS DEFINITION

Front Side

No.	SYMBOL	NAME	FUNCTION
46	SNIP	INP	Status output signal which indicates that that the address bus contains the address of an input device and the input data should be placed on the data bus when PDBIN is active
47	SMEMR	MEMR	Status output signal which indicates that the data bus will be used for memory read data
48	SHLTA	HLTA	Status output signal which acknowledges a HALT instruction
49	$\overline{\text{CLOCK}}$	CLOCK	2 MHz clock signal
50	GND	GROUND	

Back Side

No.	SYMBOL	NAME	FUNCTION
51	+8V	+8 volts	Unregulated input to 5v regulators
52	-16V	-16 volts	Negative unregulated voltage
53	$\overline{\text{SSW DSB}}$	SENSE SWITCH DISABLE	Disables the data input buffers so the input from the sense switches may be strobed onto the bidirectional data bus
54	$\overline{\text{EXT CLR}}$	EXTERNAL CLEAR	Clear signal for I/O devices (front panel switch closure to ground)

BUS DEFINITION

Back Side No.	SYMBOL	NAME	FUNCTION
55	CGND	CHASSIS GROUND	
56 to 67	UNUSED		
68	MWRT	MEMORY WRITE	From the Front Panel indicates that the current data on the Data Out Bus is to be written into the memory location currently on the address bus
69	$\overline{PS}$	PROTECT STATUS	Reserved to indicate the status of the memory protect flip-flop on the memory board currently addressed
70	PROT	PROTECT	Reserved for input to the memory protect flip-flop on the memory board currently addressed
71	RUN	RUN	Indicates that the RUN/STOP flip-flop is set to run on the front panel
72	PRDY	READY	Processor command/control input that controls the run state of the processor; if the line is pulled low the processor will enter a wait state until the line is released

BUS DEFINITION

Back Side No.	SYMBOL	NAME	FUNCTION
73	$\overline{\text{PINT}}$	INTERRUPT REQUEST	The processor recognizes an interrupt request on this line at the end of the current instruction or while halted. If the processor is in the HOLD state or the Interrupt Enable flip-flop is reset, it will not honor the request
74	$\overline{\text{PHOLD}}$	HOLD	Processor command input signal which requests the processor to enter the HOLD state; allows an external device to gain control of address and data buses as soon as the processor has completed its use of these buses for the current machine cycle
75	$\overline{\text{PRESET}}$	RESET	Processor command input; while activated the content of the program counter is cleared and the instruction register is set to 0
76	PSYNC	SYNC	Processor control output provides a signal to indicate the beginning of each machine cycle
77	$\overline{\text{PWR}}$	WRITE	Processor control output used for memory write or I/O output control; continued next page.

BUS DEFINITION

Back Side

No.	SYMBOL	NAME	FUNCTION
77	$\overline{PWR}$	WRITE	Con't.: data on the data bus is stable while the $\overline{PWR}$ is active
78	PDBIN	DATA BUS IN	Processor control output signal indicates to external circuits that the data bus is in the input mode
79	A0	Address Line #0	
80	A1	Address Line #1	
81	A2	Address Line #2	
82	A6	Address Line #6	
83	A7	Address Line #7	
84	A8	Address Line #8	
85	A13	Address Line #13	
86	A14	Address Line #14	
87	A11	Address Line #11	
88	DO2	Data Out Line #2	
89	DO3	Data Out Line #3	
90	DO7	Data Out LIne #7	
91	D14	Data In Line #4	
92	D15	Data In Line #5	
93	D16	Data In Line #6	
94	D17	Data In Line #1	
95	D10	Data In Line #0	

BUS DEFINITION

Back Side

No.	SYMBOL	NAME	FUNCTION
96	SINTA	INTA	Status output signal to acknowledge signal for INTERRUPT request
97	SWO	WO	Status output signal indicates that the operation in the current machine cycle will be a WRITE memory or output function
98	SSTACK	STACK	Status output signal indicates that the address bus holds the pushdown stack address from the Stack Pointer
99	$\overline{POC}$	Power-On Clear	
100	GND	GROUND	

IMSAI

PS-28U

POWER SUPPLY PS-28U

FUNCTIONAL DESCRIPTION

The IMSAI PS-28U is a modular, unregulated power supply for the IMSAI 8080 System. It provides the basic unregulated +8, +16, and -16 system supply voltages and can be configured for the following AC input voltages at either 50 or 60 Hz: 92, 103.5, 115, 126.5, 184, 207, 230, and 253 VAC single phase input.

A power switch location is provided on the PS-28U for use when a front panel is not installed in the system. There is also a line filter and 50/60 Hz switched and unswitched terminals for connecting auxillary power outlets on the back panel.

Physically, the PS-28U measures 16.5" x 5.75" x 5.5" (42 x 15 x 14 cm) and weighs 16 pounds (7.3 kg).

SPECIFICATIONS PS-28U SUPPLY

Power Requirements:

Input Voltages:	92, 103.5, 115, 126.5, 184, 207, 230, and 253 volts, single phase, 500 watts (max)
No Load Voltages:	115 VAC, 60 Hz input, nominal taps #6 and #9 in parallel with taps #1 and #4
	+ 8v. supply: + 9.7 volts +16v. supply: +18.0 volts -16v. supply: -18.0 volts

Current Supplied:

At 115 VAC, 60 Hz, resistive load:

28.0 amperes at 7.0 volts ripple valley
4.5 amperes at +13.5 volts ripple valley
4.5 amperes at -13.5 volts ripple valley

At 100 VAC, 50 Hz, resistive load:

25.0 amperes at +7.0 volts ripple valley
4.0 amperes at +13.5 volts ripple valley
4.0 amperes at -13.5 volts ripple valley

THEORY OF OPERATION

The PS-28U is an unregulated power supply that provides the basic +8, +16, and -16 voltages for the 8080 system. It is comprised of four major component assemblies: line filter, transformer, rectifiers, and filters.

Line Filter: The line filter is a triple PI L-C filter designed to remove high frequency noise present on the AC line. This filter attenuates . line noise above 1MHz in frequency.

Transformer: The transformer is primarily designed for a number of AC input voltages: 92, 103.5, 115, 126.5, 184, 207, 230, and 253 VAC, 50/60 Hz, single phase input. The transformer secondary is connected as three series windings with a center tap. Four MR 1121 diodes full-wave rectify the +8 volts, while a full-wave bridge of four MR 501 diodes rectify the ± 16 volts.

Filtering: The ±16 volt supplies are each filtered by a 10K uF capacitor to ground, providing ±15 average volts at 4.0 amps. The +8 volts is filtered by two 95K uF capacitors to ground, providing 7.3 average volts at the 28 amp rated current.

.1 uF capacitors high frequency bypass each voltage supply and bleeder resistors discharge the filter capacitors when power is turned off.

BOARD: PS-C

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Solder	15-0000001	5'	Rosin Core
Heat Sink	16-0100006	1	Wakefield 690-220-P, Modified
Screw	20-3402001	4	6-32x3/8" Phillips Pan Head Machine
Screw	20-3702001	4	6-32x3/4" Phillips Pan Head Machine
Screw	20-4401001	3	8-32x3/8" Binding Head Machine
Screw	20-4901001	5	8-32x1¼" Binding Head Machine
Screw	20-5402000	8	10-32x3/8" Binding Head Machine
Screw	20-6901001	4	¼-20x1½" Binding Head Machine
Nut	21-3120001	8	6-32 Cad Hex Nut
Lockwasher	21-3350001	8	#6 Internal Tooth
Nut	21-4120001	5	8-32 Cad Hex Nut
Lockwasher	21-4350001	5	#8 Internal Star
Spacer	21-4600002	5	8-32x½" Nylon Threaded
Nut	21-5120001	4	10-32 Cad Hex Nut
Lockwasher	21-5320001	4	#10 Cad Split Ring
Lockwasher	21-5350001	8	#10 Internal Star
Nut	21-6120001	4	¼-20 Cad Hex Nut
Washer	21-6310001	4	¼"x1/16" Cad Flat Washer
Lockwasher	21-6320001	4	¼" Split Ring
Washer	21-6390001	4	¼"x1/16" Nylon Washer
Spacer	21-6600001	4	¼-20x½" Nylon Internal Thread
Wire	22-1014001	48"	14 AWG, White, Alpha 1559, 14-41/30 PVC
Wire	22-1014002	60"	14 AWG, Black, Alpha 1559, 14-41/30 PVC

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Wire	22-1018001	60"	18 AWG, Orange, Gavitt 8522
Wire	22-1018002	60"	18 AWG, Yellow, Gavitt 8522
Wire	22-5018001	12"	Twisted Pair, 18 AWG, Yellow/orange, Stranded and Insulated
Line Cord	22-6000001	1	Belden 17239
Grommet	24-0600001	1	Strain Relief Bushing Grommet
Terminal Lug	25-0100001	5	Panduit PV-14-10LF (Vinyl)
Terminals	25-0100002	10	Solderless, ¼", Vaco # D 18304
Transformer	29-0100010	1	Tranex 4-3819-1 Dual Primary
Inductor	29-0200001	3	8uH, 5 Amp, Airco Speer 025834-001K
Resistor	30-3470462	1	470 Ohm, ½ Watt/Yellow, Violet, Brown
Resistor	30-4100462	2	1K Ohm, ½ Watt/Brown, Black, Red
Capacitor	32-2004010	6	.04uF, 500 V Disk Ceramic (.01uF, 1000 V)
Capacitor	32-2010010	3	.1uF, 30 V Disk Ceramic
Capacitor	32-2510060	2	10KuF, 25 V Electrolytic
Capacitor	32-2595060	2	95KuF, 15 V Electrolytic
Fuse	33-0100003	1	Bussman Fusetron MTH 5, 5 Amp
Fuse	33-0100004	1	Bussman Fusetron AGC 2½, 2½ Amp
Fuse Clip	33-0200001	2	# 102068
Fan Guard	34-0200001	1	Rotron 476042
Rectifier	35-1000002	4	MOT MR 1121
Diode	35-1000003	4	MOT MR 501 (Alt: 30S1)
PC Board	92-0000024	1	PS-C Rev. 1
Label Plate	93-0000001	1	Voltage/Frequency Label Plate

ASSEMBLY INSTRUCTIONS

( ) 1. Unpack your board and check all parts against the parts lists enclosed in the package.

COMPONENT INSTALLATION

( ) 2. Insert and solder each of the two 1K Ohm, ½ watt resistors (brown, black, red) at locations R1 and R2 as shown on the Assembly Diagram.

( ) 3. Insert and solder the one 470 Ohm, ½ watt resistor (yellow, violet, brown) at location R3 as shown on the Assembly Diagram.

( ) 4. Insert and solder each of the three .1uF capacitors at locations C5, C6 and C4 as shown on the Assembly Diagram.

( ) 5. Next, bend each of the cathode leads on each of the four rectifier diodes CR4, CR5, CR6 and CR7 as shown in Figure 2. Insert the anode end of of the diodes down as shown in Figure 2 and solder. NOTE: See Assembly Diagram for diode mounting position.

( ) 6. Insert and solder each of the six .04 uF capacitors at locations C7 through C12 as shown on the Assembly Diagram.

( ) 7. Insert and solder each of the three AC filter inductors at locations L1, L2 and L3 as shown on the Assembly Diagram.

( ) 8. Insert and solder each of the two fuse clips in the appropriate locations as shown on the Assembly Diagram. Snap in the appropriate fuse.

TRANSFORMER WIRING

NOTE: There are five pages of diagrams following the the Assembly Instructions. Refer to them when wiring the transformer.

( ) 9. Transformer terminals are designated and used as follows:

Primary A		Primary B	
Pin 1	Common	Pin 6	Common
Pin 2	20% Lo Line	Pin 7	20% Lo Line
Pin 3	10% Lo Line	Pin 8	10% Lo Line
Pin 4	Nominal (115/230 VAC)	Pin 9	Nominal (115/230 VAC)
Pin 5	10% Hi Line	Pin 10	10% Hi Line

Secondary (8080 Chassis)

Pin 13 AC Phase 1 to 8V Rect
Pin 15 AC Phase 2 to 8V Rect
Pin 14 Ground

Pin 11 AC Phase 1 to 16V Rect
Pin 17 AC Phase 2 to 16V Rect
Pin 12 tie to Pin 13
Pin 16 tie to Pin 15

Primary Wiring Configurations

Input VAC 50/60 Hz	Strap these Primary lugs	Connect input VAC wires to these lugs
92 VAC	1 to 6, 2 to 7	6 and 7
103.5 VAC	1 to 6, 3 to 8	6 and 8
115 VAC	1 to 6, 4 to 9	6 and 9
126.5 VAC	1 to 6, 5 to 10	6 and 10
184 VAC	6 to 2	1 and 7
207 VAC	6 to 3	1 and 8
230 VAC	6 to 4	1 and 9
253 VAC	6 to 5	1 and 10

Again, be sure to refer to the accompanying diagrams when wiring the transformer.

( ) 10. Solder a ¼" solderless terminal to one end of two 9" yellow wires. Then solder the other ends to the pads at CR4 - CR7. These wires then go to lugs #11 and #17 on the secondary of the transformer.

( ) 11. The other secondary is wired as follows: Lugs #12 and #13 are wired together, and lugs #15 and #16 are wired together. Again, use the ¼" solderless terminals for the connections to lugs #12 and #16; use black wire 5 inches long (#14 or larger). The connection to lugs #13 and #15 are made with the crimp terminals.

( ) 12. Attach a crimp terminal to a 3 inch piece of #14 black wire. Solder one end to the ground trace below lug #14 and then attach the crimp terminal to lug #14.

( ) 13. Note: the AC input lines should be twisted together to avoid radiation.
When operating between 92 VAC and 126 VAC, both COMMONs are tied together, the nearest applicable voltage taps selected and jumpered together, and the AC applied between COMMONs and the taps, essentially paralleling the primaries. It may be desirable to select the next lower taps when operating on 50 Hz line, or when using a fully-loaded chassis.

( ) 14. For AC inputs between 184 VAC and 253 VAC, the primaries should be series connected. This entails selecting the taps as previously described. Now, the AC input goes between the COMMON of one primary and the selected tap of the other primary. A jumper is used between the selected tap of the first primary and the COMMON of the second primary to complete the series circuit. The same considerations regarding 50 Hz and full chassis apply here also as in the 115 VAC case preceeding. For 230 VAC operation, the AC line fuse should be changed to one-half the value recommended for 115 VAC to maintain the same overload protection.

( ) 15. The fan (optional) leads always should be connected to lugs #6 and #9 or #1 and #4 to supply 115 VAC to the fan. This wiring is standard for all input AC wiring configurations.

HEAT SINK INSTALLATION

NOTE: Keep all wiring as short as possible, an extra two inches of #14 wire will reduce the current capacity of the Power Supply.

( ) 16. Insert the four 1121 rectifier diodes CR0 through CR3 through the heat sink (only two are shown in Figures 1 and 3). Solder a 4 inch wire between the anodes of CR0 and CR1 and solder a 4 inch wire between the anodes of CR2 and CR3. The wire used should be #14 or larger (the black wire).

( ) 17. Attach a crimp terminal to the wire from CR0 and CR1. Connect it to terminal #15 of the transformer.

( ) 18. Repeat above (#17) procedure for black wire from CR2 and CR3 and connect it to terminal #13 of the transformer.

( ) 19. Install and bolt heat sink (and diodes) onto the PSC board.

NOTE: WARNING!!! OBSERVE POLARITY

The 4 large capacitors will be destroyed if power is applied while they are installed backwards.

On the two large capacitors C0 and C1, the negative side of the capacitor bolts to the DC ground plane of the PSC board. The positive end of capacitors C0 and C1 bolts to the unregulated 8 volt plane of the PSC board.

( ) 20. Place lockwashers on four 10-32x3/8" screws, insert them from the underside of the board and mount capacitors C0 and C1.

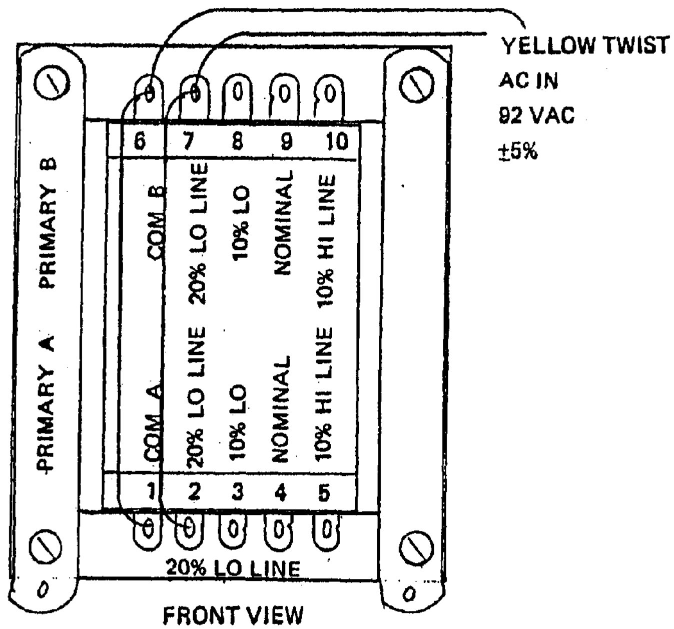
( ) 21. In a similar manner, mount C3 with the negative terminal bolted to the ground plane and positive terminal bolted to the +16 volt plane.

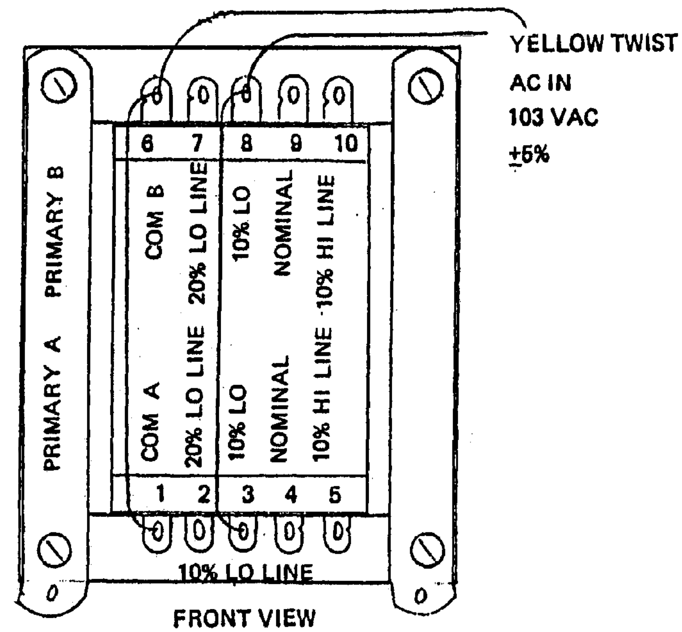
( ) 22. To install capacitor C2, bolt the positive terminal to the DC ground plane and the negative terminal to the negative (-16 volt) plane.

FAN INSTALLATION (OPTIONAL)

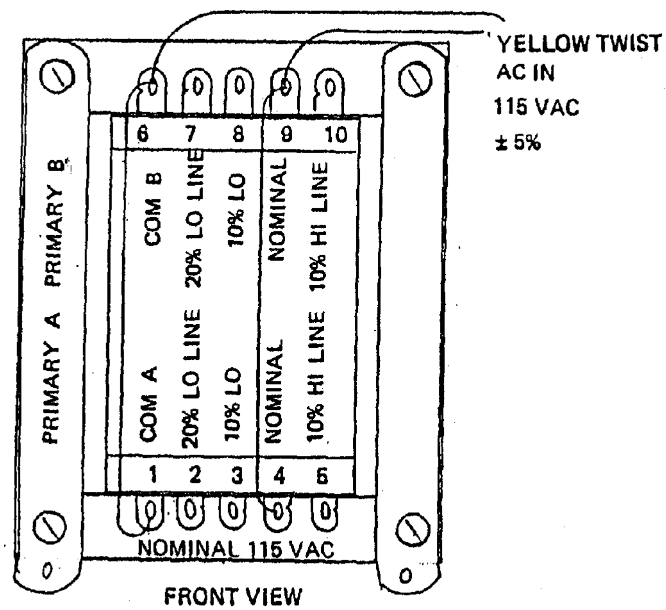
( ) 23. Attach the fan leads to lugs #6 and #9 or #1 and #4 to supply 115 VAC to the fan. This wiring is standard for all input AC wiring configurations.

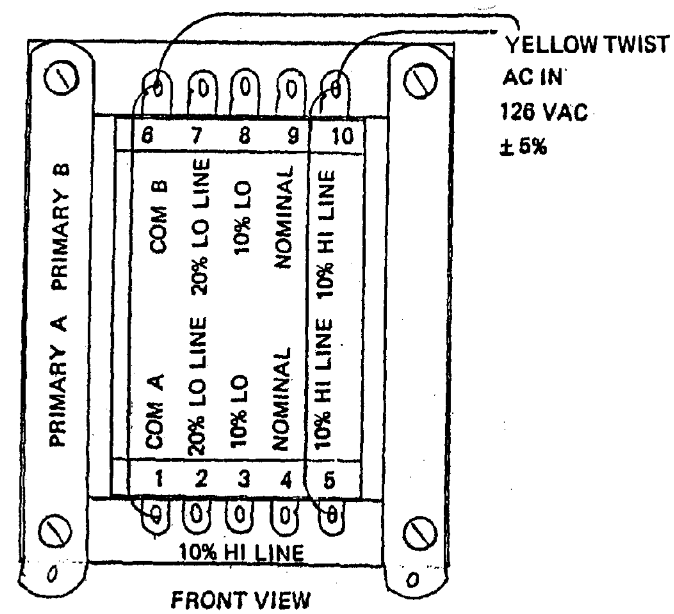
SEE MAINFRAME ASSEMBLY SECTION TO INSTALL POWER SUPPLY IN CHASSIS AND CONNECT TO MOTHER BOARD.



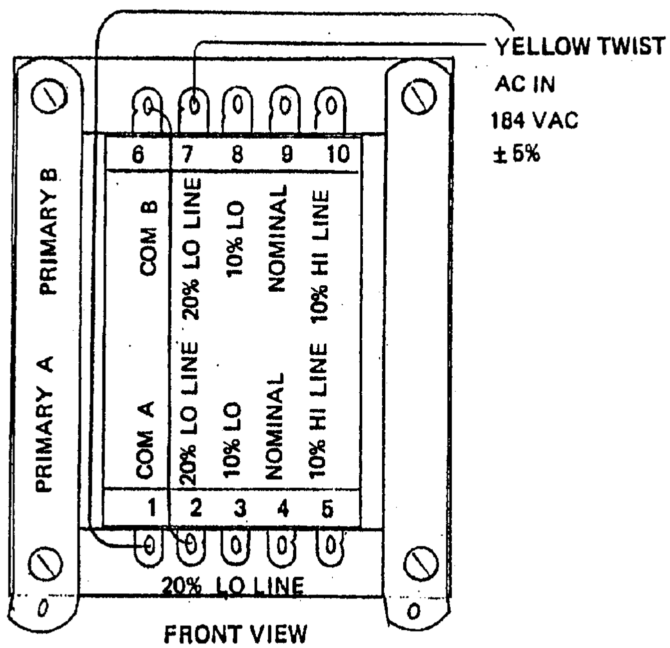


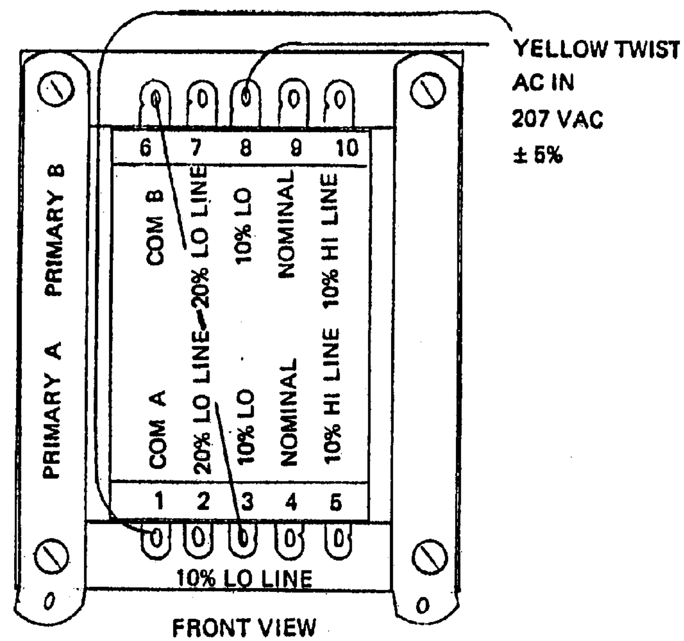
WIRING CHART: 92 – 103 VAC 60 Hz IN
Use next lowest line input taps when operating full chassis or on 50 Hz. See User Guide for more information. Use 5A fuse.



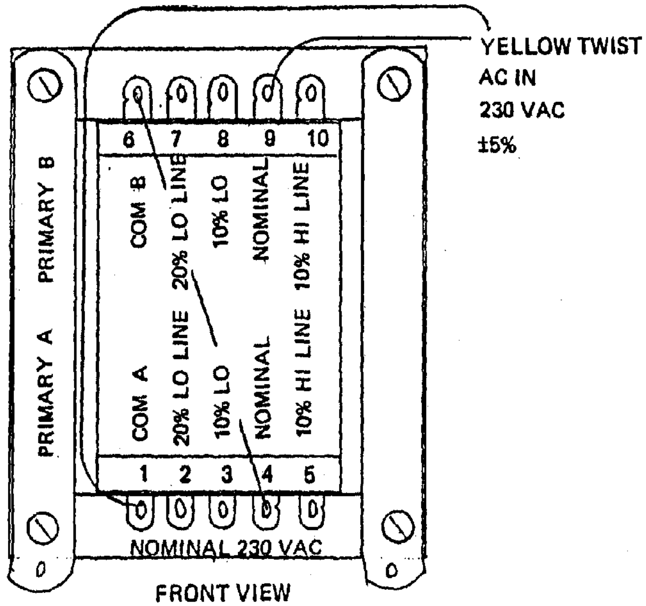


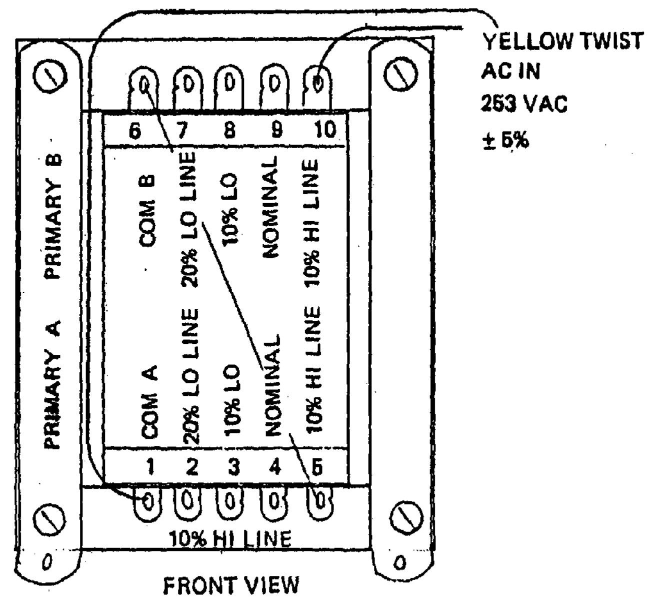
WIRING CHART: 115 – 126 VAC 60 Hz IN
Use next lowest line input taps when operating full chassis or on 50 Hz. See User Guide for more information. Use 5A fuse.



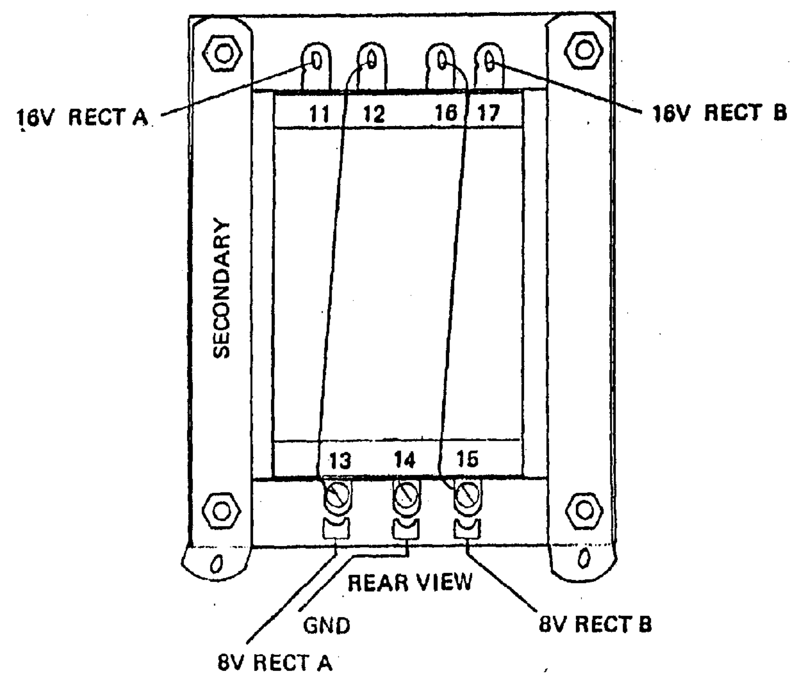


WIRING CHART: 184 – 207 VAC 60 Hz IN
Use next lowest line input taps when operating full chassis or on 50 Hz. See User Guide for more information. Use 2½A fuse.





WIRING CHART: 230 – 253 VAC 60 Hz IN
Use next lowest line input taps when operating full chassis or on 50 Hz. See User Guide for more information. Use 2½A fuse.



SECONDARY WIRING DIAGRAM

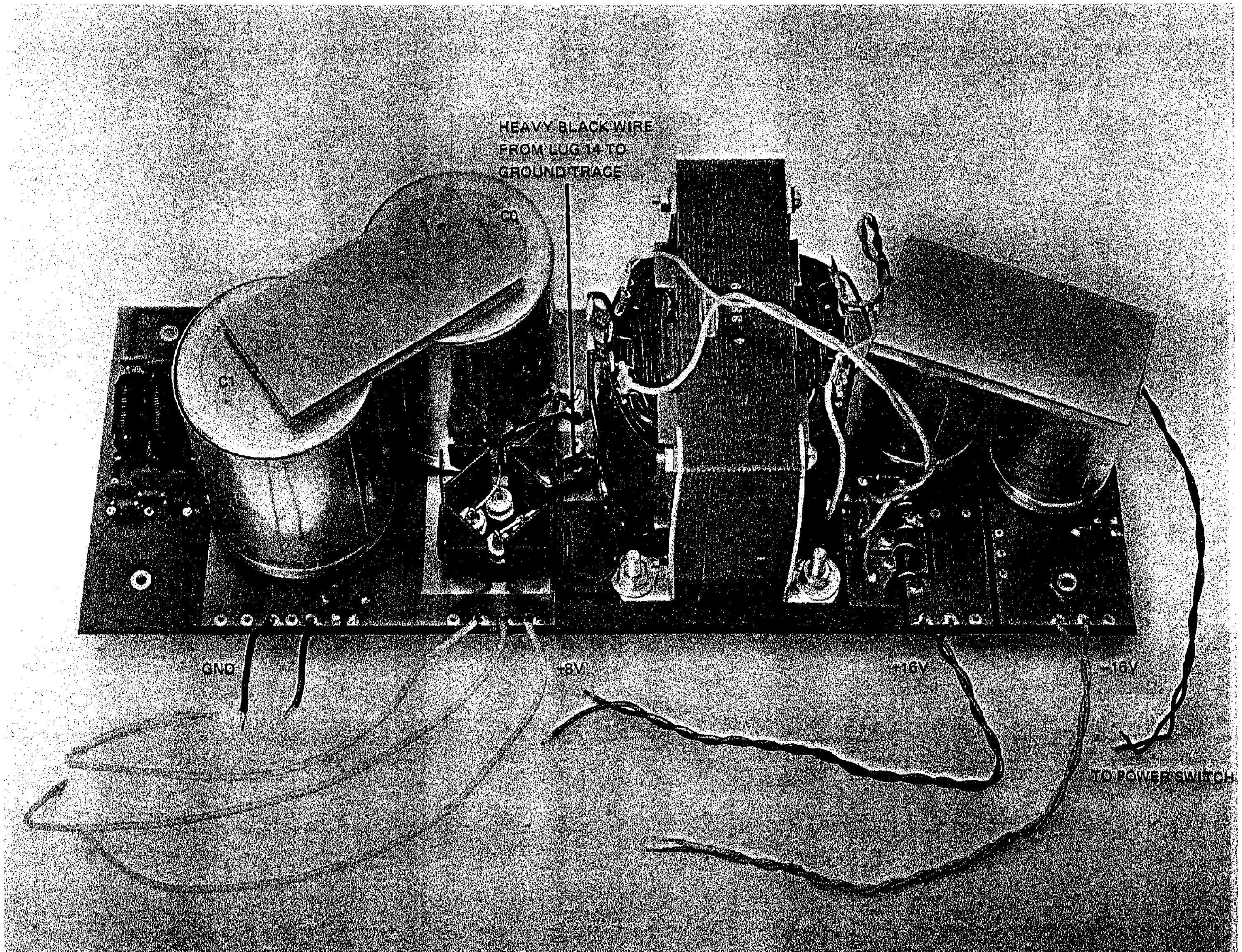


FIG. 1 PS-28U OVERALL

FIG. 2 PRIM [illegible] WIRING

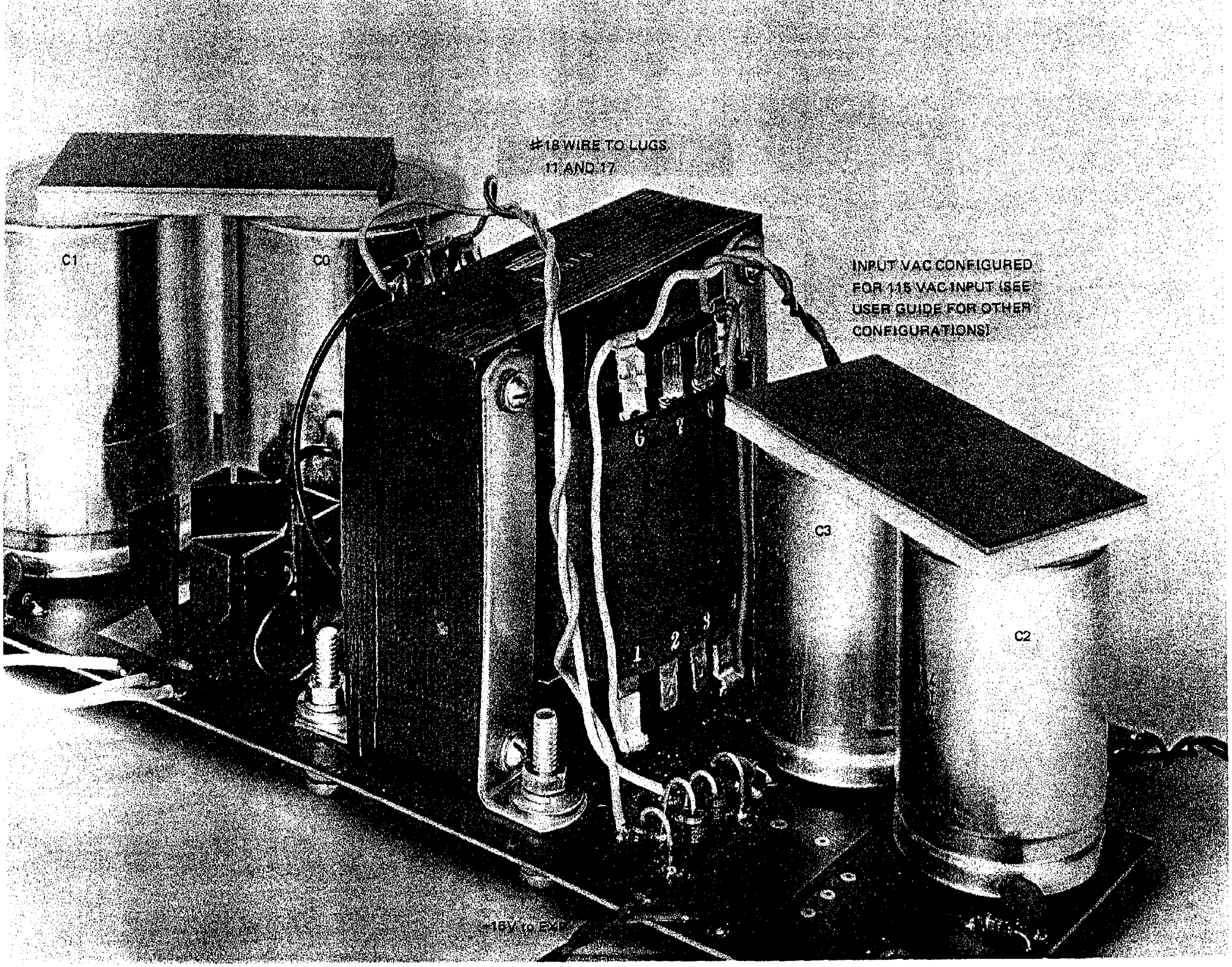
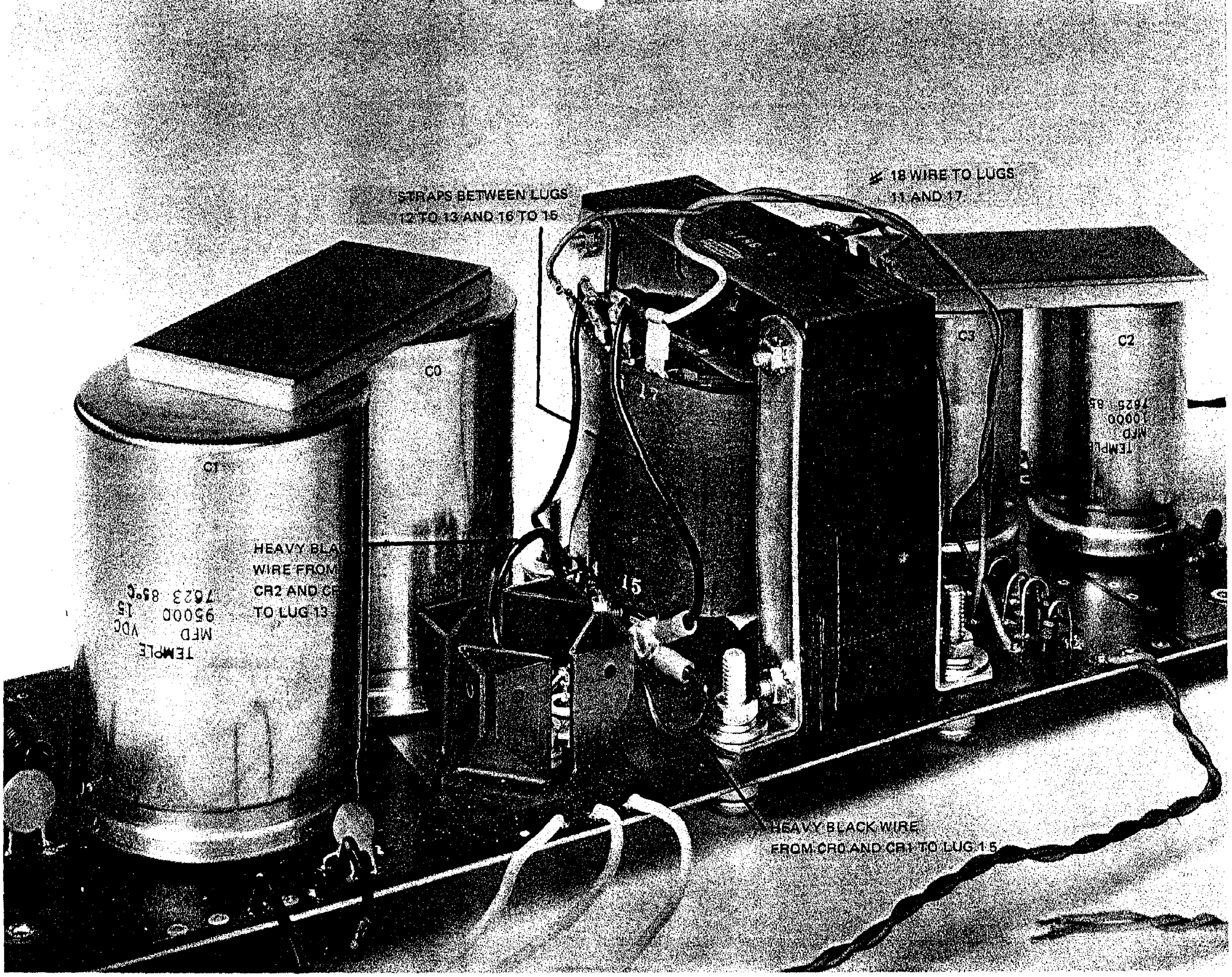
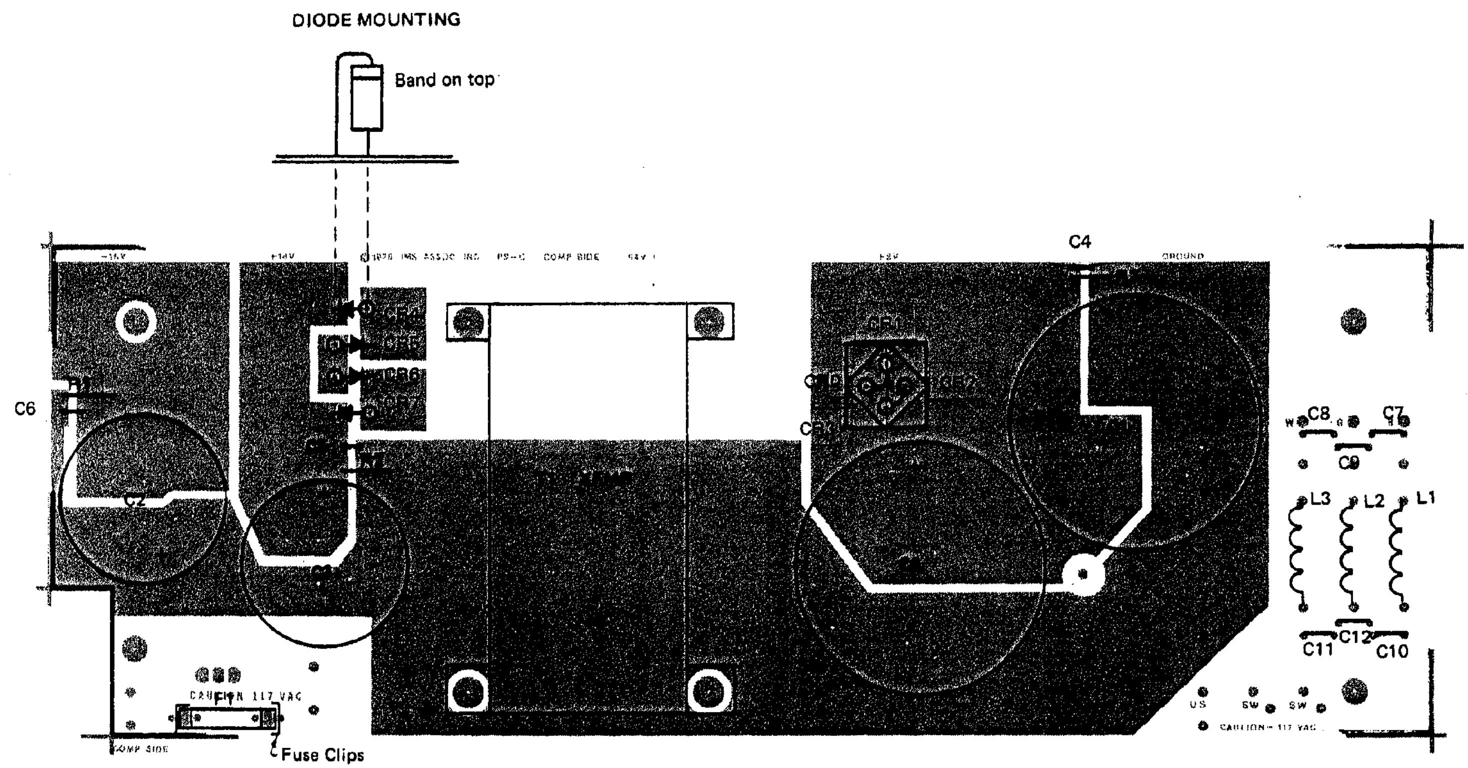


FIG. 3 SE Y WIRING



0	ORIGINAL	5/76
1	MODIFIED	1/77



ASSEMBLY DIAGRAM
PSC-U REV. 1 1/77

USER GUIDE

The PS-28U User's only option is the selection of a transformer primary tap. The transformer provides primary taps which allow selection at AC input voltages ranging from 92 - 126.5 and 184 - 253 VAC at 50/60 Hz.

As the PS-28U is an unregulated supply, the supply voltages are dependent on the load conditions. The user may adjust his/her loaded voltage by picking an appropriate primary tap, but should be careful that the no load voltages do not exceed +11, +18 and -18 volts. These maximums are selected so that the power dissipated in the system's voltage regulators and zener diodes does not exceed the device ratings. Similarly, the user should not allow the +8 supply to fall below 7.0 volts, the point at which the 7805 regulators cease to regulate.

It may be desirable when operating at 50 Hz or with a fully loaded chassis to select the next lower primary taps. This will increase the amount of current available. But, in all cases, the load voltages should not exceed the above levels. Also, the +8 supply should not fall below 7.0 volts.

Large currents require extremely low resistance paths from the power supply to the motherboard. It is suggested that #14 wire in multiple lengths be used to connect the power supply to the motherboard, and that all wires be only as long as necessary. Special care is required to insure low resistance solder connections; the +8 and ground leads are especially critical in this regard. Any significant loss in the supply wiring reduces the power available at the motherboard.

The power switch leads may be connected to the pads provided, or the user may mount a switch directly on the PSC board. Unswitched (marked US) and switched (marked SW) AC pads allow the connection of external equipment.

POWER SUPPLY

FUNCTIONAL DESCRIPTION

The IMSAI PS-28D Assembly is a modular unregulated power supply for the IMSAI 8080 System. It provides the basic unregulated +8, +16, and -16 system supply voltages. The PS-28D requires a 117 volt AC single-phase line input, and includes a line noise filter. 117 volt terminals, both switched and unswitched, are available for line powered options such as the ventilating fan and auxiliary power outlets on the back panel. A power switch location is provided on the PS-28D for use when a front panel is not installed in the system.

Physically, the PS-28D measures 16.5" x 5.75" x 5.5", (42 x 15 x 14 cm), and weighs 16 pounds (7.3 kg).

SPECIFICATIONS: PS-28D SUPPLY

Power Requirements:

110-120 volts AC, single phase, 500 Watts (maximum)

No load voltages - 117 V input, nominal taps (0, #3):

9.7 volts	+18.0 volts	-18.0 volts

Current Supplied - 117 volt input, resistive load:

28.0 amperes at +7.0 volts ripple valley

4.5 amperes at +13.5 volts ripple valley

4.5 amperes at -13.5 volts ripple valley

THEORY OF OPERATION

The PS-28D Assembly is an unregulated power supply that provides the basic +8, +16, and -16 voltages for the 8080 system. It is comprised of four major component assemblies: line filter, transformer, rectifiers, and filters.

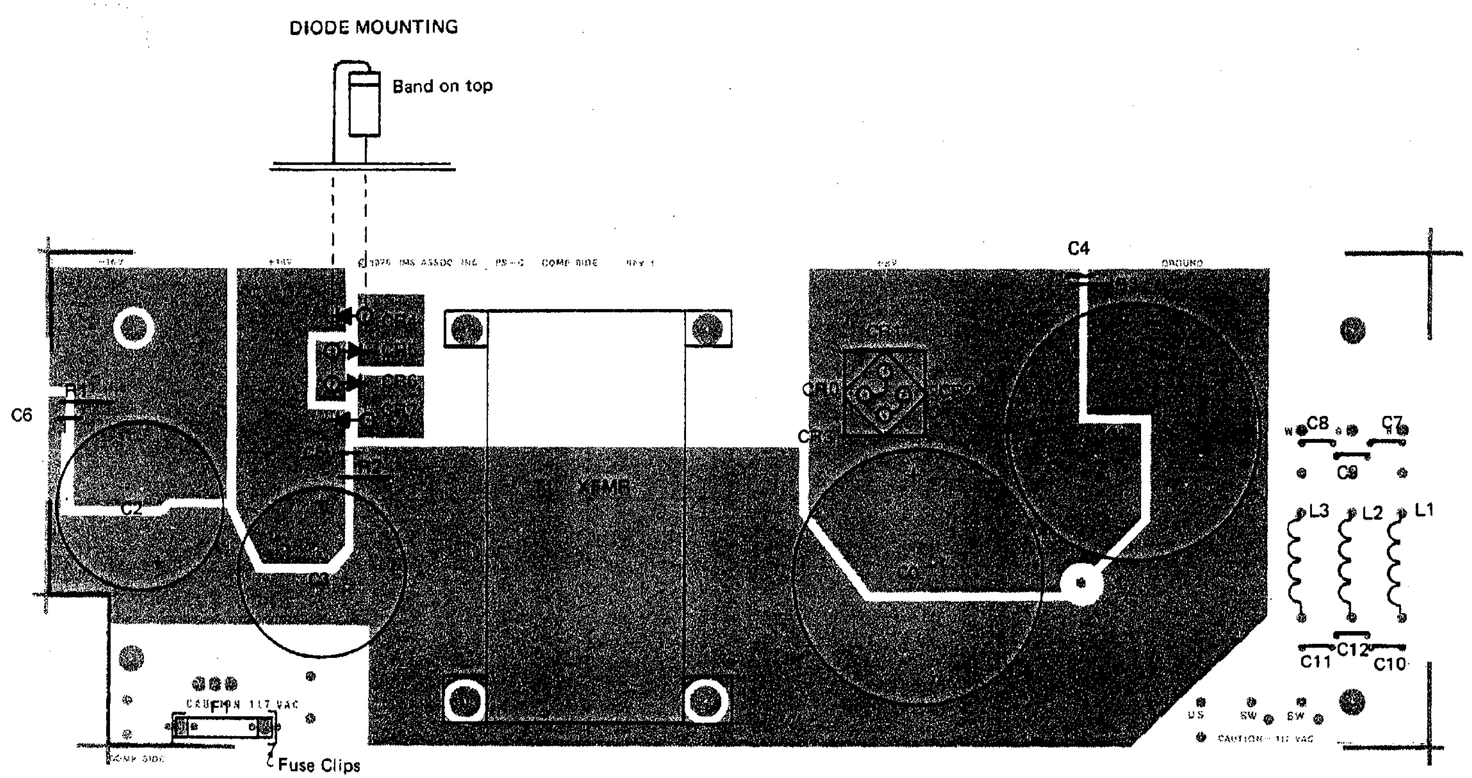
The line filter is a triple PI L-C filter designed to remove high frequency noise present on the AC line. This filter attenuates line noise above 1 MHz in frequency.

The transformer primary is designed for a nominal 117 volt line, and 0, -10%, and +10% taps allow for adjustment to line voltage variations. The transformer secondary is connected as three series winding with a center tap. Four MR 1121 diodes full-wave rectify the +8 volts, while a full-wave bridge of four MR601 diodes rectify the ±16 volts.

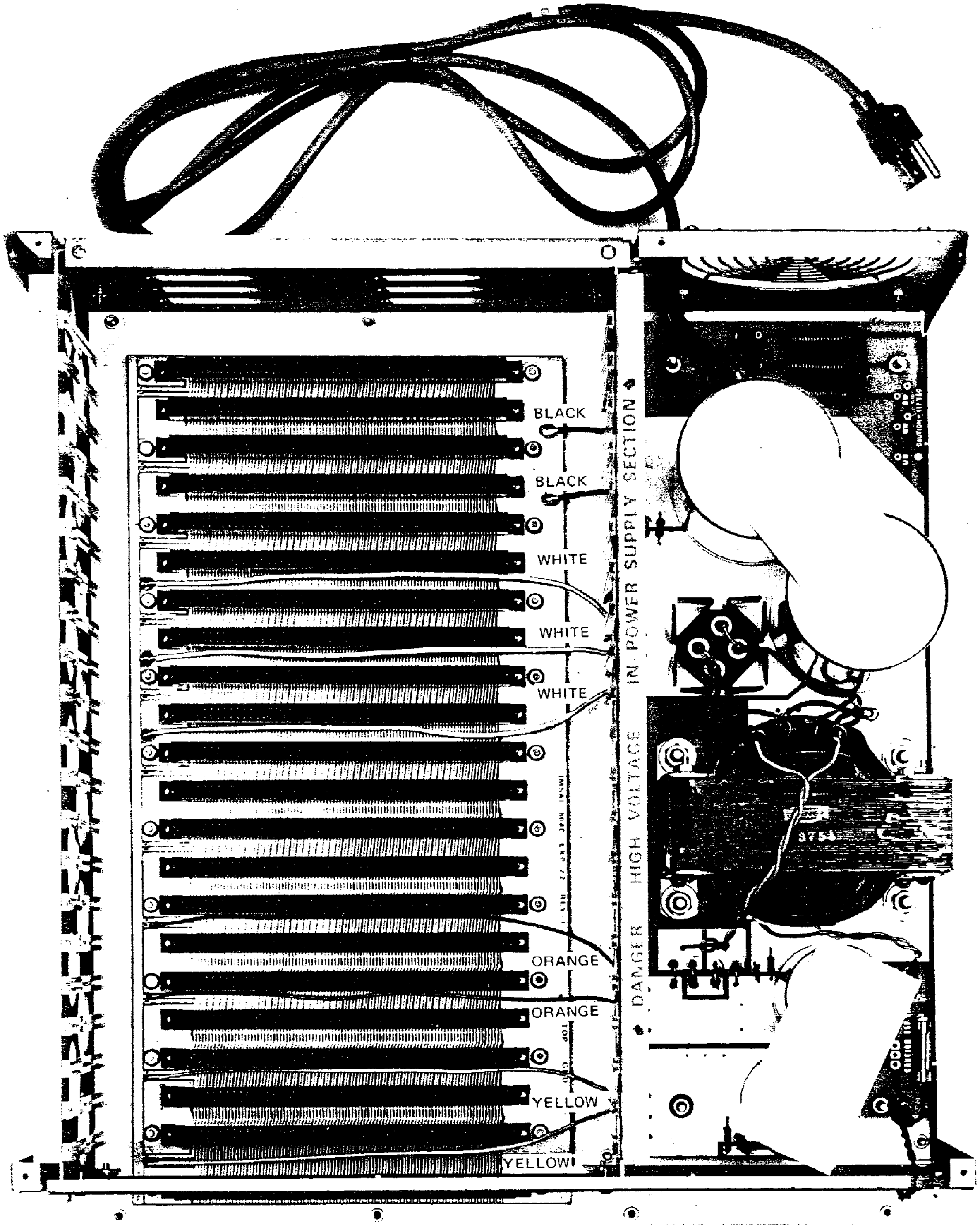
The ±16 volt supplies are each filtered by a 10K uF capacitor to ground, providing +15 average volts at 4.0 amps. The +8 volts is filtered by two 95K uF capacitors to ground, providing 7.3 average volts at the 28 amp rated current.

.luF capacitors bypass each voltage supply, and bleeder resistors discharge the filter capacitors when power is turned off.

0	ORIGINAL	5/76
1	MODIFIED	1/77



ASSEMBLY DIAGRAM
PSC–D REV. 1 1/77



IMSAI 8080 Overview of Chassis, EXP-22 and PS-28

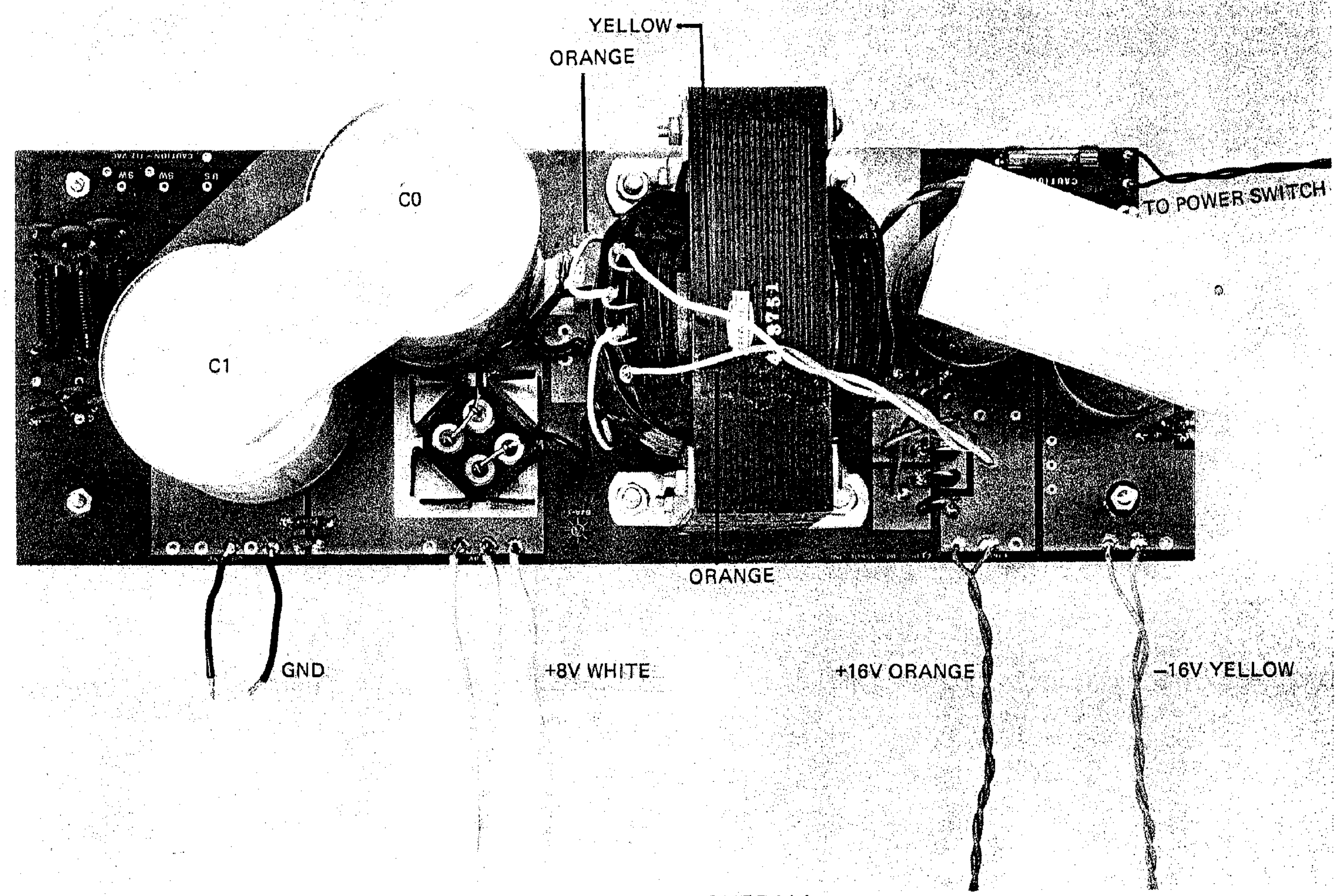


FIGURE 1 PS-28D OVERALL

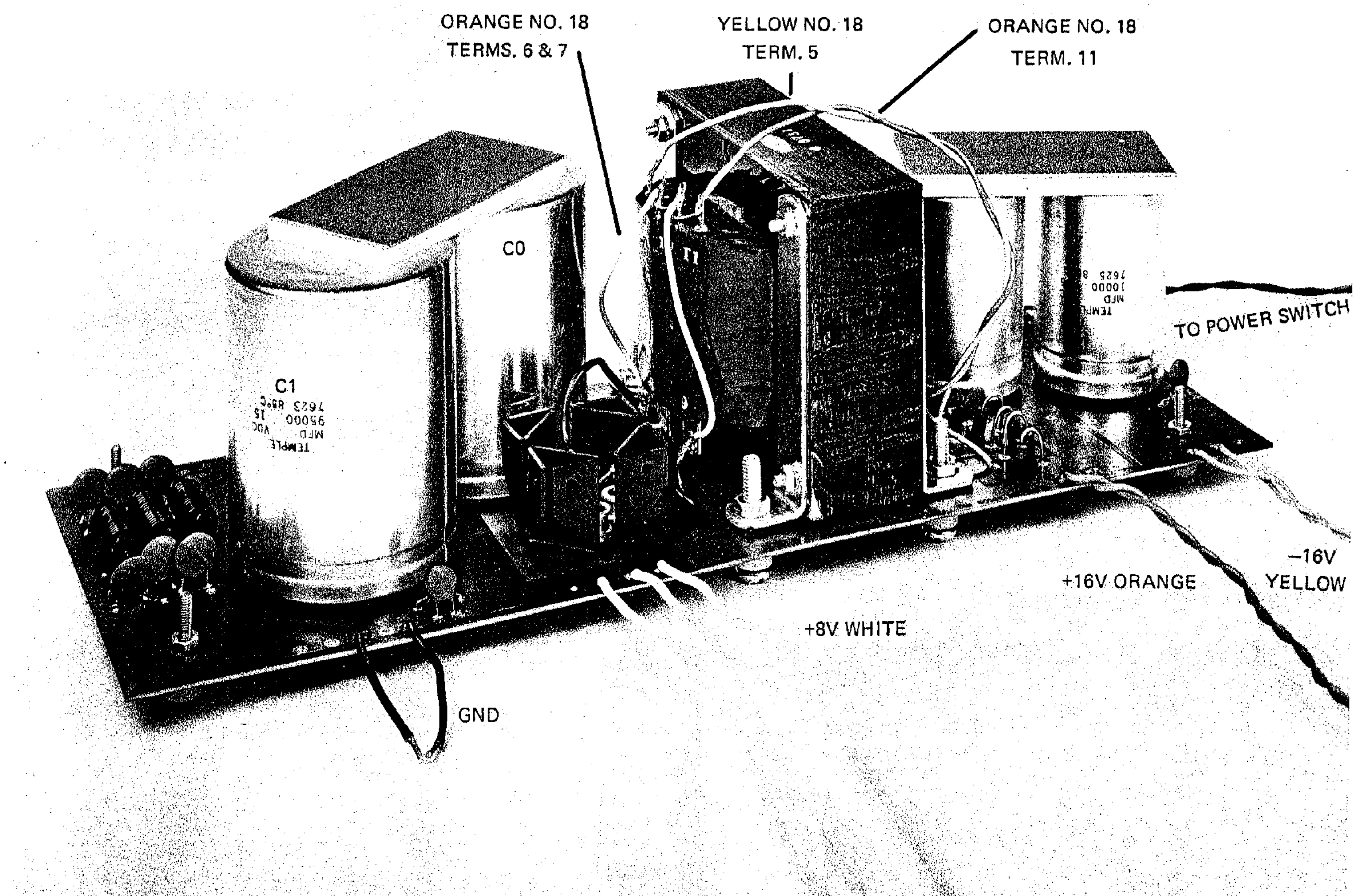


FIGURE 2 WIRING DIAGRAM

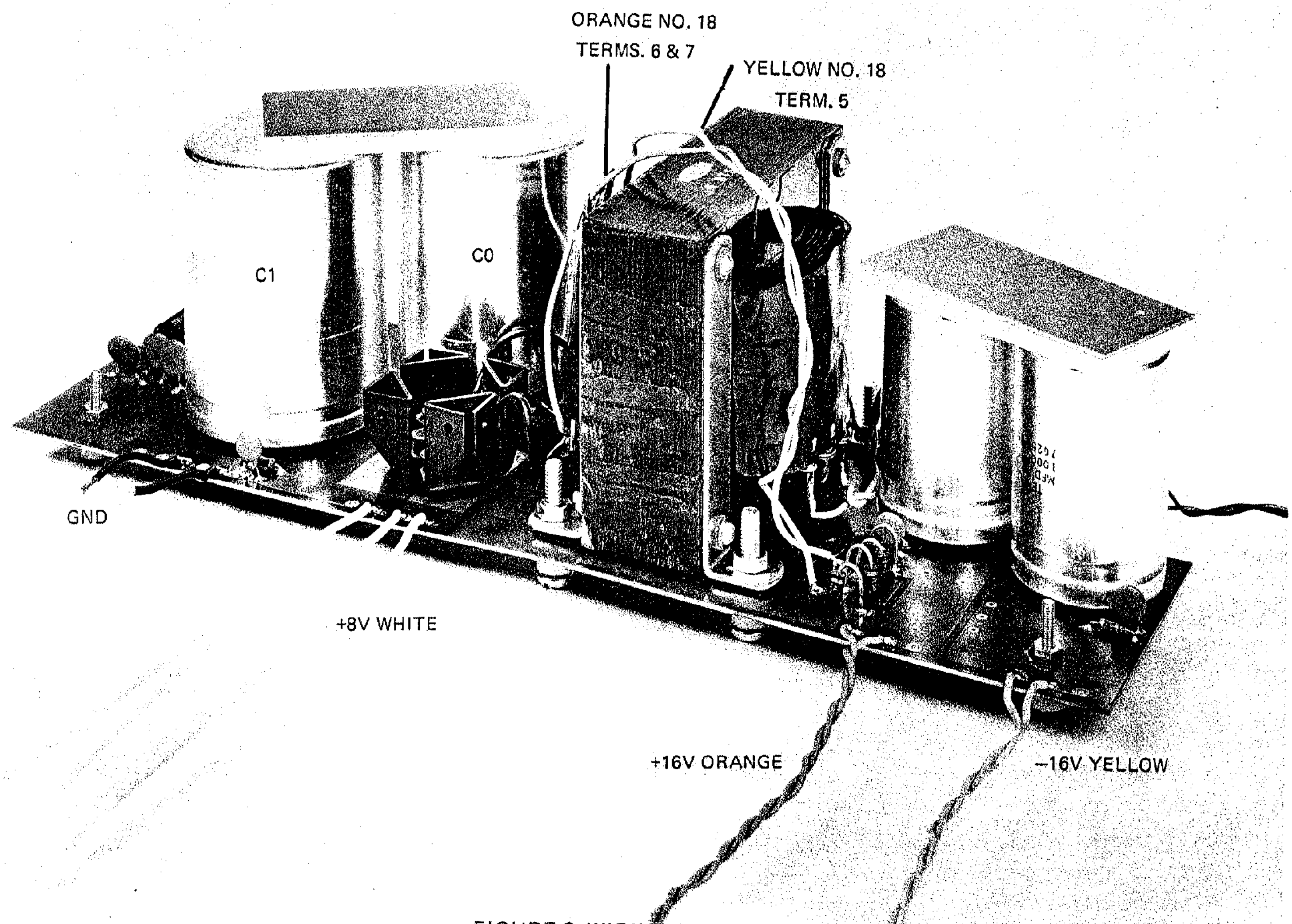


FIGURE 3 WIRING DIAGRAM

BOARD: PS-C

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Solder	15-0000001	5'	
Heat Sink	16-0100006	1	Wakefield, 690-220-P (Modified)
Screw	20-3402001	4	6-32x3/8" Phillips Pan Head CAD Machine
Screw	20-3702001	4	6-32x3/4" Phillips Pan Head CAD Machine
Screw	20-4901001	5	8-32x1¼" Binding Head CAD Machine
Screw	20-5402000	8	10-32x3/8" Binding Head CAD Machine
Screw	20-6901001	4	¼-20x1½" Binding Head CAD Machine
Nut	21-3120001	8	6-32 CAD Hex Nut
Lockwasher	21-3350001	8	#6 CAD Internal Star Lockwasher
Nut	21-4120001	5	8-32 CAD Hex Nut
Lockwasher	21-4350001	5	#8 CAD Internal Star Lockwasher
Spacer	21-4600002	5	8-32x½" Nylon Threaded Spacer, H H Smith
Nut	21-5120001	4	10-32 CAD Hex Nut
Lockwasher	21-5320001	4	#10 CAD Split Ring Lockwasher
Lockwasher	21-5350001	8	#10 CAD Internal Star Lockwasher
Nut	21-6120001	4	¼-20 CAD Hex Nut
Washer	21-6310001	4	¼"x1/16" CAD Flat Washer
Lockwasher	21-6320001	4	¼" CAD Split Ring Lockwasher
Washer	21-6390001	4	¼"x1/16" Nylon Washer, H H Smith 2664
Spacer	21-6600001	4	¼-20x½" Nylon Internal Thread Spacer
Wire	22-1014001	48"	14 AWG, White, Alpha 1559 14-41/30 PVC
Wire	22-1014002	48"	14 AWG, Black, Alpha 1559 14-41/30 PVC
Wire	22-1018001	60"	18 AWG, Orange, Gavitt 8522

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Wire	22-1018002	60"	18 AWG, Yellow, Gavitt 8522
Wire	22-5018001	12"	18 AWG, Twisted Pair, Yellow/Orange, Stranded and Insulated
Cord	22-6000001	1	Line Cord, Belden 17239
Bushing	24-0600001	1	Strain Relief Bushing Grommet
Transformer	29-0100001	1	Tranex 4-3751 or Equiv.
Inductor	29-0200001	3	8uH, 5 Amp, Airco Speer 025834-001K
Resistor	30-3470462	1	470 Ohm, ½ Watt/yellow, violet, brown
Resistor	30-4100462	2	1K Ohm, ½ Watt/brown, black, red
Capacitor	32-2004010	6	.04uF, 500V Disk Ceramic (or .01uF, 1000V)
Capacitor	32-2010010	3	.1uF, 30V Disk Ceramic
Capacitor	32-2510060	2	10K uF, 25V Electrolytic
Capacitor	32-2595060	2	95K uF, 15V Electrolytic
Fuse	33-0100003	1	Bussman Fusetron MTH 5 5 Amp
Fuse Clip	33-0200001	2	Fuse Clip, 102068
Fan Guard	34-0200001	1	
Rectifier	35-1000002	4	MOT MR 1121
Diode	35-1000003	4	MOT MR 501
PC Board	92-0000024	1	PS-C Rev. 1
Label	93-0000001	1	Voltage/Frequency Label Plate

General

1) Unpack your board and check all parts against the parts lists enclosed in the package.

Component Installation

2) Insert and solder each of the two 1K ohm, ½ watt resistors (brown/black/red) at locations R1 and R2 as shown on the Assembly Diagram.

3) Insert and solder the one 470 ohm, ½ watt resistor (yellow/violet/brown) at location R3 as shown in the Assembly Diagram.

4) Insert and solder each of the three .1uF capacitors at locations C5, C6 and C4 as shown on the Assembly Diagram.

5) Next bend each of the cathode leads on each of the four rectifier diodes CR4, CR5, CR6 and CR7 as shown in Figure 3. Insert the anode end of the diodes down as shown in Figure 3 and solder.
NOTE: See Assembly Diagram for diode mounting position.

6) Insert and solder each of the six .04uF capacitors C7 through C12 in the locations shown on the Assembly Diagram.

7) Insert and solder each of the three AC filter inductors L1, L2 and L3 in locations as shown on the Assembly Diagram.

8) Insert and solder each of the two fuse clips in locations as shown on the Assembly Diagram. See Figure 1 for clarification. Snap in the fuse.

Transformer Wiring

9) Solder a yellow and an orange wire to the transformer lugs numbers 1 and 3 for AC. Wires should be twisted together. Solder twisted pair ends to the PSC board as shown in Figure 1.

10) Attach an orange wire between lugs 6 and 7 and a yellow wire between lugs 10 and 9. Solder the wire at terminals 6 and 10 only.

11) Next connect a yellow and an orange #18 wire to terminal numbers 5 and 11 respectively. Twist and run the wires across the transformer to the anode pads of the CR4 and CR7. Solder.

12) Temporarily install the transformer with ¼" hardware. Terminals 1, 2, 3 and 4 should face towards the fuse end of the PSC board.

NOTE: Scrape or sand lugs 7, 8, and 9 of the transformer to facilitate solder connections of #14 wires.

13) Take a short, black wire (#14 or larger) and solder to terminal 8 on the transformer.

14) Solder the other end of the short wire used above to the ground trace below terminal number 8 (see Figure 1 for clarification).

Heat Sink Installation

NOTE: Keep all wiring as short as possible. An extra two inches of #14 wire will reduce the current capacity of the Power Supply.

15) Cut a 4½" length of #14 (black) wire. Strip 1" of insulation from one end and ¼" of insulation from the other. Insert the 1" bare wire end through the anode terminals of diodes CR0 and CR1. Then solder the anode terminals CR0 and CR1. Crimp the remaining end of this lead to transformer terminal 9. Reference the board photos to verify correct orientation.

16) Cut a 3½" length of #14 (black) wire. Strip 1" of insulation from one end and ¼" of insulation from the other. Insert the 1" bare wire end through the anode terminals of diodes CR2 and CR3. Then solder the anode terminals CR2 and CR3. Crimp the remaining end of this lead to transformer terminal 7. Reference the board photos to verify correct orientation.

17) Solder transformer terminal 9, which should now support the black wire from CR0-CR1 and the yellow wire from terminal 10. Solder transformer terminal 7, which should now support the black wire from CR2-CR3 and the orange wire from terminal 6.

18) Install and bolt heat sink (and diodes) onto the PSC board.

NOTE: The schematic of the PSC shows the CR0-CR1 wire and the CR2-CR3 wire going to transformer terminals 7 and 9 respectively. While this configuration is acceptable, the configuration resulting from steps 15-17 provides for shorter lead lengths and should be adhered to whenever possible.

NOTE!! WARNING!! OBSERVE POLARITY

The large capacitor will be destroyed when power is applied if they are installed backwards.

On the two large capacitors C0 and C1, the negative side of the capacitor bolts to the DC ground plane of the PSC board. The positive end of the capacitors C0 and C1 bolts to the unregulated 8 volts plane of the PSC board.

19) Place lockwashers of four 10-32x3/8" screws. Insert them from the underside of the board and mount capacitors C0 and C1.

20) In a similar manner, mount C3 with the negative terminal bolted to the ground plane and positive terminal bolted to the +16 volt plane.

21) To install capacitor C2, bolt the positive terminal to the DC ground plane and the negative terminal to the negative (-16 volt) plane.

22) Attach the capacitor brace plates to the bases of the large capacitors with the adhesive backed foam tope located on one side of the brace plate.

SEE MAINFRAME ASSEMBLY SECTION TO INSTALL POWER SUPPLY IN THE CHASSIS AND TO CONNECT TO THE MOTHER BOARD.

IMSAI

CP-A

CP-A

FUNCTIONAL DESCRIPTION

The CP-A board is the operator's panel for the IMSAI 8080 System. It includes operator switches, indicator lights and all logic necessary to operate the IMSAI 8080 System.

The panel is completely self-contained and plugs into the back plane's 100 pin edge connector. With this design it is not necessary to mount the CP-A at the front of the cabinet. Instead, the board can be plugged (via an extender card) into any available slot in the back plane.*

A full set of 16 address switches and 6 control function switches accept operator control and input. LED indicators are provided for the 16 bit address bus, the 8 bit data bus, the 8 bit status byte (control indicators for INTERRUPTS, ENABLED, RUN, WAIT and 8 bits of programmed output.

The CP-A board contains the logic necessary to drive the 8 programmed output indicators and the logic needed to read an 8-bit input byte from the high-order address switches. The DATA BUS indicators are run from the bi-directional portion of the data bus (via a flat cable to the MPU board) and show data either being read or written by the 8080 processor.

The indicators on the panel are wide-angle-view light emitting diodes mounted behind a contrast-enhancing acrylic panel assembly. All indicators and switches are explicitly marked. The photographically produced labels are very clear, protected by clear acrylic, and can never wear off. Bit positions are numbered, and binary bit values are labeled for both hexadecimal

*The switches are included on the front panel whether it is mounted in the front of the panel or not.

and octal formats. Special labels may be easily inserted to identify special functions for the programmed output port. Switches on the panel are high-quality paddle switches, and are color-coded for easy and error free use.

For situations in which it is not desired to locate the operator's panel at the cabinet front (such as use of the IMSAI 8080 as a dedicated controller), the CP-A front panel may be inserted (via extender card) into any back plane slot. In this arrangement, programs may be easily tested and debugged without time-consuming mounting and un-mounting of the front panel. For these applications, the front slot of the machine can be reserved for the parallel I/O board with its LED indicators showing through the front panel mask.

THEORY OF OPERATIONS

The CP-A front panel assembly provides machine status indicators, user controlled switches, and control functions to the IMSAI 8080 operator. The CP-A board communicates with the MPU-A microprocessor and other boards through the 8080 back plane and, additionally, connects (via 16 conductor flat cable) to the bi-directional data bus of the 8080 microprocessor.

The CP-A panel uses 44 Light Emitting Diodes as front panel indicators. Many of these indicators directly correspond to signal levels on the IMSAI 8080 back plane, and are driven directly from the bus with no intervening logic. Indicators in this group are the 16 Address Bus LED's, the 8 STATUS byte LED's, the INTERRUPT ENABLED LED, the WAIT LED, and the HOLD LED.

The 8080 microprocessor chips bi-directional data bus levels (provided by a 16-conductor cable) are displayed on the DATA bus indicators via the 74LS04 (low power schottky hex inverter) sections. Also driven from the bi-directional bus is the 8212 8 bit latch used to drive the PROGRAMMED OUTPUT indicators. The RUN indicator is driven directly from the run/stop flip-flop (74107) on the CP-A Board.

The 16 ADDRESS-PROGRAMMED INPUT and ADDRESS-DATA switches allow the operator to place desired value (program, data, addresses) on the 8080's bi-directional bus.

As shown on the schematic, these switches connect 7405 (open collector) inverters to the bus in a wired- AND configuration.

Pullup resistors on the MPU Board ensure that the bus levels are all high unless any inverter on any one of the bus lines goes low. Thus, if an inverter goes low, (this condition will be discussed shortly) the address switch can be used to put either a high or low value on that line.

The function switches provide the operator with direct control of the microprocessor. The RUN/STOP switch controls the X-READY line via the RUN/STOP flip-flop.

If the switch is set to RUN, on the next falling edge of the Phase II clock, the RUN and X-READY lines are set high. If the switch is set to STOP, the high STOP value and the Phase II clock are NANDed (U16) and this value NANDed with the DATA OUT 5 bit (fetch/status) and the PROCESSOR SYNC line.

Thus, when the processor is fetching a new instruction, the RUN/STOP flip-flop will be reset, the processor X-READY line goes low, and the processor stops.

Several CP-A function switches operate by providing the 8080 with an instruction, executing the instruction, and then stopping the processor on the next cycle. The open collector 7405's and support gating put these instructions on the 8080's bi-directional bus. The EXAMINE function uses a jump instruction (hex C3) followed by two bytes of the address selected on the front panel switches.

This operation causes the processor to jump to the selected address and, then, the processor is stopped during the next cycle. When stopped, the processor was reading the selected byte from memory as if it were going to execute it. Therefore, the processor stops with the desired address displayed on the address bus and the contents of that address is displayed on the data bus.

If the RUN switch is operated at this time, the processor will continue to pull the selected byte from memory and execute it.

The EXAMINE NEXT and DEPOSIT NEXT switches use similar schemes and the NO-OP (hex 00 or octal 000) instruction to increment the address.

Much of the remaining logic of the CP-A is used to sequence these commands to provide the desired functions. The RUN/STOP flip-flop line, the SINGLE STEP line, the EXAMINE line, and the EXAMINE NEXT line are all input to an OR-gate controlling the X-READY line. (The X-READY line must be high for the processor to run. Its function is identical to the P-READY line used by the memory and I/O boards. The X-READY line is reserved for use of the front panel to avoid conflicts of two gates driving the same backplane line). During each of these functions, the processor is permitted to execute an instruction, and then is stopped in the next cycle in a manner similar to the RUN/STOP flip-flop cycle described earlier.

For the SINGLE STEP function, a one-shot, triggered by the SINGLE-STEP switch, is used to produce a pulse and the trailing edge of that pulse is used to set a flip-flop which controls the SINGLE STEP line. This permits the processor to execute the present instruction. The SINGLE STEP flip-flop is reset by the occurrence of the sync pulse on the following instruction, thus causing the SINGLE STEP level to be removed and the processor to stop on the following cycle.

The EXAMINE-NEXT flip-flop is similarly controlled by the leading edge of a pulse from a one-shot driven by either the DEPOSIT NEXT or EXAMINE NEXT switch. The output of the flip-flop is used both to put the NO-OP (hex 00 or octal 000) onto the bi-directional data bus, and also to provide the READY signal so that the processor will execute the instruction. It is reset by the sync pulse on the following cycle, thus stopping the processor again.

The EXAMINE function involves a 4-step sequence produced by two flip-flops arranged as a counter. The pulse produced by the EXAMINE switch's one-shot starts the counter and on the first count, the jump instruction is inserted on the data bus. On successive counts of the two bit counter, the lower and upper address byte are inserted on the data bus in turn, and on the 4th count (that is, when the counter is back to 0), the processor is again stopped by the removal of the READY line. Thus, the EXAMINE logic provides the processor with the jump instruction and the two address bytes that the processor expects after a jump instruction and stops the processor during the fetch of the designated memory byte.

Similarly, the DEPOSIT switch, when operated, produces a pulse from the DEPOSIT one-shot which is buffered to the MEMORY WRITE line on the backplane. The leading edge of this pulse also starts a second one-shot with a much longer period which puts the data from the data switches on to the data bus for the duration of the longer pulse. The DEPOSIT one-shots are triggered either by the operation of the DEPOSIT switch or by the trailing edge of the DEPOSIT NEXT one-shot so that the DEPOSIT function will operate at the end of the EXAMINE NEXT cycle.

The 7427 gate in U15.5 is used to insure that during the time the front panel is inserting any information on the bi-directional data bus, the MPU-A board's bi-directional data bus driver is not also trying to drive the bus at the same time.

The inputs to this gate are the DATA-ON line, the EXAMINE NEXT line and the EXAMINE line. These are the three functions during which the front panel is transferring data or instructions to the bus.

The inputs to the 7405 open-collector inverter bus drivers are the lines NO-OP, C3, HAD, and LAD. These levels are ANDed with the PDBIN signal so that the information appears on the bus during the time the processor is expecting to see it there.

The input port from the high order address switches is implemented simply by decoding the address FF and ANDing it with the DBIN signal so that switch values appear on the data bus during the time that the processor is expecting information from the port FF.

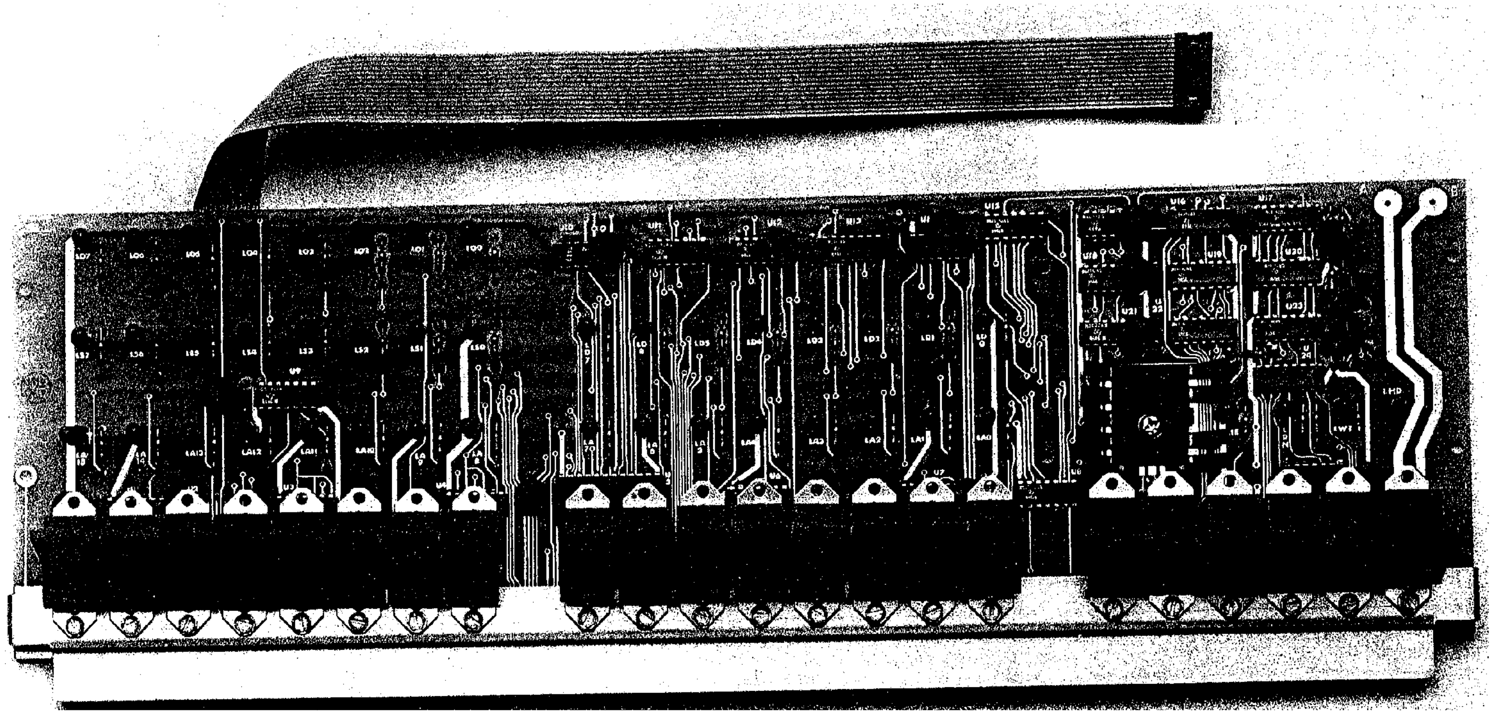
The same address decode signal is ANDed with the STATUS OUT line to enable the 8212 8 bit latch which drives the PROGRAMMED OUTPUT indicators. The information on the bi-directional data bus is then latched onto the output port at the time of the processor write strobe.

The STATUS WORD DISABLE line (SSWDSB, Pin 53 backplane) is gated to insure that no conflicts are created between the bi-directional bus drivers on the MPU and CP-A boards. This signal is controlled by the same gating that places the high order address switch values on the data bus for a front panel (address hex FF) read.

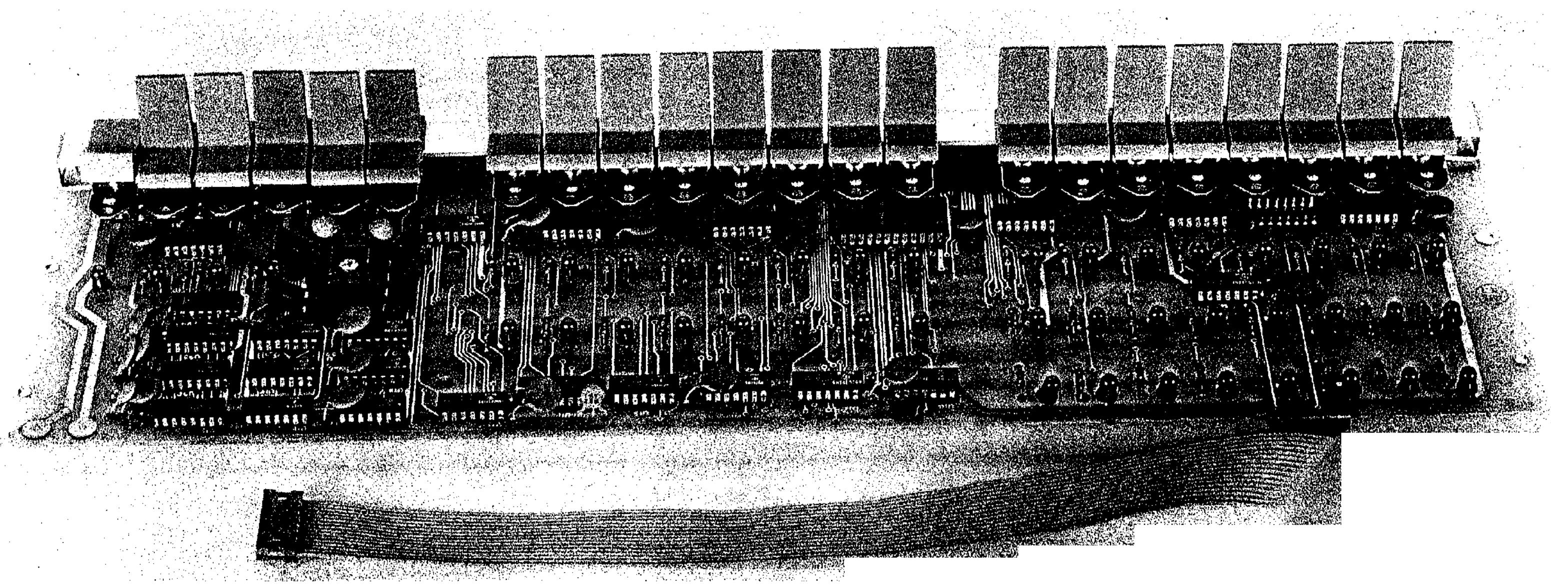
The STATUS WORD DISABLE line, Pin 53 in the backplane, is also run by the signal which puts the high order address switches onto the data bus for the port FF read instruction so that the bi-directional data bus is not being driven by the bi-directional drivers on the MPU board at the same time that the front panel is inserting the switch information on the data bus.

The RESET switch directly grounds the RESET line on the backplane which is detected by the MPU board and processed to form a RESET pulse which re-appears on the backplane as a Power On Clear.

When the RESET switch is thrown to EXTERNAL CLEAR, the switch directly grounds the EXTERNAL SWITCH line on the backplane. There is a diode between the RESET line and the EXTERNAL CLEAR line so that during a reset operation an EXTERNAL CLEAR is also generated.



CP-A REV. 4



CP-A REV. 4

BOARD: CP-A

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Solder	15-0000001	10'	
Heat Sink	16-0100002	1	Thermalloy/6106B-14
Screw	20-2203001	22	4x¼" Slotted Hex Head, Self-Tapping, Type A Sheet Metal
Screw	20-3203001	2	#6x¼ Self-Tapping Sheet Metal
Screw	20-3302001	1	6-32x5/16" Phillips Pan Head Machine
Screw	20-3916002	8	6-32x1¼" Button Head Allen Machine, Black
Nut	21-3120001	1	6-32 Hex
Lockwasher	21-3350001	1	#6 Internal Star Lockwasher
Spacer	21-3600001	8	#6x¼" White Nylon
Spacer	21-3600002	8	7/16" White Nylon
Switch	26-1500001	8	Blue Paddle Switch, on/none/on
Switch	26-1500002	8	Red Paddle Switch, on/none/on
Switch	26-1500003	2	Red Paddle Switch, momentary
Switch	26-1500004	3	Blue Paddle Switch, momentary
Switch	26-1600001	1	Red Rocker Switch, on/none/on
Resistor	30-3220362	44	220 Ohm, ¼ Watt/red, red, brown
Resistor	30-4100362	9	1K Ohm, ¼ Watt/brown, black, red
Resistor	30-5470362	6	47K Ohm, ¼ Watt/yellow, violet, black
Resistor	30-6270362	1	270K Ohm, ¼ Watt/red, violet, yellow
Capacitor	32-2000110	2	.001uF Disk Ceramic
Capacitor	32-2001010	1	.01uF Disk Ceramic
Capacitor	32-2010010	17	.1uF Disk Ceramic

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Capacitor	32-2233070	2	33uF Tantalum
Diode	35-1000006	1	Signal Diode/1N914
LED	35-3000001	44	Light Emitting Diode/red
8T97	36-0089701	1	Hex Tri-State Buffer/N8T97B
7400	36-0740001	2	Quad 2 Input NAND/SN7400N
7402	36-0740201	2	Quad 2 Input NOR/SN7402N
7404	36-0740401	1	Hex Inverter/SN7404N
74LS04	36-0740402	2	Hex Inverter (Low Power Schottky)/SN74LS04N
7405	36-0740501	5	Hex Inverter Open Collector/SN7405N
7410	36-0741001	1	Triple 3 Input NAND/SN7410N
74LS10	36-0741002	1	Triple 3 Input NAND (LPS)/SN74LS10N
7427	36-0742701	1	Triple 3 Input NOR/SN7427N
7430	36-0743001	1	8 Input NAND/SN7430N
74LS30	36-0743002	1	8 Input NAND (LPS)/SN74LS30N
7805	36-0780501	1	5V Positive Volt Regulator/MC7805CP
8212	36-0821201	1	I/O Port/P8212/S1002
74107	36-7410701	3	Dual J-K Flip Flop With Clear/SN74107N
74123	36-7412301	3	Dual Monostable Multivibrator, Retriggerable with Clear/SN74123N
Cable Assembly	91-0400001	1	Cable K Assembly
PC Board	92-0000002	1	CP-A, Rev. 4
Plastic Panel	93-3010006	1	Clear Plastic Panel
Plastic Panel	93-3010007	1	Red Plastic Panel
Bracket	93-3010011	1	Switch Bracket
Photo Mask	93-3010015	1	
Paper Backing	93-3010016	1	

CP-A Rev. 4

ASSEMBLY INSTRUCTIONS

1) Unpack your board and check all parts against the parts lists enclosed in the package.

2) If gold contacts on the edge connector appear to be corroded, use pencil eraser to remove any oxidation. NOTE: Do not use Scotchbright or any abrasive material as it will remove the gold plating.

LED INSTALLATION

3) For a professional appearing finished CP-A Board two items in the assembly are important: first, the mounting of the LED indicator lamps, and second, the mounting of the paddle switches. Care is necessary in the mounting of both of these items to insure evenly spaced, straight line rows of components. If they are assembled carefully, the panel will have a professional appearance second to none. If these two items are assembled haphazardly, the panel will function; however, it will have a distinctly less than workmanlike appearance.

For maximum ease in uniform assembling, the LED indicator lamps should be installed on the board first, before any other components have been installed. They should not be pushed fully against the board, but, rather, should be set up approximately 1/8 inch to place them closer behind the acrylic panel mask; this provides for a greater viewing angle during panel operation. A small easy-to-make jig is extremely useful in accurate positioning of the LED indicators. This mounting aid consists of 1/8 inch thick material. A piece of 1/8 inch plastic, aluminum or masonite, or two pieces of 1/16 inch material such as vector board or old printed circuit board make ideal jigs.

A 3/4 square inch piece of the 1/8 inch material, or two of the 1/16 inch material should be cut and a narrow slot, such as would be produced by a hacksaw or coping saw blade, cut into one side a little bit past the center. As each light-emitting diode is installed in the board, leads can be inserted through the short slot cut into this piece and then through the board and the LED should be pushed up hard against the 1/8 inch piece so that its base sits flat and it will be held accurately 1/8 inch away from the surface of the front panel board. The lead should be soldered from the back while someone is holding the LED against the mounting aid from the front. The mounting aid can then be slipped out from under the LED.

Take care that every LED is mounted in the correct direction with the cathode down towards the 100 pin edge connector at the bottom of the board. The cathode can be recognized by its proximity to the flat side on the base of the light emitting diode.

4) Insert and solder each of the forty-four red LED's at locations:

LO0 through LO7

LA0 through LA15

LD0 through LD7

LS0 through LS7

LIE, LHD, LRN, LWT

RESISTOR INSTALLATION

5) Insert and solder each of the forty-four 220 ohm ¼ watt resistors (red/red/brown) R16 through R59. See Assembly Diagram for location.

6) Insert and solder each of the six 47K ohm ¼ watt resistors (yellow/violet/orange) R3, R4, R5, R8, R9, and R12. See Assembly Diagram for location.

7) Insert and solder one 270K ohm ¼ watt resistors (red/violet/yell R1. See Assembly Diagram for location.

8) Insert and solder each of the nine 1K ohm ¼ watt resistors (brown/black/red) R2, R6, R7, R10, R11, R13 through R15, and R60. See Assembly Diagram for location.

IC INSTALLATION

9) Insert and solder each of the two 7400 IC's at locations U14 and U25.

10) Insert and solder each of the two 7402 IC's at locations U13 and U16.

11) Insert and solder the one 7404 IC at location U10.

12) Insert and solder each of the two 74LS04 IC's at locations U8 and U15.

13) Insert and solder each of the five 7405 IC's at locations U1, U3, U4, U6, and U7.

14) Insert and solder the one 7410 IC at location U12.

15) Insert and solder the one 74LS10 IC at location U11.

16) Insert and solder the one 7427 IC at location U15.5.

17) Insert and solder the one 7430 IC at location U21.

18) Insert and solder one 74LS30 IC at location U9.

19) Insert and solder each of the three 74107 IC's at locations U18, U19, and U22.

20) Insert and solder each of the three 74123 IC's at locations U17, U20, and U23.

21) Insert and solder the one 8T97 IC at location U24.

22) Insert and solder the one 8212 IC at location U5.

DISCRETE COMPONENT INSTALLATION

NOTE: Lead allowance on all capacitors must be long enough to allow them to be flattened (or laid down) flush on the board or chip to facilitate front panel mounting.

23) Insert and solder each of the seventeen .1uf capacitors at locations C1, C2, C5, C6, C7, and C11 through C22.

    C2, C3, C5, C6, and C7 should be laid down.

24) Insert and solder each of the two .001 uf capacitors at locations C3 and C8.

25) Insert and solder the one .01 uf capacitor at location C4.

26) Insert and solder the two 33 mf 25 volt tantalum capacitors at locations C9 and C10. NOTE: Observe polarity as marked on board.

27) Insert and solder the 1N914 diode at position CR1.

REGULATOR AND HEAT SINK

28) Before installing heat sink bend all the heat sink fins horizontally (outward) to facilitate front panel mounting. The middle fin located on the right hand side of the board (when mounted) should be broken off or bent inward in order to allow space for the INTERRUPT/ENABLE LED (LIE) to be seen through the front panel.

29) To install the regulator and heat sink first bend the 7805 regulator leads at 90 degree angles to a length which allows their insertion into the hole pattern of the CP-A board. Then place heat sink as shown in Assembly Diagram and insert regulator as described above. Use a #6 screw on the component side of the board and lockwasher and nut on the solder side of the board. Tighten the screw carefully to insure proper alignment of the heat sink to prevent shorting to adjacent traces.

CP-A TO MPU-A INSTALLATION

30) Using the 16 conductor ribbon cable with 16 pin 3M dual inline connector, insert one end into the hole pattern U2 from the back side of the CP-A board so that it can be soldered from the front (component side) of the CP-A board. The cable should be mounted so that it extends upward from the top of the chassis when the board is mounted.

SWITCH INSTALLATION

NOTE: There are three types of switches included for installation on the front panel (disregarding color). They include:

A. One 2 position red rocker switch. This is the AC power switch.

B. 5 momentary 3 position with spring return to center paddle switches - identified by the lack of a Nipple (raised portion) on the front of the switch mounting tab.

C. 16 2 position no spring return paddle switches.

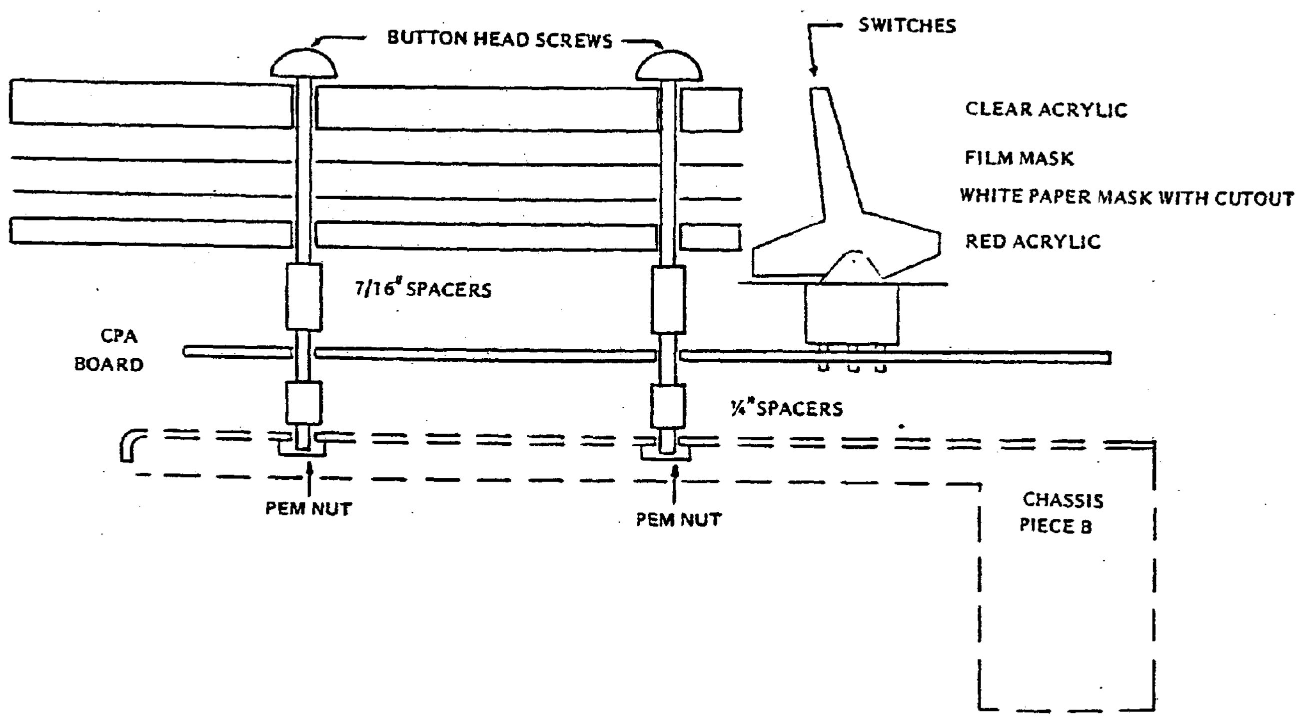
NOTE: Temporarily plug a 100 pin edge connector on the CP-A Board while switches are being soldered to help insure proper spacing between the PC Board and switch bracket.

31) The last step is the assembly of the switches and the switch mounting bracket. Note that the front panel includes switches whether mounted in the front of the cabinet or not. The POWER/ON/OFF Rocker Switch mounts at the extreme right switch position. The Paddle Switches are provided in both two-position and center-off spring return types. The two-position switches are used for the ADDRESS-DATA and ADDRESS PROGRAMMED INPUT location while the center off-spring return are used for the Control Functions.

32) When the entire row has been spaced accurately, the board should be turned over and a center switch should be soldered in place taking care that the board is not bowed towards or away from the switches. When the board is positioned correctly, there will be a small space approximately 3/64 inch or slightly under 1/16 inch between the bottom of the switch and the front of the front panel board. The two end switches should be similarly checked to make sure that the spacing to the board is correct and soldered in place, and then one switch each at the ¼ positions checked as to spacing from the board and soldered into place. Then the remainder of the switches can be soldered. Examine visually for solder splash or bent/unsoldered pins.

PANEL ASSEMBLY

Refer to the diagram to see how the clear front acrylic piece, the photograph mask, the die cut paper backup and the red acrylic panel are assembled in sequence with the 6/32 x 1½ inch button head screws.



Both the Photographic mask and the paper backup sheet should be trimmed to size after assembly. Marks are provided on both, and they should be cut out carefully using a straight edge and a very sharp knife against a wooden cutting board. Scissors may be used if a guide line is first drawn on the sheets. The 8 holes for the assembly screws should be cut out on the mask and the paper sheet as indicated in the diagram. Then the protective paper may be removed from the two acrylic sheets and the sandwich assembled carefully. Avoid getting dush caught in between any of the pieces. A soft lint-free rag very slightly moistened can be an aid in cleaning any dust off plastic or file surfaces.

When the acrylic pieces, film and paper have been assembled, eight 7/16 inch spacers may be slipped over the screws and then the whole assembly inserted through the mounting holes on the CP-A board. Take care that there is no interference from any component standing too high and that the acrylic panels sit down completely on the 7/16 inch spacers against the board.

Eight ¼ inch spacers can then be slipped over the screws behind the CP-A board and eight #6 nuts and lockwashers can be put on to hold the sandwich together.

The panel board should now be ready to plug in and use. If the board is going to be assembled in the front location of a cabinet, to serve as a permanent front panel, the eight nuts should be removed at this time. Install the cap screws in the PEM nuts in the front panel sheet metal.

USER GUIDE

The CP-A board contains no user option jumpers or any other special connections that must be made to use the board. If the panel is mounted in the IMSAI 8080 cabinet then the power on/off switch should be connected using a separate wire to the power supply section as described in the Power Supply documentation. If the panel board is not going to be mounted in the cabinet, then the power switch should not be connected. In this case, the power switch on the inside or on the back of the cabinet would be used for controlling power to the IMSAI 8080.

Panel installation requires a backplane slot. The panel may be plugged directly into the front slot of the mother board (with the mounting screws from the acrylic face plate assembly extending through the metal panel immediately behind and secured with lock washers and nuts) or plugged on an extender card into any location in the back plane.* The 16 pin DIP plug on the end of the flat cable must also be inserted into the corresponding socket in the upper right hand corner of the MPU-A board.

Only one front panel should be plugged into the bus at any time to avoid conflicts between multiple driving sources on the same signal lines for some of the control lines between the front panel and the system bus. The front panel is now ready to operate.

The 16 ADDRESS-DATA and ADDRESS-PROGRAMMER INPUT switches are 2 position paddle switches and represent a 0 in the down position and a 1 in the up position. The switches are provided in two colors, and can be arranged either in color groups of four to assist programming in hexadecimal or color groups of 3, 3 and 2, to aid in octal programming.

The low order byte of address switches, serve to enter into memory either data or program instructions. These switches are labeled ADDRESS-DATA 0 through 7. Each byte of data or program that is to be entered from the front panel is set into these switches after the appropriate address has been selected and entered. The switch

----

*Switches should be included whether the front panel is mounted in the front of the cabinet or not.

positions are not indicated on the indicator lights until the information is deposited in memory. At that time the information from these switches appears on the data bus. The high order byte of address switches is labeled ADDRESS-PROGRAMMED INPUT and these switches can be read by the program as input port position hex FF or octal 377. The additional labels 0 through 7 are provided above these switches to assist in interpreting the switch positions when being used as an input port. The position of these switches does not appear in the indicators until the input instruction from position FF is executed, during which execution time the switch positions appear on the data bus as it is being read into the 8080 processor.

The six control switches are grouped at the right end of the panel. They are center-off two-position spring-return switches with the exception of the POWER ON/OFF switch, which is a rocker type to eliminate accidental powerdowns. The function switches are provided in alternating colors for easy identification and to reduce operator error.

The RUN/STOP switch controls program execution. When the switch is pushed to the RUN position, a control signal is sent to the processor board and enables it to start or continue executing program instruction beginning in the location indicated at that time in the address bus lights. When the address switch is depressed to the STOP position, this enable signal is removed from the processor board at the beginning of the next instruction cycle so that the processor will stop executing during the fetch cycle for that following instruction.

When the processor is enabled to RUN, the RUN light above the RUN/STOP switch will be lit. When the processor has been stopped, the WAIT light to the right of the RUN light will be lit. During normal operation, the RUN light will be on full and the WAIT light will be on partially, the exact amount depending on how many wait cycles are required by the memory and peripheral devices being run by the processor at the moment.

The front panel must be holding the processor in the stopped condition for the SINGLE STEP switch, the DEPOSIT/DEPOSIT NEXT switch, or the EXAMINE/EXAMINE NEXT switch to operate.

The EXAMINE/EXAMINE NEXT switch provides the facility for observing what is stored in memory in any location or for setting the program counter to any desired location to initialize program execution there.

When examining the contents of a location in memory, the 16 address switches are used to enter the 16 bit address. This 16 bit address is normally said to be divided into two 8-bit sections labeled high order and low order. The high order address is on the left hand side of the panel, and the low order address is in the center. The low order byte contains bits 0 through 7 and the high order byte contains bits 8 through 15. When only a small amount of memory is being used the high order bits are normally 0 and the switches must be in down position, unless the address jumper selection on the memory board is wired otherwise.

When the EXAMINE switch is actuated, the processor jumps to the address location set in the 16 address switches and is stopped during the fetch cycle out of that memory location. At that time, the address bus indicators will show the address set in the 16 address switches and the data bus indicators will show the contents of that memory location. Any additional locations in memory may be observed by setting the 16 address switches to that desired address and actuating the EXAMINE switch again. When the EXAMINE NEXT switch is actuated, the address shown in the address bus indicators is incremented by 1 and the contents of that following memory location are displayed on the data bus lights. Thus, a program or data would normally be checked by setting the first address in the address switches and actuating the EXAMINE switch to see the first byte, and thereafter actuating the EXAMINE NEXT position to observe each succeeding byte of data or program.

The DEPOSIT/DEPOSIT NEXT switch is similar in its operation but provides for changing the data or program stored in the memory. When the switch is actuated to the DEPOSIT position, the values of the lower address byte switches, that is, bits 0 through 7 labeled Address-Data, are deposited into the address currently being indicated on the 16 address bus indicators. After the DEPOSIT switch is actuated, the data will appear on the data bus indicators. If the data was incorrect because the switches were set wrong, the switches can be changed, the DEPOSIT switch actuated again, and the new values will be deposited to memory in that same location.

When the DEPOSIT NEXT position is actuated, the address currently appearing in the 16 address bus indicators is first incremented by 1 and the data entered into the ADDRESS/DATA switches is deposited in that following location and will appear in the data bus. The DEPOSIT NEXT position functions exactly the same as depressing EXAMINE NEXT to increment the address bus by 1 and then actuating DEPOSIT to deposit the ADDRESS-DATA switch positions into that location.

When the processor is stopped, instructions may be executed one at a time through the use of the SINGLE STEP switch to the right of the RUN/STOP switch. If this switch is depressed or raised, the processor board is permitted to run one instruction, and it will stop when it is in the fetch cycle in the following instruction. Thus repeated operations of this switch permit the programmer to step through his program one instruction at a time and follow what the machine is doing, noticing on the data bus what the fetched instruction is, and on the address bus the location from which that instruction is being fetched. For instructions requiring multiple memory accesses, for instance those with an address following in the second or third byte, each operation of the SINGLE STEP switch advances through only one part of the instruction. Thus, each byte of the instruction being read in and each byte of data being read in or out may be observed on the panel.

The RESET/EXTERNAL CLEAR switch provides the system reset functions. When depressed to the EXTERNAL CLEAR position the CLEAR signal is given to all external input/output interface cards which are wired to be reset by this signal. When raised to the RESET position, the 8080 processor is reset. This sets the program counter to location 0 and then returns control to the processor. If the front panel is permitting the processor to run when the RESET switch is actuated, upon release of the RESET switch the processor continues execution starting at position 0. If the front panel was holding the processor in a stopped state, during the time the RESET switch was actuated, then the program counter will be set to 0. When the RESET switch is released, the processor will remain stopped and will be positioned at memory location 0.

The 8-BIT PROGRAMMED OUTPUT INDICATOR lights can be controlled by the program through the use of the output instruction to port location hex FF or octal 377. When 0 bits are output into this port, the indicator lights will be turned on and when 1 bits are output into this port, indicator lights will be turned off.

The STATUS BYTE INDICATOR LIGHTS display the condition of the status byte during the execution of that instruction. The 8 status bits included in the status byte are the Memory Read Bit, the Input Instruction Bit, the Instruction Fetch Bit M1, the Output Instruction Bit, the Halt/Acknowledge Bit, the Stack Operation Bit, the Write Output Complement Bit, and the Interrupt Acknowledge Bit. In normal front panel operation, whenever the machine is stopped and the EXAMINE, DEPOSIT, EXAMINE NEXT or DEPOSIT NEXT switches are being used, the MEMORY READ, the M1 INSTRUCTION FETCH, and the WRITE OUT COMPLEMENT STATUS lights should be on.

While single-stepping through a program, either these or other status lights will be on as appropriate to the instruction function being executed at that moment.

For a more complete description of the functions of the status bits, reference should be made to the Intel 8080 Micro Computer Systems User's Manual. The INTERRUPT ENABLED indicator is turned on whenever the interrupts are enabled into the 8080 processor by the INTERRUPT ENABLE INSTRUCTION. This light is turned off either by an interrupt occurring and the processor acknowledging it, or by the instruction to disable interrupts. The HOLD indicator light is lit whenever a special-purpose input/output card is holding the processor so as to gain direct access to the memory on the system bus.

March 5,1977

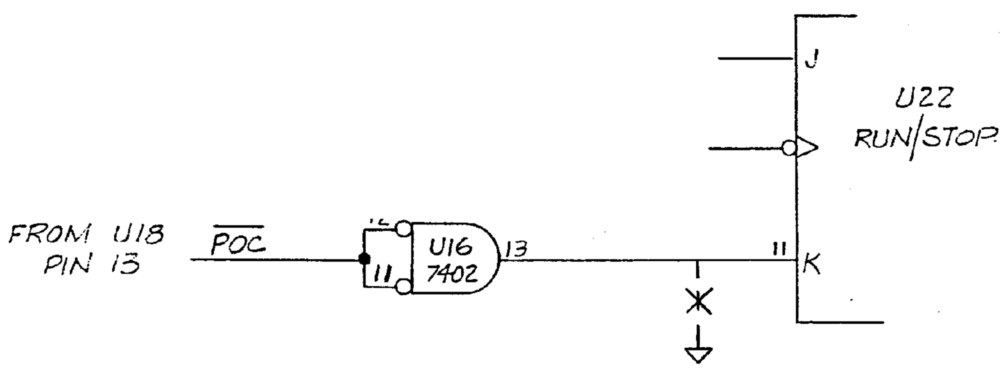
CPA REV 4 MODIFICATION

Modification to cause front panel to always come up in "stop" mode at power-up time.

1) Cut (comp. side) U-22 pin 11 free. (U-22 pin 11 was connected to U-22 pin 4 (ground) by a heavy trace under the chip.)

2) Connect (solder side) U-18 pin 13 to U-16 pins 11 and 12. Connect wire at the pads.

3) Connect (solder side) U-16 pin 13 to U-22 pin 11. Connect wire at the pads.



ERRATA FOR CPA REV-4 AND EARLIER

1. The following modification must be made to the CPA REV-4 or earlier REV's if it is to be used with the RAM-16, RAM-32 or RAM-65 memory boards. This change makes the signal on backplane line 71 (RUN) agree with the bus definition. The change does not affect the CPA's compatability with other IMSAI products.

2. Refer to Fig 1 and make the following cut on the component side of the board:

   (a) Cut the trace extending down from U24 pin 9.

3. Refer to Fig 2 and make the following cuts on the solder side of the board.

   (a) Cut the trace from U24 pin 10 between this pin and feed through A.

   (b) Cut the trace from feed through B near the spare IC location.

   (c) Remove the entire pad of the feed through connected to edge connector pin 71.

4. Refer to Fig 2 and install the following jumpers on the solder side of the board.

   (a) From U22 pin 5 to U24 pin 10.

   (b) From edge connector pin 71 to U24 pin 9

   (c) From feed through A to feed through B.

5. Correct the schematic as shown in Fig 3.

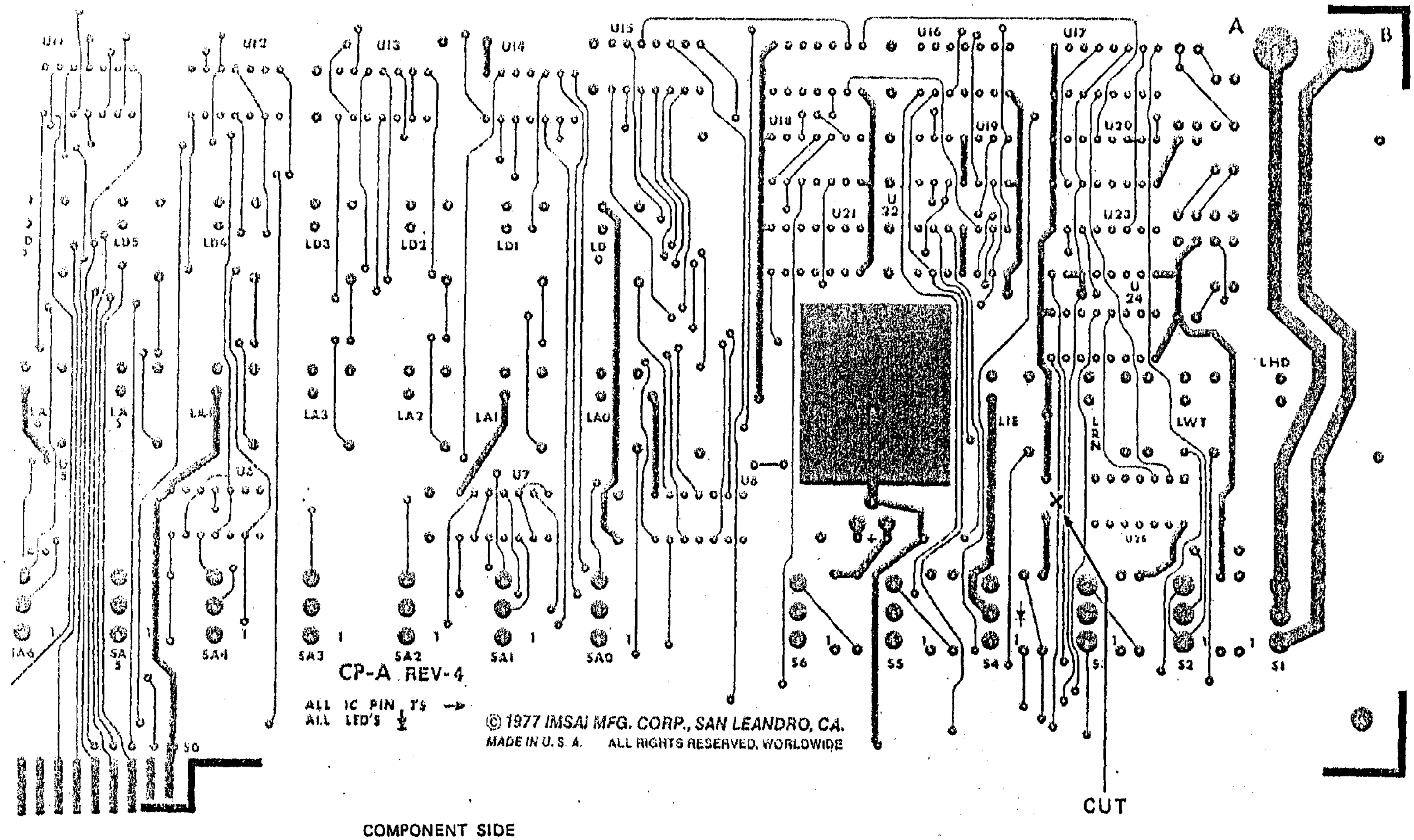


FIGURE 1
MODIFICATION TO CPA REV. 4 & EARLIER
FOR DYNAMIC RAM OPERATION
ECN - 77 - 0039

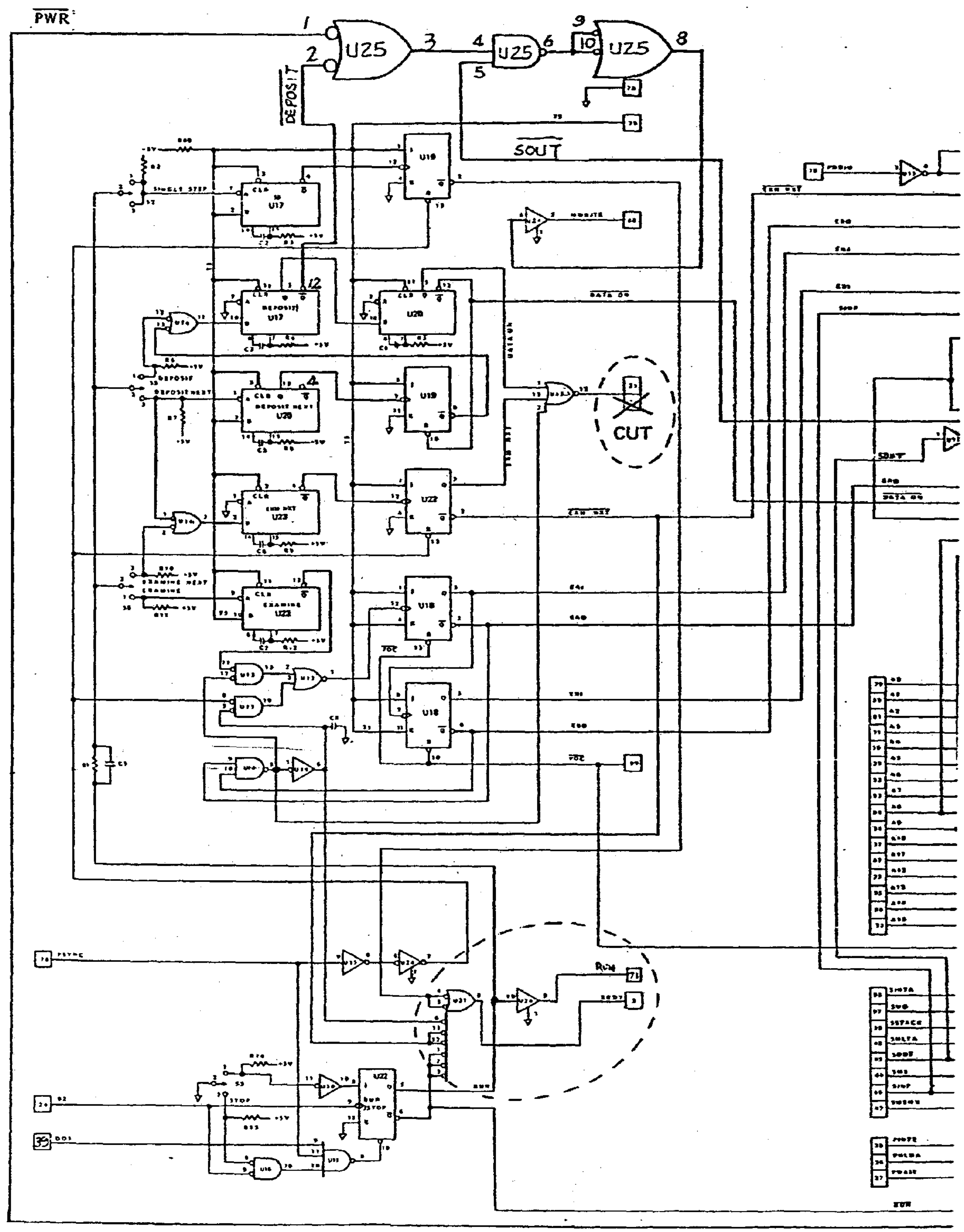


FIGURE 3
MODIFICATION TO CPA REV. 4 & EARLIER
ECN 77–0039

OPTIONAL MODIFICATION OF CP-A REVISION 4 OR EARLIER CP-A BOARDS TO CHANGE POWER SWITCH TO WRITE PROTECT/UNPROTECT SWITCH FOR USE WITH RAM 4A BOARDS.

REMOVE CP-A FROM CHASSIS

A. Remove AC leads from pads A & B on CP-A, route to miniature toggle switch (e.g., C&K type 7101) mounted in ¼" hole (provided) in rear of chassis. Connect to center and bottom terminals of switch.

B. Carefully suck solder away from terminals of Power switch on CP-A using a solder sucker or pieces of copper braid. Use enough heat to melt solder, but do not overheat board. Unscrew the Power switch from the bracket and heat all 3 terminals simultaneously. (Use 3 irons, or "timeshare" one--moving between the terminals.) When all 3 terminals are hot enough, the switch will easily lift out. DO NOT PULL. Pulling will damage the pads.

C. Cut the following traces (refer to diagram):

   Between center and upper AC Power switch terminal (front side),
   ground lead going to HOLD light (back side),
   trace to resistor on HOLD light (after feed through) (backside),
   trace to Mother board connector pin 20 (after feed through) (backside),
   trace to Mother board connector pin 70 (after feed through) (front side).

D. Re-install a 3-position momentary switch in old AC Power switch position.

E. Install two 470 Ohm, ¼ watt resistors between ground and the heavy traces from the switch (or to U24 pins 12 & 14).

F. Install the following jumpers:

   From +5 to resistor from old HOLD light (other end than LED)

   From U22 pin 6 to switch center terminal

   From pin 15 to pin 1 on U24

   From the heavy trace (to the bottom AC Power switch terminal) to U24 pin 14

   From the heavy trace (to the top AC power switch terminal) to U24 pin 12

   From U24 pin 13 to the pad connected to pin 70

OPTIONAL MODIFICATION OF CP-A (Continued)

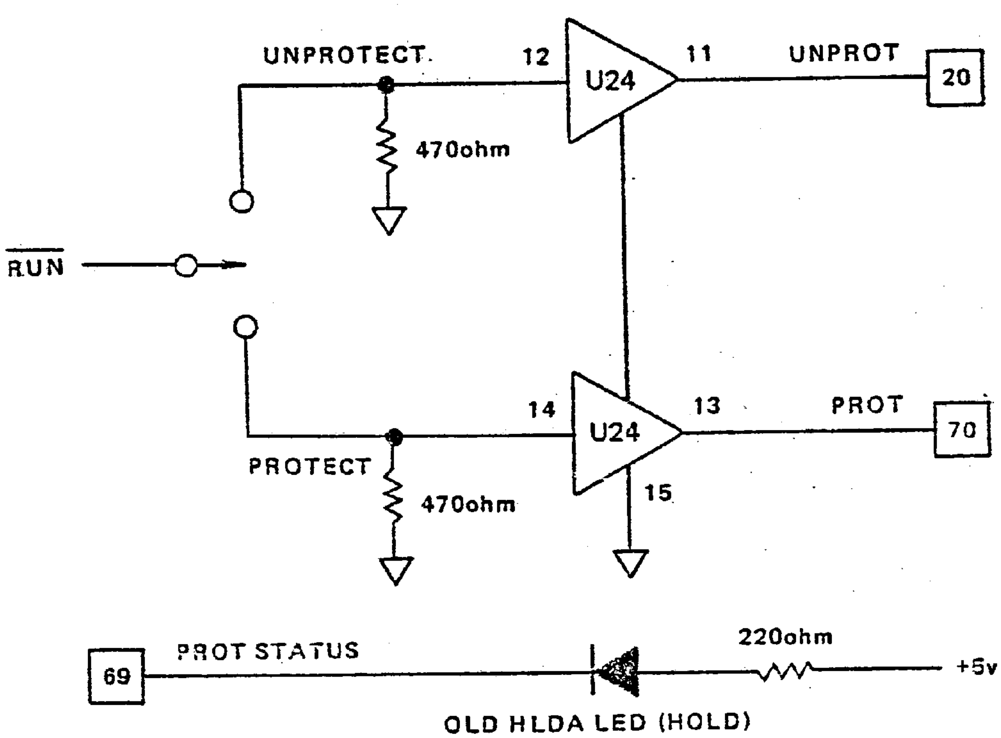
From U24 pin 11 to the pad connected to pin 20

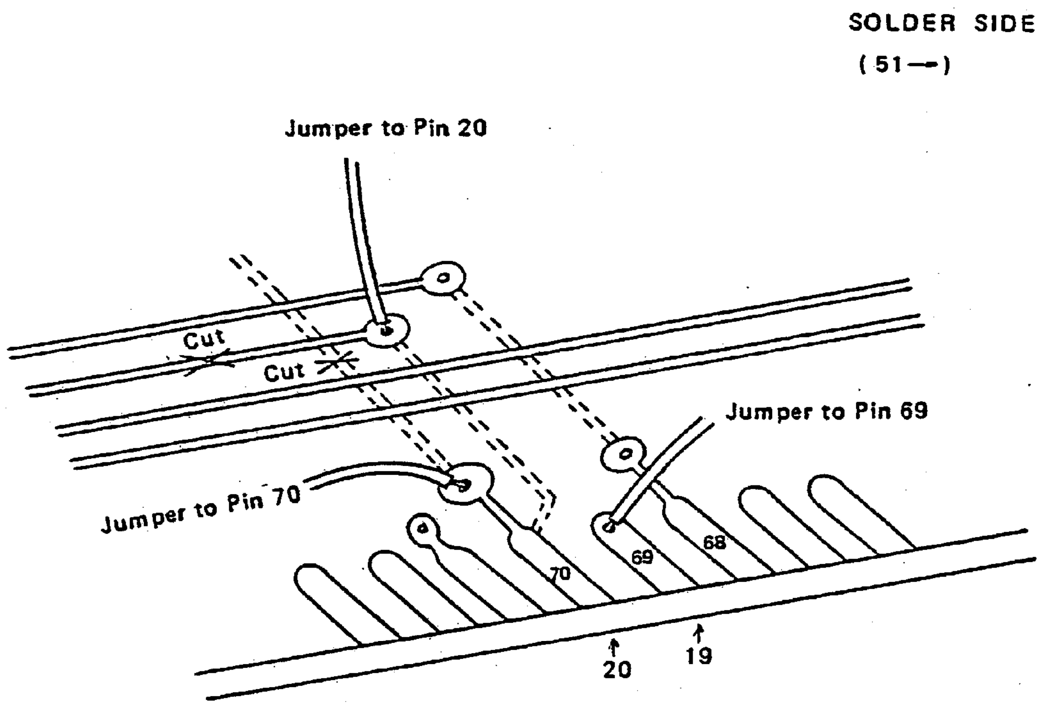
From the bottom terminal of the HOLD light (cut from ground) to pin 69 (solder to the top 1/16" of finger only)

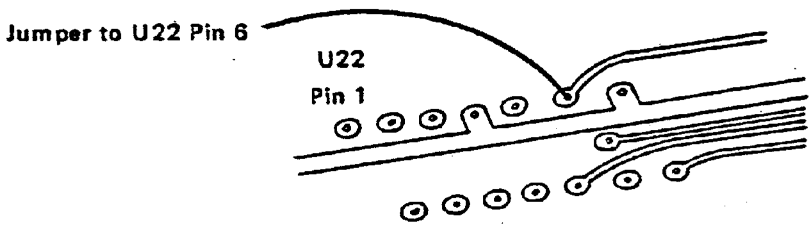
nsert labels to change HOLD (for light) to MEM. PROTECT and to hange POWER ON/POWER OFF to PROTECT/UNPROTECT.

e-assemble CP-A to chassis

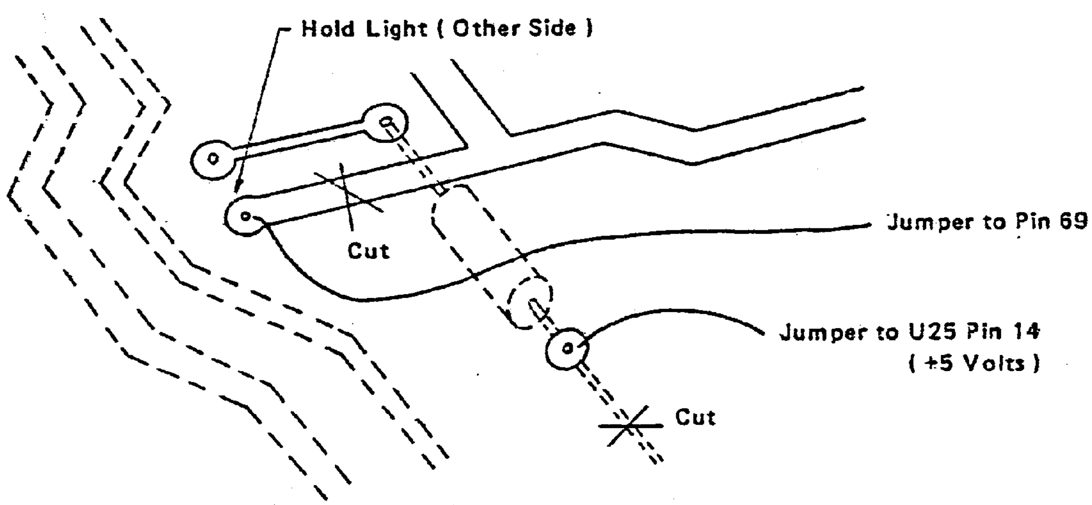
he right-hand switch now serves to change the protect status of he currently addressed block of memory when the machine is not n run mode. The LED which previously showed HOLD status now is it when the currently addressed block of memory is protected.


UNPROTECT
12
U24
11
UNPROT
20
470ohm
$\overline{RUN}$
14
U24
13
PROT
70
PROTECT
470ohm
15
PROT STATUS
69
220ohm
+5v
OLD HLDA LED (HOLD)

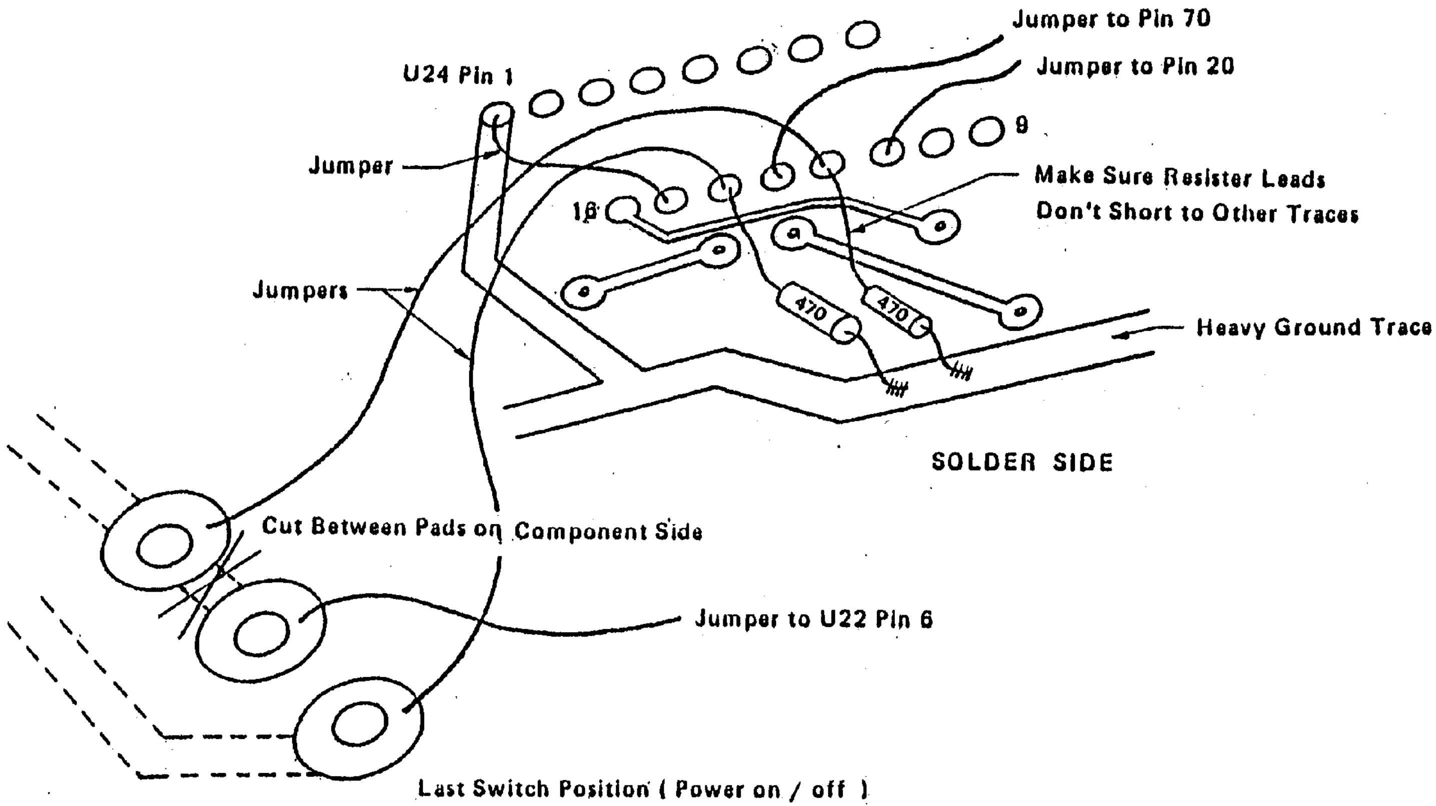

SOLDER SIDE
(51→)
Jumper to Pin 20
Cut
Cut
Jumper to Pin 69
Jumper to Pin 70
70
69
68
20
19

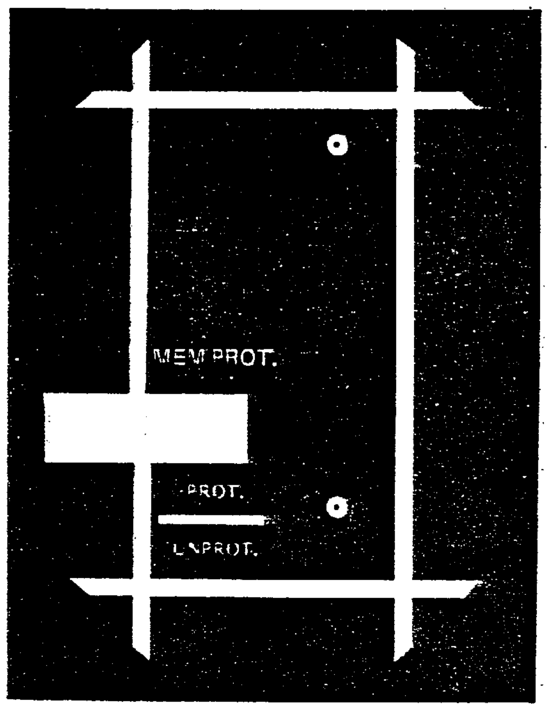


SOLDER SIDE

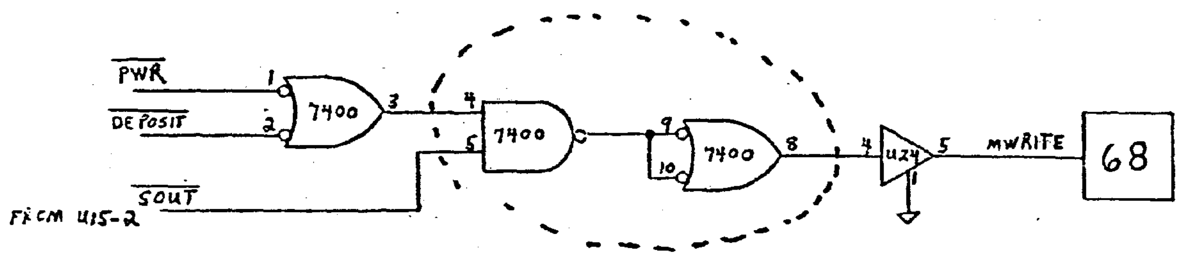


SOLDER SIDE


Jumper to Pin 70
Jumper to Pin 20
U24 Pin 1
9
Jumper
16
Make Sure Resister Leads
Don't Short to Other Traces
Jumpers
470
470
Heavy Ground Trace
SOLDER SIDE
Cut Between Pads on Component Side
Jumper to U22 Pin 6
Last Switch Position (Power on / off)


MEM PROT.
PROT.
UNPROT.

CP-A FRONT PANEL FOR REVISIONS 3 & 4
FOR USE WITH NON-IMSAI MEMORY BOARDS



This modification should be made to your front panel board using two additional sections of the 7400 that is located directly above S-2 and S-3. This will keep your unit from writing into memory during an output.

February 3, 1976

IMSAI

MPU-A

MPU-A

FUNCTIONAL DESCRIPTION

The MPU-A board is the processor board for the IMSAI 8080 Microcomputer System. It is designed using the 8080 microprocessor chip. The bus arrangement and board connector has been chosen to be 100% compatible with the MITS Altair M8800 Microcomputer system so that all boards are 100% interchangeable between the Altair system and the IMSAI 8080 system.

Every effort has been made to keep the design simple and straight-forward to maximize reliability and ease of maintenance. MSI and LSI are used where appropriate, and discrete components are held to a minimum for greater circuit reliability and ease of assembly.

The 8224 clock driver chip and an 18 Megahertz crystal are used to generate the 2-phase, 2 Mehagertz non-overlapping clock for the 8080A. An 8212 is used as a latch for the status signals and two 8216 tri-state bi-directional bus drivers are used to interface the 8080A with the IMSAI 8080 input and output data buses. All other address, status, and control lines are driven by tri-state bus drivers.

Unregulated +16, -16, +8 volts, and ground must be supplied to the bus. On-board regulation is used to arrive at the power supply levels needed to run the chips. Integrated circuit power regulators with overload protection are used. The board is supplied with ample bypass filtering using both disc ceramic and tantalum capacitors.

The board connector is a 100 pin edge connector on .125 inch centers 50 pins on each side. Dimensions are 5 inches by 10 inches, using 2 sided glass reinforced epoxy laminate, with plated feed-through holes to eliminate the need for any circuit jumpers. The contact fingers are gold-plated over nickel for reliable contact and long life. All other circuitry is tin-lead plated for better appearance and more reliable solder connections.

Power-on reset is included on this board along with pull up resistors for all inputs required so that with the front panel removed from the INSAI 8080 machine, the power-on reset will start the program at position 0 out of a ROM. All other necessary conditions are met so that the system will run without the front panel attached, for use in dedicated controller applications where no operator-processor interaction is desires.

THEORY OF OPERATION

The IMSAI MPU-A board is structured around the 8080A microprocessor chip, and much of the MPU-A board is wired to support the 8080A device. The MPU-A board provides interfacing between the 8080A chip and the data and address busses, clock and synchronization signals, and the voltage regulation necessary for the 8080A and other chips. The internal functioning of the 8080A is thoroughly described in the Intel 8080 Microcomputer System User's Manual. Reference should be made to this manual for information concerning the operation and use of the 8080A.

The address lines from the 8080A drive the address bus on the back plane through 8T97 tri-state buffer drivers. These drivers may be disabled through the ADDRESS DISABLE line on pin 22 of the back plane. The 8216 bi-directional bus drivers connect the 8080's bi-directional DATA IN and DATA OUT busses. The direction of data transmission is determined by the DIRECTION ENABLE line. The DIRECTION ENABLE line is in turn controlled by the front panel and the processor status signals DATA BUS IN and HALT ACKNOWLEDGE. The 8216 can be disabled by the DATA OUT DISABLE line on pin 23 of the back plane.

The 8080A's bi-directional data bus is also connected to the data bus socket and the 8212 status byte latch. The data bus socket is used to connect the front panel to the bi-directional bus, while the 8212 latch transfers the status byte to the back plane via 8T97 drivers. These drivers are disabled by the STATUS DISABLE line on pin 18 of the back plane. The 8212 is latched up by the STATUS STROBE signal of the 8224 clock chip to store the status information for each instruction cycle.

One K pullup resistors to +5 volts are connected to all the bi-directional bus lines to ensure that during the time the bus is not driven, the 8080A reads all 1's.

The 8224 clock chip and crystal oscillator provide the two-phase non-overlapping 2 megacycle system clock for the 8080A. These clocks are also driven onto the back plane through 8T97 tri-state buffered drivers.

The CLOCK line on the back plane is driven from the TTL Phase II clock line through a delay so that the phase relation of the clock signal to the Phase II and Phase I back plane signals, is nearly identical to that produced by the MITS Altair 8800 system. Six sections of a 7404 are used for this delay to provide greater simplicity and higher reliability than a one-shot. The 8224 chip also provides the power-on reset function through use of a 4.7K resistor and 33 uf capacitor connected to the reset input of the 8224. The power-on reset is applied to the 8080A and is applied to the POWER ON CLEAR line, pin 99 on the back plane.

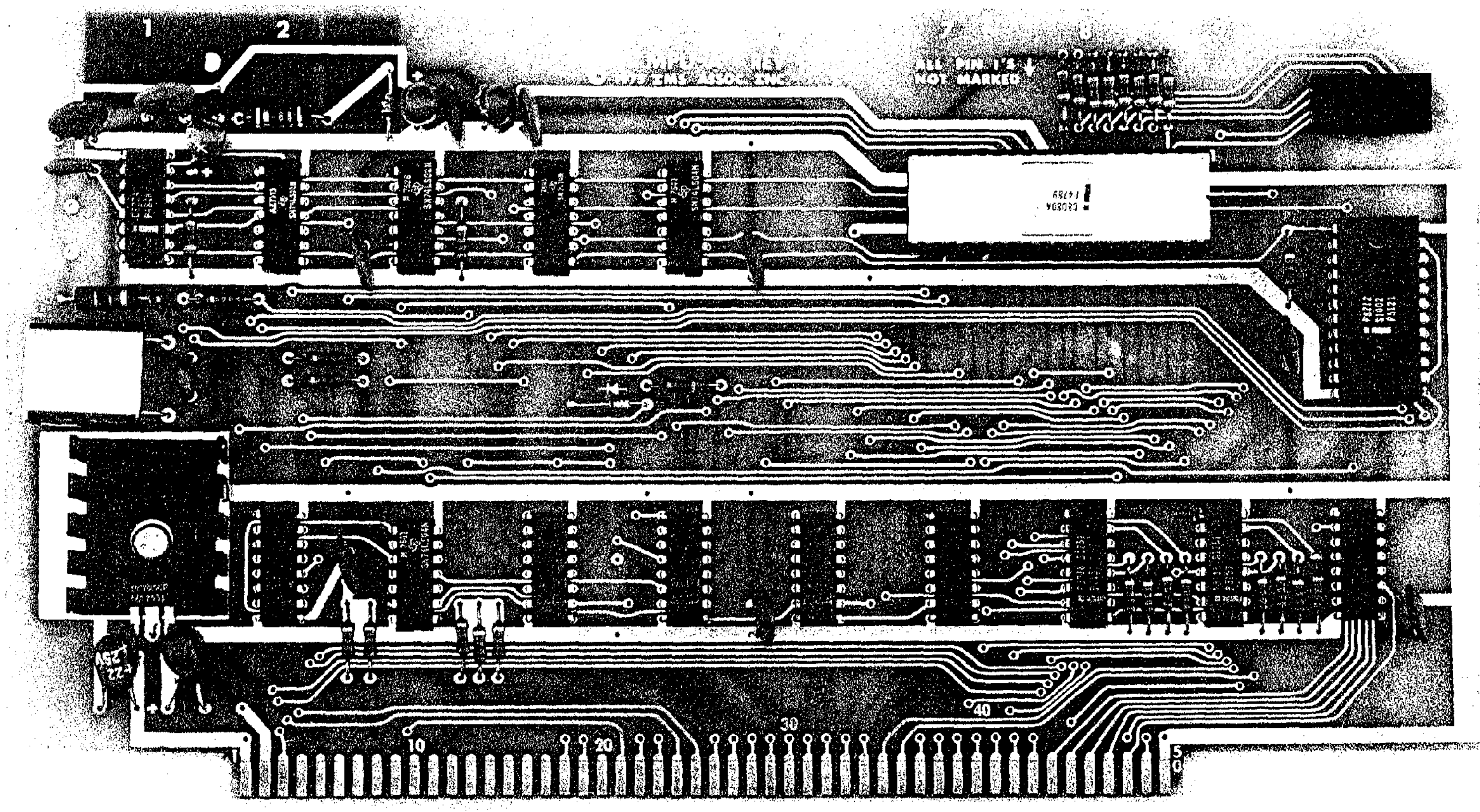
The two BACK PLANE READY signals are ANDed and connected to the 8224 for synchronization with the Phase II clock before being connected to the 8080A chip. The INTERRUPT line is connected directly to the 8080A, while the HOLD REQUEST line is synchronized with the Phase II clock and then connected to the 8080A.

The six processor status signals (SYNC WRITE, STROBE DATA BIT IN, INTERRUPT ENABLED, HOLD ACKNOWLEDGED, and WAIT ACKNOWLEDGE) are all driven onto the back plane through 8T97 tri-state buffered drivers. These drivers may be disabled by the CONTROL DISABLE line, pin 19 on the back plane.

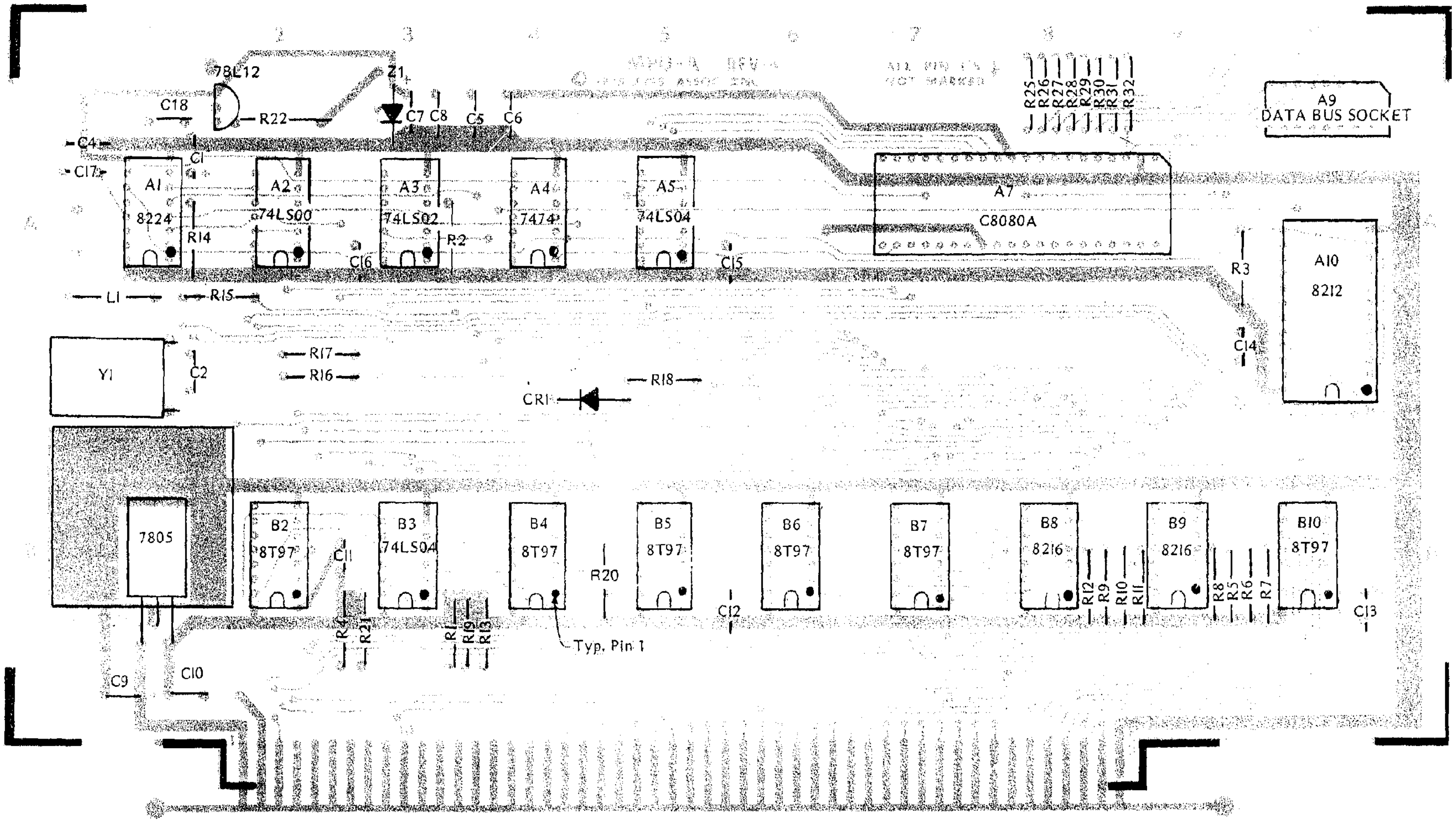
The +5 volts is regulated from the +8 volts by a 7805 integrated circuit regulator, while the -5 volts is regulated by a 5 volt zener and a 470 ohm resistor from the 16 volt bus. The +12 volts is regulated by a 12 volt Zener and connected to the +16 volt line by two 82 ohm ½ watt resistors in parallel. All voltages are filtered with .33 microfarad tantalum and disc ceramic capacitors.

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
74LS00	36-0740002	1	Quad 2 Input NAND (Low Power Schottky)/ SN74LS00N
74LS02	36-0740202	1	Quad 2 Input NOR (LPS)/SN74LS02N
74LS04	36-0740402	2	Hex Inverter/SN74LS04N
7474	36-0747401	1	Dual D Flip Flop/SN7474N
7805	36-0780501	1	5V Positive Voltage Regulator/ MC7805CP
8080A	36-0808001	1	Microprocessor/C8080A
8212	36-0821201	1	Input/Output Port/P8212/S1002
8216	36-0821601	2	Bi-Directional Bus Driver/D8216
8224	36-0822401	1	Clock Generater and Driver/D8224
Diode	35-1000006	1	Silicon Diode 1N914/blue, black
8T97	36-0089701	6	Hex Tri-State Driver/N8T97B
Capacitor	32-0239010	1	39 pF Disk Ceramic
Capacitor	32-0256010	1	56 pF Disk Ceramic
Capacitor	32-2010010	10	.1uF Disk Ceramic
Heat Sink	16-0100002	1	Thermalloy Heat Sink/6106B-14
78L12	36-0781202	1	12V Regulator/MC78L12CP
Inductor	29-400001	1	1 uH Inductor/WEE 1.0
Resistor	30-3470462	1	470 Ohm, ½ Watt/yellow, violet, brown
Resistor	30-4100362	19	1K Ohm, ¼ Watt/brown, black, red
Resistor	30-4470362	10	4.7K Ohm, ¼ Watt/yellow, violet, red
Socket	23-0800001	1	16 Pin Solder Tail Socket

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Socket	23-0800004	1	40 Pin Solder Tail Socket
Capacitor	32-2233070	5	33-25 Tantalum Capacitor
Crystal	35-5000001	1	18.00 MHz Crystal
Diode	35-1000005	1	1N751A Zener Diode
Screw	20-3402001	1	6-32x3/8" Phillips Pan Head Machine
Nut	21-3120001	1	6-32 Hex Nut
Lockwasher	21-3350001	1	#6 Internal Star Lockwasher
Solder	15-0000001	5'	



MPU-A REV 4



IMS Associates Inc.

ASSEMBLY DIAGRAM

MPU-A REV 4 2/76

MPU-A ASSEMBLY INSTRUCTIONS

1) Unpack your board and check all parts against the parts lists enclosed in the package.

2) If gold contacts on the edge connector appear to be corroded, use pencil eraser to remove any oxidation. NOTE: Do not use Scotchbright or any abrasive material as it will remove the gold plating.

RESISTOR INSTALLATION

3) Insert and solder nineteen 1K ohm 1/4 watt resistors (brown/black/red) R1 through R13, R15 through R17, R19, R20 and R21. See Assembly Diagram for location.

4) Insert and solder ten 4.7K ohm 1/4 watt resistors (yellow/violet/red) R14, R18, and R25 through R32. See Assembly Diagram for location.

5) Insert and solder one 470 ohm 1/2 watt resistor (yellow/violet/brown) R22. See Assembly Diagram for location.

IC INSTALLATION

NOTE: All IC pin 1's point in the direction of the edge connector unless otherwise indicated on the board.

6) Insert and solder the one 74LS00 in location A2.

7) Insert and solder the one 74LS02 at location A3.

8) Insert and solder each of the two 74LS04's at location B3 and A5.

9) Insert and solder the one 7474 at location A4.

10) Insert and solder the one 8224 at location A1.

11) Insert and solder each of the six 8T97's at locations B2, B4, B5, B6, B7, and B10.

12) Insert and solder each of the two 8216's at locations B8 and B9.

13) Insert and solder the one 8212 at location A10.

DISCRETE COMPONENT INSTALLATION

14) Insert and solder the 16 pin IC socket at location A9.

15) Insert and solder the 40 pin IC socket at location A7. (Do not install 8080 at this time.)

16) Insert and solder the one 1N914 diode (CR1) as shown in the Assembly Diagram. NOTE: Observe polarity as indicated on the board.

17) Insert and solder the one 5.1 volt zener diode 1N751 (Z1) as shown on the Assembly Diagram. NOTE: Observe polarity as indicated on the board.

18) Insert and solder the one 18 MHz crystal (Y1) as shown on the Assembly Diagram.

19) Insert and solder the one 1uH inductor (L1) as shown on the Assembly Diagram.

20) Insert and solder the one 39 pf disk capacitor (C2) as shown on the Assembly Diagram.

21) Insert and solder the one 56 pf disk capacitor (C17) as shown on the Assembly Diagram.

22) Insert and solder each of the ten .1 uf disk capacitors at locations C4, C6, C8 and C11 through C16, and C18.

23) Insert and solder each of the five 33 uf tantalum capacitors C1, C5, C7, C9 and C10 as shown on the Assembly Diagram. NOTE: Observe polarity as marked on the board.

REGULATOR AND HEAT SINK INSTALLATION

24) Insert and solder the one 78L12 -12 volt regulator observing orientation as shown on the Assembly Diagram and on the board.

25) Bend the leads of the 7805 regulator at 90 degree angles approximately ¼" from the bottom edge of the regulator to facilitate insertion on top of the heat sink.

26) Insert the #6 screw through the regulator and heat sink and attach washer and nut from the back side of the board. NOTE: Be sure to hold the heat sink in proper vertical position while tightening the screw in order to prevent shorting to adjacent traces. Solder in the 7805 leads.

Before Installing the 8080 Chip

If possible, before plugging in the 8080A chip, the board should be inserted in a chassis, the power turned on, and the the voltage levels checked on the 40 pin socket. Pin 2 should be ground and pin 11 should be -5 volts. Pin 20 should be +5 volts and pin 28 should be +12 volts. If one of these three voltages is not correct, ascertain the cause and correct it before plugging in the 8080A chip. When these voltages measure correctly, the 8080A chip should be inserted carefully into the 40 pin socket (with the board removed and the power off!)

27) Finally insert the 8080A Microprocessor chip in the 40 pin IC socket located at A7. Orient pin 1 as indicated on the board.

NOTE: The 16 pin IC socket located at A9 is where the front panel data bus cable plugs into the MPU-A board.

USER GUIDE

The IMSAI MPU-A board requires no jumpers or user options for its use. The board is ready to function after connection to the back plane and the bi-directional bus. The bi-directional bus lines are provided by a 16-conductor cable from the CPA board, connected via a 16-pin DIP plug in location A-10. Verify proper insertion of this plug (i.e., pin 1 to pin 1) before use of the board.

The clock crystal frequency is 18 megahertz, and the 8224 device derives from this 18 MHz signal the necessary 2 MHz two-phase non-overlapping system clock. These 2 MHz clocks are brought out onto the back plane for use by other system boards. The board must be used with an 8080A chip as the 8080 chip is not compatible with the 8224 clock generator. Information on the timing of the logic signals and the description of the 8080A instruction set can be found in the <u>Intel 8080 Micro Computer Systems User's Manual</u>.

IMSAI

RAM 4A-4

RAM 4A BOARD

FUNCTIONAL DESCRIPTION

The IMSAI RAM 4A board provides up to 4K bytes of static random access memory. The board is implemented with 2102-style memory chips that each have the capacity to store 1024 words of one bit for each word. Thus eight chips are used to store one block of 1024 eight-bit words. Up to four sets of eight-chip units can be used on the board, giving a maximum capacity of 4096 eight-bit words.

Each eight-chip unit has the circuitry to allow or prevent the ability to write information into their memory storage space. This "write-protect" feature can be controlled either by software commands or from the computer front panel. Software commands can both affect the write protect and test the status of the write protect. If the program attempts to write into a write-protect block of memory, an interrupt will be generated. (This feature may be disabled if desired.) Four red LED's are provided to indicate the protect status of each of the 1K blocks of memory. Four green LED's are also provided which illuminate when their respective block of memory is addressed.

The RAM 4A board will support a front panel write protect switch. If the machine is stopped, the 1024 word block at which the machine address is pointing will have its memory write protect status affected through the use of a PROTECT/UNPROTECT switch on the front panel. Attempts to write into this section of the memory will, of course, not succeed.

The RAM 4A board is designed to allow the user to provide battery backup power. Trickle-charging facilities to allow the battery to be charged while the computer is running may also be installed on the board by the user.

The 8080A microprocessor can address up to 65,536 words of memory, thus allowing up to 16 4096 word RAM 4A-4 boards to be installed in one IMSAI 8080 system. (Additional memory can be accessed by using IMSAI's Shared Memory Facility.)

RAM 4A BOARD

THEORY OF OPERATION

The memory circuits used on the IMSAI RAM 4A memory board are 2102-style integrated circuits housed in sixteen pin DIP packages. Their organization is 1024 words, each of which is one bit wide. Ten address inputs are used to select the desired word and there is a chip enable to select the chip. There is a read/write input. One input is provided for data in, and one output is provided for data out. To implement the storage of data words that are eight bits wide, eight of the above described chips are used to store 1024 words. Three more of these eight chip groups can be used to give the IMSAI RAM 4A memory board a maximum storage capacity of 4096 eight bit words.

Bits A9, A8, A7, A6, A5, A4, A3, A2, A1, and A0 of the address bus come onto the memory board and go directly to the appropriate address pins on each memory chips. Bits A11 and A10 are decoded by a section of the 74LS156 at location D8 to select the desired 1024 word block by assertion of the chip enable signal for only those eight memory chips comprising the desired 1024 word block.

Bits A15, A14, A13, and A12 of the address bus are used to give each memory board on the bus a unique address. These bits first go through (if the memory board is involved in the utilization of its memory function through a memory-read operation, or memory write operation) the 74LS157 data selector at location D5. The direct output, and the complement of the direct output (obtained through the 74LS04 inverters at location C6) of the four output pins of the 74LS157 at location D5 go to DIP jumper provision at location C5. Provision is made so that either the equivalant polarity, or its complement, of the above mentioned four address bits can be implemented through the correct use of jumpers at location C5. When the polarity of the above-mentioned four address bits are in such an arrangement that they satisfy the address requirements of a particular memory board the four input pins of a section of the 74LS20 at location C4 will be high. This effects the selection of an individual memory board. Thus, only one board should respond in this manner for each of the sixteen different polarity arrangements of these four address bits.

Each 1024 word block of memory has its own circuitry to implement the write-protect feature. This feature is manipulated in two ways. One is from the "PROTECT/UNPROTECT" switch on the front panel. The other is from program commands contained in software.

There are four flip/flops whose two states enable or prevent the changing of the contents of their respective 1024 word blocks when a memory write is received. Each of these four flip/flops is a section of a 74LS74 at location C10 and at location C9. Memory block 0 is controlled by half of C9, memory block 2 is controlled by the other half of C10, and the other half of C9 controls memory block 3. The individual status of these four flip/flops is indicated by the desigated red light-emitting diodes located in the upper left hand corner of the board. If the red LED for a block is illuminated then that block is protected and writing into that block cannot occur. NOTE: A system reset will unprotect all blocks of memory.

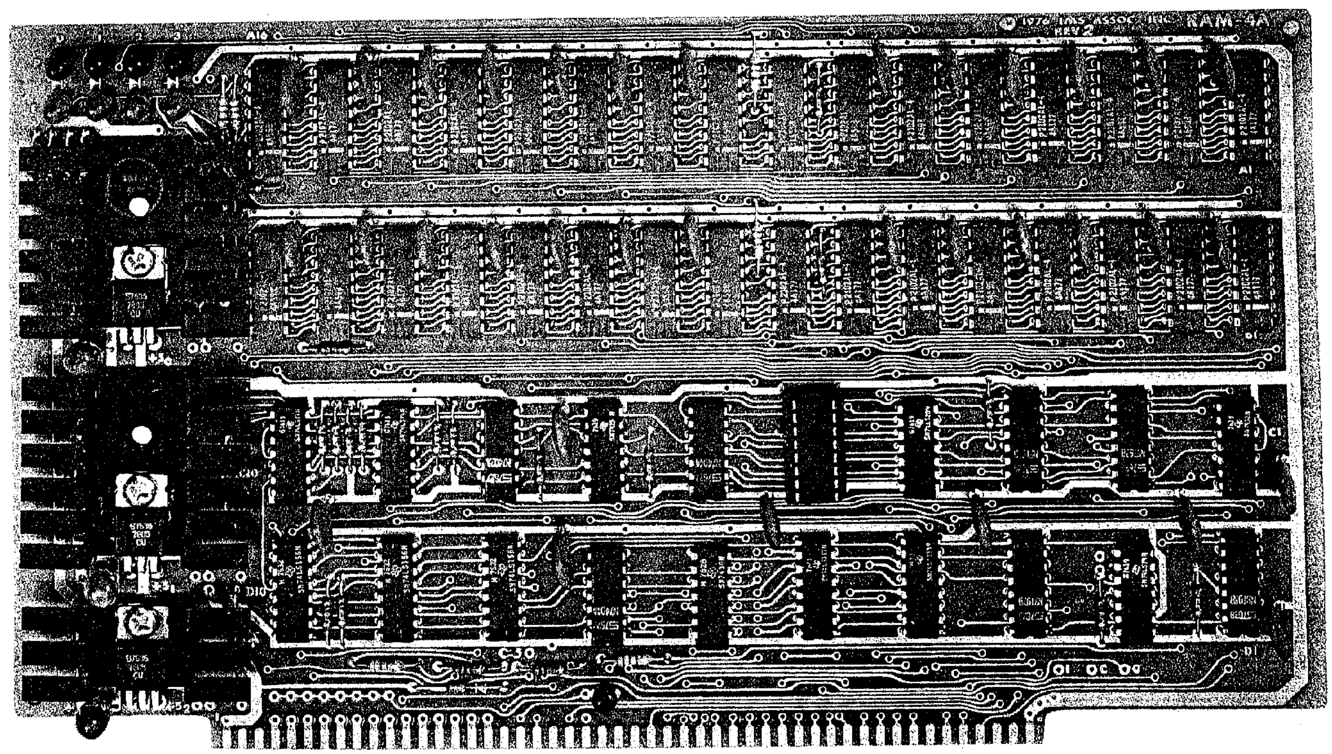
If a 1024 word block of memory is selected by its chip enable being decoded by the 74LS156 at location D8, and its respective write protect flip/flop at locations C10 or C9 are not in the protect state, then the section or the 74LS02 at location C8 associated with this block will have a high output. This high output, seen at the input of the 7425 at location C7, will cause the output of C7 to go low and this will assert one of the chip enable pins (pin 15) of the 74LS156 at location D10. The second chip enable of D10 is asserted on the PWR bus line; the second is an assertion on the MWRITE bus line. D10 will decode address bus bits A11 and A10 (as at D8) and issue a write pulse only to the selected 1024 word block.

The four write protect flip/flops at locations C10 and C9, as described earlier, are set and reset under the control of two sets of decoders whose outputs are wired ORed. One set, a section of the 74LS156 at location D10 that is used to set the flip/flops, and a section of the 74LS156 at location D8 that is used to clear (or reset) the flip/flops, is utilized when the protect/unprotect switch controls the assertion of the protect and the unprotect bus lines whose assertion is utilized via the chip enable input (pin 1) of D10 and D8. The other chip enable (pin 2) of both D10 and D8 is connected to the BDENA signal generated by the output (pin 8) of the 74LS20 at location C4. The two input lines to D10 and D8 that will be decoded to one of four output assertions are the address bus lines A11 and A10.

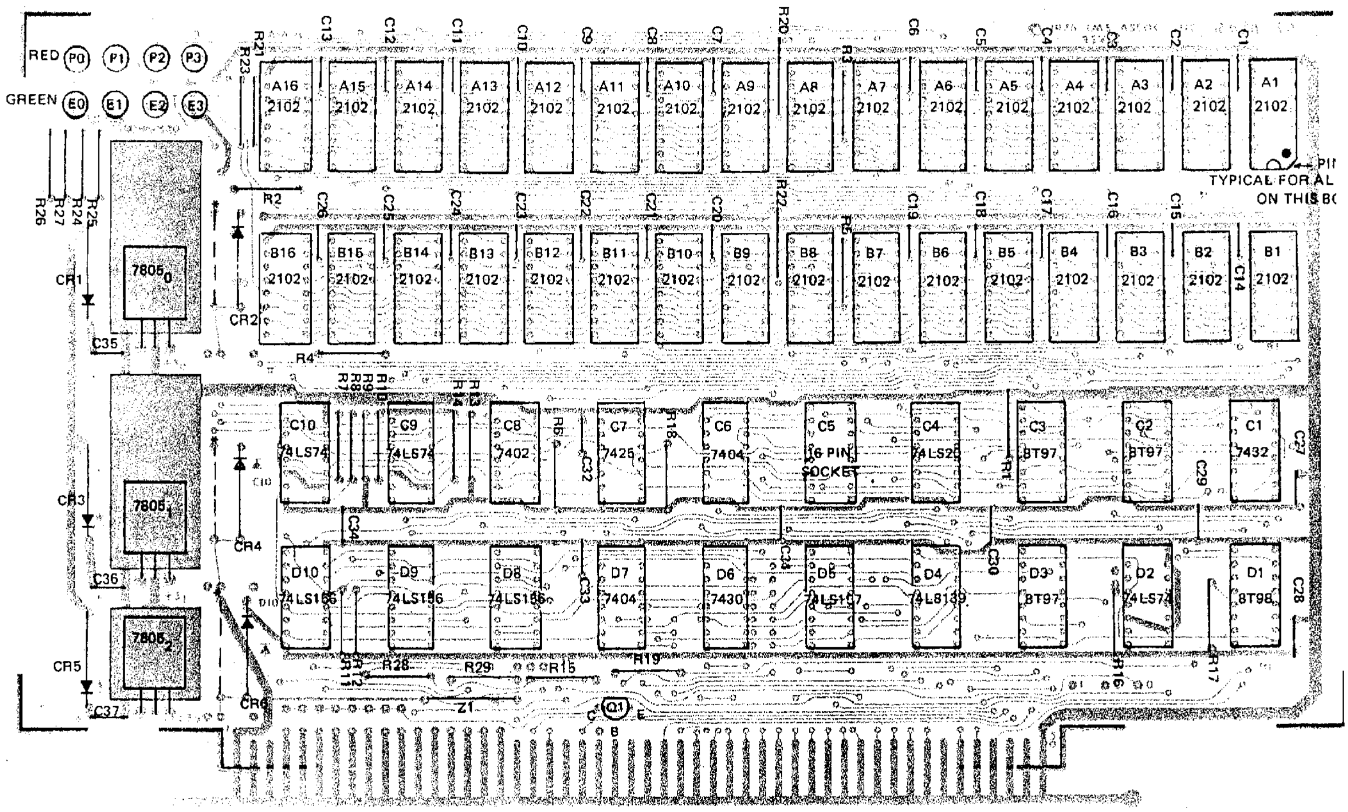
The other set of decoders are both sections of the 74LS156 at location D9. These are utilized when the four write protect flip/flops are going to have their status changed by programmed commands in the software. The command used is an output command, one of 256 available. The board is created to use output command FE, and only this one command is used for all (a maximum of 16) RAM 4A memory boards on a bus. The necessary board selection, and block selection, is done by putting board address (the same one as is used for board selections from the address bus-this feature is provided by the 74LS157 data selector at location D5), the two bits used to select one-of-four blocks of memory, and the two bits that are decoded to perform one-of-three actions, out on the system data bus at the time an FE output command bus is used. Two of the actions decoded by the 74LS139 at location D4 are the setting or the clearing (resetting) of the write protect flip/flop of the memory block as decoded from DO 3 and DO 2 by the 74LS156 one-of-four decoder at location D9.

The third action decoded from DO 1 and DO 0 by the 74LS139 one-of-four decoder at location D4 is the setting of the board select flip/flop, a section of the 74LS74 at location D2, which is used to select that board which puts data on the DATA IN (DI) bus when a data input FE command is issued so that the protect status can be read by the microprocessor. DI 0, DI 1, DI 2, and DI 3 carry the status of the write protect flip/flops for memory blocks 0, 1, 2 and 3. This status information is gated onto the DI bus through the 8T97 at location D3. The remaining four bits of the DATA IN bus, DI 4, DI 5, DI 6, and DI 7, carry the board address as set by the jumpers at location C5.

There is a flip/flop, a section of a 74LS74 at location D2, that becomes set if a write operation is attempted into a block of memory that is write protected. This flip/flop drives a transistor whose open collector output can be jumper connected to the INTERRUPT REQUEST (PINT) bus line pin 73, or to one of the vectored interrupt lines on bus pins 4 through 11. This interrupt notifies the user that a write has been attempted in a protected block of memory. The user may handle this interrupt with an interrupt routine.



RAM 4A–4 REV. 2



NOT SUPPLIED *

NOTE: THESE ARE USER DEFINED RESISTORS

SEE USER GUIDE SECTION FOR EXPLANATION

IMS ASSOCIATES INC.

ASSEMBLY DIAGRAM

RAM 4A–4 REV. 2 5/76

BOARD: RAM 4A

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Solder	15-0000001	10'	Solder
Heat Sink	16-0100003	1	3-Prong Heat Sink
Heat Sink	16-0100004	2	6-Prong Thermalloy Heat Sink
Screw	20-3302001	3	6-32x5/16" Phillips Pan Head Machine
Nut	21-3120001	3	6-32 Hex Nut
Lockwasher	21-3350001	3	#6 Internal Star Lockwashers
Header	23-0400001	1	16 Pin IC Header
Socket	23-0800001	1	16 Pin Solder Tail Socket
Resistor	30-3100362	1	100 Ohm, ¼ Watt/brown, black, brown
Resistor	30-3220362	10	220 Ohm, ¼ Watt/red, red, brown
Resistor	31-4100362	18	1K Ohm, ¼ Watt/brown, black, red
Capacitor	32-2010010	15 34	(For 1K) .1uF Disk Ceramic (For 4K)
Capacitor	32-2233070	1	33-25 Tantalum (or 22-25)
Diode	35-1000005	1	1N751-A Diode
Diode	35-1000007	6	1N4002 Rectifier Diode
LED	35-3000001	1 4	(For 1K) Red LED (For 4K)
LED	35-3000002	1 4	(For 1K) Green LED (For 4K)
8T97	36-0089701	3	Hex Tri-State Buffer/N8T97B
8T98	36-0089801	1	Hex Tri-State Buffer/N8T98B

RAM 4A
Parts List

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
2102	36-0210201	8	(For 1K) 1Kx1 Organization Static Memory Chip/P2101AL4
		32	(For 4K)
7402	36-0740201	1	Quad 2 Input NOR/DM7402N
7404	36-0740401	2	Hex Inverter/7404-N
74LS20	36-0472002	1	Dual 4 Input NAND (Low Power Schottky)/ SN74LS20N
7425	36-0742501	1	Dual 4 Input NOR with Strobe/SN7425N
7430	36-0743001	1	8 Input NAND/SN7430N
7432	36-0743201	1	Quad 2 Input OR/SN7432N
74LS74	36-0747402	3	Dual D Flip-Flop Preset and Clear (LPS)/SN74LS74
7805	36-0780501	3	5V Positive Volt Regulator/MC7805CP
74LS139	36-7413902	1	Dual 2 to 4 Line Decoder (LPS)/ SN74LS139N
74LS156	36-7415602	3	Open Collector (LPS)/ SN74156N
74LS157	36-7415702	1	Quad 2 to 1 Line Data Selector (LPS)/ SN74157N
PC Board	92-0000017	1	RAM 4A, Rev. 3

RAM 4A-4 Assembly Instructions

1) Unpack your board and check all parts against the parts lists enclosed in the package.

2) If gold contacts on the edge connector appear to be corroded, use pencil eraser to remove any oxidation. NOTE: Do not use Scotch-bright or any abrasive material as it will remove the gold plating.

RESISTOR INSTALLATION

3) Insert and solder each of the eighteen 1K ohm 1/4 watt resistors (brown/black/red) R1 through R18. See Assembly Diagram for location.

4) Insert and solder each of the ten 220 ohm 1/4 watt resistors (red/red/brown) R19 through R28. See Assembly Diagram for location.

5) Insert and solder the one 100 ohm 1/4 watt resistor (brown/black/brown) R29. See Assembly Diagram for location.

) Insert and solder each of six 1N4002 diodes, CR1 through CR6, as shown in the Assembly Diagram. NOTE: Observe polarity marks as indicated on board.

7) Insert and solder one 1N751A zener diode, Z1 observing polarity marks as shown on the board.

IC INSTALLATION

NOTE: All IC pin 1's point in the direction of the edge connector as indicated with the square solder pad in each hole pattern.

8) Insert and solder each of the three 74LS74 at locations C10, C9, a and D2.

9) Insert and solder each of the three 74LS156 at locations D8, D9, and D10.

10) Insert and solder each of the three 8T97 at locations C2, C3, and D3.

11) Insert and solder one 8T98 at location D1.

12) Insert and solder one 7402 at location C8.

13) Insert and solder one 74LS20 in location C4.

14) Insert and solder one 7425 at location C7.

15) Insert and solder two 7404 at location C6 and D7.

16) Insert and solder one 7430 at location D6.

17) Insert and solder one 74LS157 at location D5.

18) Insert and solder one 74LS139 at location D4.

19) Insert and solder one 7432 at location C1.

20) Insert and solder each of the eight 2101 memory chips at locations A9 through A16 for 1K RAM Board and each of the thirty-two 2102 memory chips at locations B1 through B16 for 4K RAM Board.

DISCRETE COMPONENT INSTALLATION

21) Insert and solder the 16 pin IC socket located at C5 and plug in the 16 pin jumper header. (This jumper header is used for board addressing).

22) Insert and solder one 2N3904 transistor at location Q1 as shown on the Assembly Diagram. NOTE: Observe orientation as shown on the Assembly Diagram.

23) Insert and solder each of the fifteen .1uF capacitors at locations C7 through C13 and C27 through C34 for 1K RAM Board and each of thr thirty-four .1uF capacitors at location C1 through C34 for 4K RAM Board as shown on the Assembly Diagram.

24) Insert and solder each of the three 33uF 25 volt tantalum capacitors at locations C35 through C37 as shown on the Assembly Diagram. NOTE: Observe polarity as shown on board.

25) Insert and solder one red LED at location P0 for 1K RAM Board and each of the four red LED's at locations P0 through P3 for 4K RAM Board as shown on the Assembly Diagram.

26) Insert and solder one green LED at location E0 for 1K RAM Board and each of the four green LED's at location E0 through E3 for 4K RAM Board as shown on the Assembly Diagram. NOTE: The LED's should be positioned so that the flat side of the cathode is to the right.

REGULATOR AND HEAT SINK INSTALLATION

27) Take each of the three 7805 regulators and bend the leads at 90 degree angles approximately ¼" from the bottom edge of the regulator to facilitate insertion on top of the heat sink.

28) The smallest heat sink is used near the bottom of the board, closest to the edge connector. Insert the #6 screw and lockwasher through the regulator and heat sink and tighten with the nut on the back side of the board. Repeat this procedure with the two remaining heat sinks and solder each of the regulator leads in place. NOTE: Be sure to hold the screw in order to prevent shorting to adjacent traces.

29) Add jumper wires for desired address onto the jumper header. (See User Guide Section). This indicates the address of the board.

JUMPER OPTIONS

30) A) Using clipped resistor leads (or bus wire) to select 0 wait states, jumper hole (C) to hole (0).

B) For one wait state, jumper hole (C) to hole (1). These holes are located on the board directly below locations D2 and D3.

31) The select interrupt jumper may be installed after reading the User Guide Section and after determining which vectored interrupt is desired.

USER GUIDE

Board Selection

In memory read or memory write operation (as well as responding to the output or input commands of FE) the IMSAI RAM 4A memory board is designed to be selected as one out of a maximum possible of sixteen RAM 4A memory boards present on the bus. To achieve this one-of-sixteen selection, the top four address lines--A15, A14, A13 and A12 in the case of a memory read or memory write operation (or the top four data out lines (DO 7, DO 6, DO 5 and DO 4) in the case of an output or input FE instruction)--are decoded on the board via the positioning of the jumpers installed at location C5 to give each memory board its unique address. These jumpers are implemented so as to route the logic 1 polarity of the above described four lines, or the complements of their polarities, in such a manner that when a board's unique address is present on the above described lines the four inputs to the 74LS20 four input NAND gate at C4 will all be high.

This will make the output (pin 8) go low and will assert the board enable (BDENA) line on the board. If the logic 1 polarity is desired then the jumper for that bit should route the output of the 74LS157 at location D5 direct to the input of the 74LS20 at location C6, associated with that bit shall be routed to the input of the 74LS20 at location C4.

TABLE 1

ADDRESS BIT	DIP POSITION C5		JUMPERING
A15	Pin 9	Pin 8	Place jumper between pins 9 and 8 if the board is to be selected when this bit is high.
	Pin 10	Pin 7	Place jumper between pins 10 and 7 if the board is to be selected when the above bit is low.
A14	Pin 11	Pin 6	Place jumper between pins 11 and 6 if the board is to be selected when this bit is high.
	Pin 12	Pin 5	Place jumper between pins 12 and 5 if the board is to be selected when the bit is low.
A13	Pin 13	Pin 4	Place jumper between pins 13 and 4 if the board is to be selected when this bit is high.
	Pin 14	Pin 3	Place jumper between pins 14 and

			3 if the board is to be selected when the above bit is low.
	Pin 15	Pin 2	Place jumper between pins 15 and 2 if the board is to be selected when this bit is high.
A12			
	Pin 16	Pin 1	Place jumper between pins 16 and 1 if the board is to be selected when this bit is low.

Hardware Write Protect

If memory PROTECT/UNPROTECT from a switch (located on the front panel or elsewhere) is to be used, jumper D8 pin 1 to I/O pin 20. In cases where a switch will not be used and I/O pin 20 is driven high (such as in the IMSAI CP-A Revision 4 or earlier front panel assembly), D8 pin 1 should be jumpered to ground. Jumper pads are provided to accomodate either case. Refer to Figure 1 for details.

Memory is protected in 1K blocks. With the computer front panel in the stop mode, the switch will affect whichever block contains the address being displayed. To protect or unprotect any block, examine any word in that block and actuate the switch. The memory protect light on the front panel will indicate the protect status of the addressed block.

A system reset will unprotect all blocks of memory.

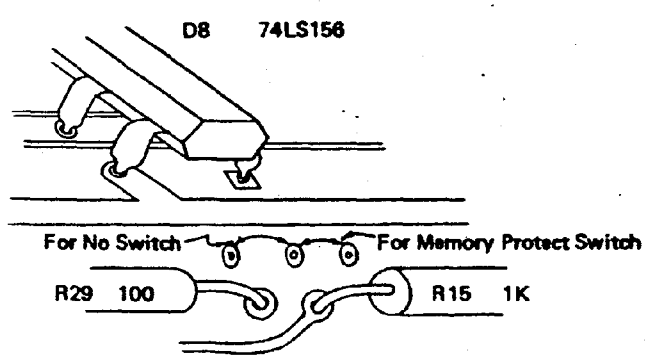


FIGURE 1

Software Write Protect

1K blocks of memory may be write protected or unprotected with an OUT command to port FE*. Selection of memory board and block is selected with the high-order 6 bits in the output data word. Bits 0 and 1 select the function (Protect, Unprotect, Select Board for Status or Clear Interrupt). Bits 2 through 7 should be the same as bits 10 through 15 of the memory address of the desired 1K block. Bits 4 through 7 select the memory board and bits 2 and 3 select the 1K block on that board. Refer to Table 2 for bit functions.

TABLE 2

Output Data Bit			
7	Board Select	Same as Memory Address Bits for desired 1K Block	15
6			14
5			13
4			12
3	1K Block Select		11
2			10
1	Function Select		
0			

Bit 0	Bit 1	
0	0	Clear Interrupt
1	0	Unprotect Addressed Block
0	1	Protect Addressed Block
1	1	Select Board for Status Read

The output command to select a board for status read must be issued before each status read. This enables the selected board to respond with status to the next INP command from port FE. The board automatically deselects after responding to the INP command. Care should be taken not to select more than one board before reading the status or the boards will interfer with each other. Refer to Table 3 for the meaning of the status data bits.

* This address may be changed if desired by using the inverters in C7 (pins 1, 2, 3, 4). Cut the trace to the desired input pins to D6 and solder jumpers to the spare inverter. The inverter line A8 may similarly be removed and placed in another bit. Be sure to reconnect bit A8.

TABLE 3

STATUS READ

Data Bit		
7	Same as Address Bits (Board Address)	15
6		14
5		13
4		12
3	Block 3	1=Unprotected 0=Protected
2	Block 2	
1	Block 1	
0	Block 0	

The Interrupt Request flip/flop is set by an attempt to write into a protected location. (The data in memory will not be affected.) In addition to requesting an interrupt (if jumpered appropriately) the Interrupt Request flip/flop enables the board to respond to the next Status Read (INP FE). The bit definitions are the same as a normal status read, which indicates what board is affected and which 1K blocks on that board are protected. The Interrupt Request flip/flop is reset by the appropriate output command. See Table 3.

Because of the possible conflict during a status read if the Interrupt Request flip/flop is set between a board select and the following Status Read, it is suggested that all status reads be performed by a subroutine which disables interrupts, selects a board, reads its status, enables interrupts and returns.

To obtain the Interrupt Request feature, a jumper must be installed to connect the RAM 4A to the desired Priority Interrupt line on the back plane. Figure 2 illustrates the placement of this jumper.

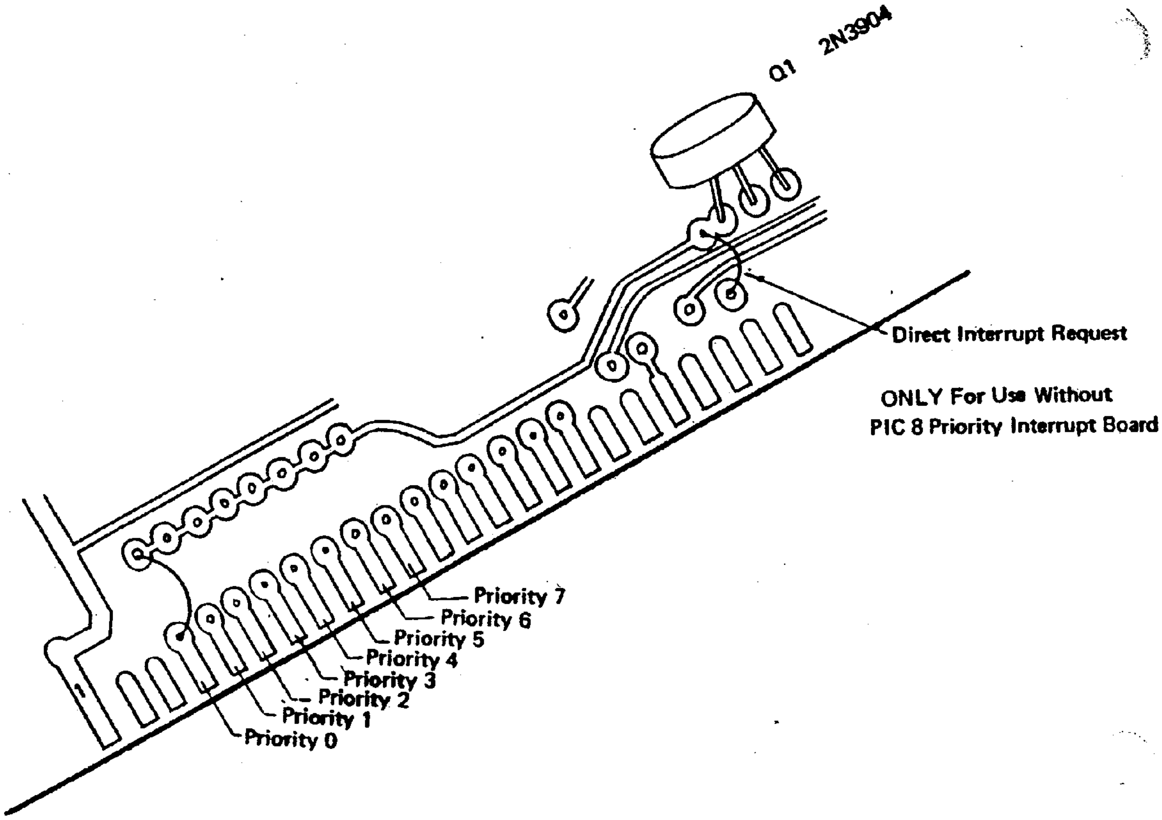

Q1 2N3904
Direct Interrupt Request
ONLY For Use Without
PIC 8 Priority Interrupt Board
Priority 7
Priority 6
Priority 5
Priority 4
Priority 3
Priority 2
Priority 1
Priority 0

FIGURE 2

If it is desired to prevent the Interrupt Request flip/flop from being set (e.g., to avoid conflict with status reads if interrupts are not being used), cut the flip/flop line between the two pads to the left of D2 on the solder side (see Figure 3).

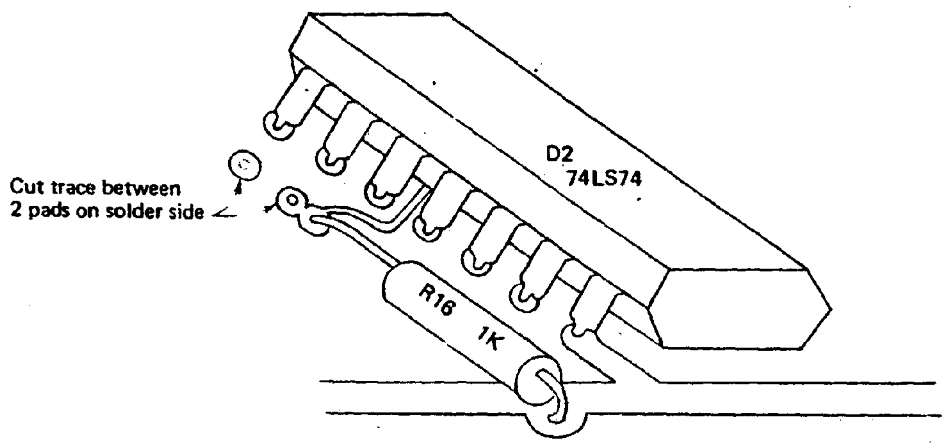


FIGURE 3

Wait Cycle Selection

No wait cycle is required for the memory chips supplied with the RAM 4A board. One wait cycle may be required if slower memory chips are substituted. Selection of the wait cycle option (zero or one wait cycle) is illustrated in Figure 4.

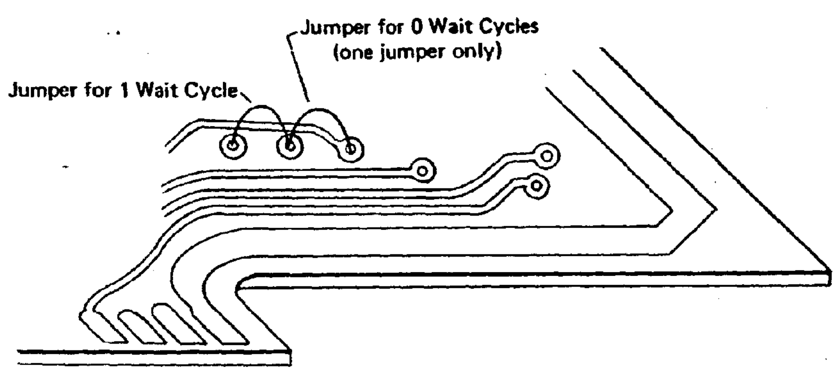


FIGURE 4

Battery Backup Operation

For operating your RAM 4A board with battery backup, simply connect your battery to the board at the location indicated on the Battery Hookup Diagram.

The battery should deliver 3 to 5 volts DC and should supply 300 milliamps of current.

A user defined resistor may be installed on the board to facilitate recharging the battery while the computer is turned on. (See Assembly Diagram for location.)

As an example for picking the value resistor that should be used to supply the trickle charge to your battery:

For a back plane voltage of	(+8V)	I=E/R
and a battery voltage of	- (+3V)	=5V/220 ohms
	(+5V)	=.0227 Amps

A resistor of 220 Ohms will supply approximately 20 ma. current as trickle charge to your battery.

It is also recommended that if you do not intend to use battery back-up, remove the three diodes in the input circuit of the three regulators and replace them with jumper wires. This will allow the board to function with a Mother board voltage of 7 volts DC rather than 7.7 volts DC.

System Features Test

The special functions of this memory board far exceeds the functions of any other memory board on the market today and, because of this, is going to take a little time for the user to understand all its capabilities. A NOTE OF CAUTION: One common mistake that is made when using this board is protecting a block of memory where you may have placed your stack.

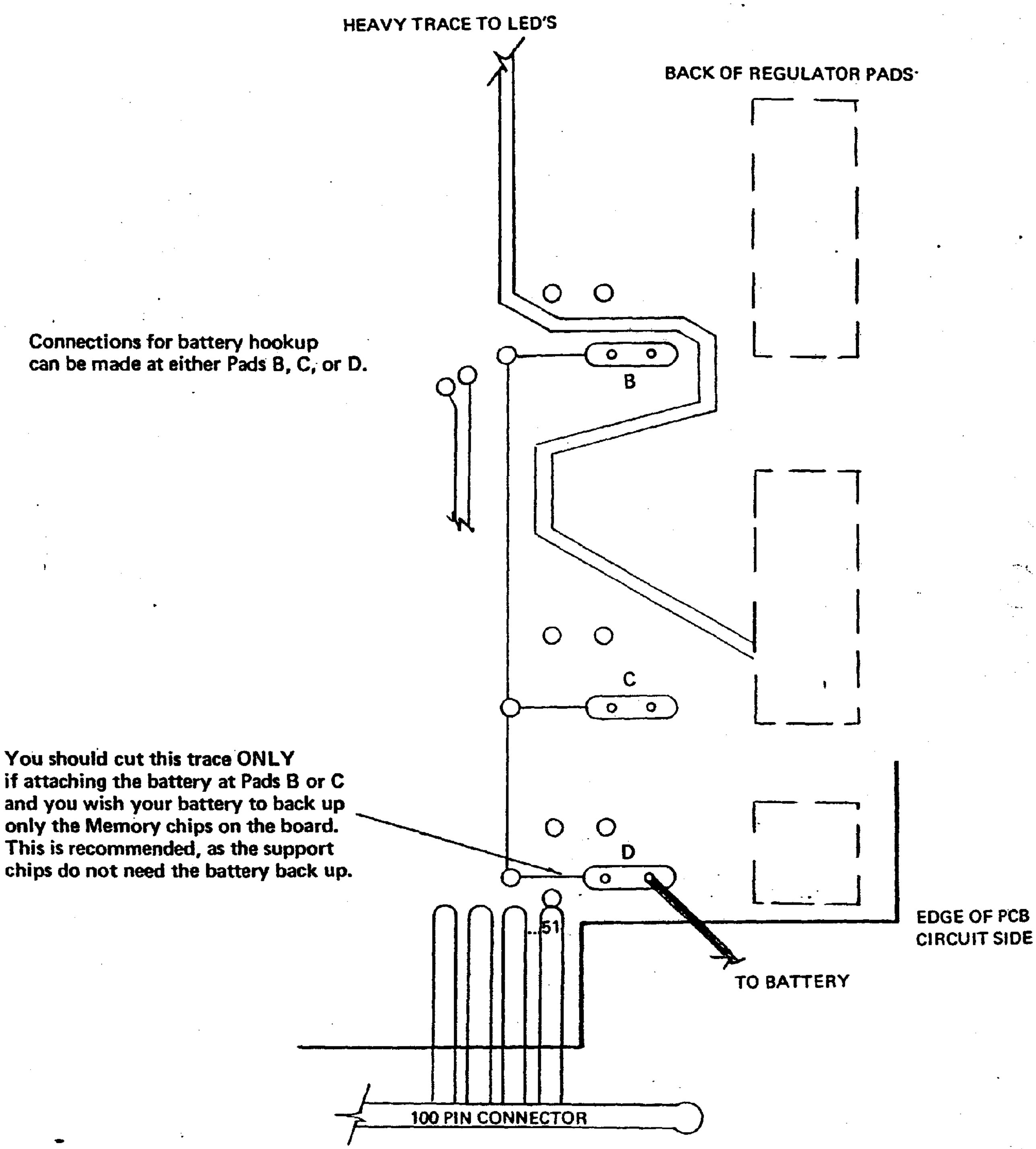
A simple test program for testing some of the special features of your new RAM 4A board follows:

Address	Instruction	Description
00	DB	INPUT
01	FF	FROM FRONT PANEL SWITCHES
02	D3	OUTPUT
03	FE	TO MEMORY BOARD
04	DB	INPUT
05	FE	FROM MEMORY BOARD
06	D3	OUTPUT
07	FF	TO FRONT PANEL LIGHTS
08	C3	JUMP
09	00	TO
0A	00	0

This simple test program allows the operator to output protect and unprotect commands to the memory board under test when the memory board is addressed at location 00 hex, by using the sense switches on the front panel (high address switches). The program resides in the first 1K block of memory of the board that is actually under test.

The interrupt feature of the board may be tested by inserting a store accumulator (32 hex) instruction before the jump to 00 hex. By locating the address of where the data is to be stored in various 1K blocks of memory, an interrupt will be generated when that particular 1K block is given a protected status either from front panel switches or from software. NOTE: Interrupts should be enabled in your program.

RAM 4A BATTERY HOOKUP DIAGRAM



The 4K board tester is at PROM location 0400H. The 1K tester is at 0500H.

TO USE:

1. Jumper the board to be tested to respond to addresses FxxxH.
2. Insert the board in an 8080 with CPU-A, CP-A and PROM containing the test routine.
3. Power the 8080 up. Set the switches to 0400H, press EXAMINE and press RUN.
4. The test routine will run, "Messages" are displayed in the 8 LED's labelled "programmed output" in the upper left corner of CP-A.

MESSAGES:

LED Display (Hex)	(Binary)	Meaning
01	00000001	Running Phase I test - no errors yet
02	00000010	Running Phase II test - no errors yet
03	00000011	Running Phase III test - no errors yet
F1	11110001	Error in Phase I: data will follow
F2	11110010	Error in Phase II: data will follow
F3	11110011	Error in Phase III: data will follow
FF	11111111	Test completed without errors: change any "programmed input" switch (#'s 8-15) to start test over.

ERROR PROCESSING:

When an error occurs, a "message" of F1, F2, or F3 will be displayed on the LED's. To get information on the errors:

1. Change <u>one</u> of switches 8-15.
2. The LED's will display the high 8 bits of the address at the location that failed.
3. Change <u>one</u> of switches 8-15.
4. The LED's will display the low 8 bits of the address.
5. Change <u>one</u> of switches 8-15.
6. The LED's will display the data that the location is supposed to contain.
7. Change <u>one</u> of switches 8-15.
8. The LED's will display the data the location actually contains.
9. Change <u>one</u> of switches 8-15.
10. The test will start over with Phase I.

INTERPRETATION OF ERRORS:

Phase I simply verifies that every location in RAM will correctly preserve data. The procedure is:

1. Write '00' in location F000.
2. Read location F000 and ensure that it is '00'.
3. Repeat 1-2 using values '01', '02',....'0F' and '10', '11',....'FF'.
4. Repeat 1-3 on F001, F002,....FFFF.

If an error occurs in Phase I, it indicates one of two hardware problems: a) a bad chip on the RAM board, or b) a bad data line (D0-D7) from the CPU to the RAM chip. The chip and a data line involved can be determined from the error data. Generally, case (b) will affect all locations in a chip or on the entire board, while case (a) will affect one locaiton or all locations on the chip. The cases can be distinguished by playing with DEPOSIT/ EXAMINE and chip replacement.

Phases II and III are actually two parts of the same test. Phase I has already determined that location "n" (F000/n/ FFFF) can hold data correctly (at least for a few microseconds). However, we have not yet proved that "n" references a unique location. Phases II and III verify this (and, in passing, prove that the RAM can hold a value for at least a few milliseconds).

Consider a RAM board in which address line is messed up in such a way that RAM always sees it as 0, regardless of its true state. Then RAM addressing will look like this:

```
True location F000 <--Referencing F000 will access this location
 "       "    F001 \_      "      F001  "     "     "      "
 "       "    F002 <--     "      F002  "     "     "      "
 "       "    F003 /       "      F003  "     "     "      "
 ↓       ↓     ↓           ↓       ↓    ↓     ↓     ↓      ↓
True location FFFE <--Referencing FFFE will access this location
 "       "    FFFF \_      "      FFFF  "     "     "      "
```

Phase II will not detect this error.

It will write 00 through FF into F000 which it thinks is location F001. Since this actually accesses F000, the data will be read back correctly. So Phase I will succeed. Now comes Phase III. This starts by writing the low 8 bits of the address of each location into that location, i.e., 00 into FD20, 01 into F001,...., FF into FFFF. Then it goes back and reads this data, verifying it. Let's watch what happens with our bad address line.

TRUE LOC.	CONTENTS	RESPONDS TO:
F000	00	F000, F001 Step 1: Write 00 into F000
F001	?	nothing
F000	01	F000, F001 Step 2: Write 01 into F001
F001	?	nothing
E000	1	F000, F001 Step 4097: Read F000,
F001	?	expecting nothing - and detect an error.

Thus, Phase III detects our error. Now for some observations on how to find the error.

1. Between steps 1 and 4097, several milliseconds pass without accessing location F000. If RAM is volatile, the data in F000 could go away and generate a Phase III error. This can be found by DEPOSITing into the bad location and EXAMINEing it to see if it changes. The reason Phase I doesn't catch this is that it reads 3.5 µs after it writes, so the data doesn't have time to deteriorate.

2. If address line 0 were stuck at 1, the same results would appear in Phase III. (Try it.) You can't tell from this test what the line is stuck at.

3. If Phase II or III fails, the bad address bits are the ones where the "supposed to be" data and the "read back" data differ. If the error was Phase II, these represent the high 8 bits of address. If the error was Phase III, these represent the low 8 bits.

```
:DBUG
IMSAI 8Ø8Ø DEBUGGER Ø4/Ø5/76

*Ø4ØØ,Ø4FF;
Ø4ØØ    F3 3E FE D3   FF 21 ØØ FØ   AF 77 46 B8   C2 56 Ø4 3C
Ø41Ø    C2 Ø9 Ø4 23   B4 C2 Ø8 Ø4   3E FD D3 FF   21 ØØ FØ 74
Ø42Ø    23 AF B4 C2   1F Ø4 21 ØØ   FØ 7E 94 C2   7C Ø4 23 B4
Ø43Ø    C2 29 Ø4 3E   FC D3 FF 21   ØØ FØ 75 23   AF B4 C2 3A
Ø44Ø    Ø4 21 ØØ FØ   7E 95 C2 88   Ø4 23 B4 C2   44 Ø4 3E FF
Ø45Ø    21 ØØ Ø4 C3   94 Ø4 EB 4F   21 6Ø Ø4 3E   F1 C3 94 Ø4
Ø46Ø    7A 21 67 Ø4   C3 94 Ø4 7B   21 6E Ø4 C3   94 Ø4 79 21
Ø47Ø    75 Ø4 C3 94   Ø4 78 21 ØØ   Ø4 C3 94 Ø4   EB 82 47 4A     4K RAM TEST
Ø48Ø    3E F2 21 6Ø   Ø4 C3 94 Ø4   EB 83 47 4B   3E F3 21 6Ø     ENTRY: 0400
Ø49Ø    Ø4 C3 94 Ø4   2F D3 FF F9   DB FF 67 DB   FF AC CA 9B
Ø4AØ    Ø4 21 18 FC   23 AF B4 C2   A4 Ø4 21 ØØ   ØØ 39 E9 FF
Ø4BØ    FF FF FF FF   FF FF FF FF   FF FF FF FF   FF FF FF FF
Ø4CØ    FF FF FF FF   FF FF FF FF   FF FF FF FF   FF FF FF FF
Ø4DØ    FF FF FF FF   FF FF FF FF   FF FF FF FF   FF FF FF FF
Ø4EØ    FF FF FF FF   FF FF FF FF   FF FF FF FF   FF FF FF FF
Ø4FØ    FF FF FF FF   FF FF FF FF   FF FF FF FF   FF FF FF FF
*Ø5ØØ,Ø5FF;
Ø5ØØ    F3 3E FE D3   FF 21 ØØ FØ   AF 77 46 B8   C2 5B Ø5 3C
Ø51Ø    C2 Ø9 Ø5 23   7C FE F4 C2   Ø8 Ø5 3E FD   D3 FF 21 ØØ
Ø52Ø    FØ 74 23 7C   FE F4 C2 21   Ø5 21 ØØ FØ   7E 94 C2 81
Ø53Ø    Ø5 23 7C FE   F4 C2 2C Ø5   3E FC D3 FF   21 ØØ FØ 75
Ø54Ø    23 7C FE F4   C2 3F Ø5 21   ØØ FØ 7E 95   C2 8D Ø5 23
Ø55Ø    7C FE F4 3E   FF 21 ØØ Ø5   C3 99 Ø5 EB   4F 21 65 Ø5
Ø56Ø    3E F1 C3 99   Ø5 7A 21 6C   Ø5 C3 99 Ø5   7B 21 73 Ø5
Ø57Ø    C3 99 Ø5 79   21 7A Ø5 C3   99 Ø5 78 21   ØØ Ø5 C3 99     1K RAM TEST
Ø58Ø    Ø5 EB 82 47   4A 3E F2 21   65 Ø5 C3 99   Ø5 EB 83 47     ENTRY: 0500
Ø59Ø    4B 3E F3 21   65 Ø5 C3 99   Ø5 2F D3 FF   F9 DB FF 67
Ø5AØ    DB FF AC CA   AØ Ø5 21 18   FC 23 AF B4   C2 A9 Ø5 21
Ø5BØ    ØØ ØØ 39 E9   FF FF FF FF   FF FF FF FF   FF FF FF FF
Ø5CØ    FF FF FF FF   FF FF FF FF   FF FF FF FF   FF FF FF FF
Ø5DØ    FF FF FF FF   FF FF FF FF   FF FF FF FF   FF FF FF FF
Ø5EØ    FF FF FF FF   FF FF FF FF   FF FF FF FF   FF FF FF FF
Ø5FØ    FF FF FF FF   FF FF FF FF   FF FF FF FF   FF FF FF FF
*
```

Errata
2/4/77

RAM 4A-4

NOTE: The RAM 4A Chapter applies to both RAM 4A-4, Rev. 2 <u>and</u> RAM 4A-4, Rev. 3.

IMSAI

PROM-4

THEORY OF OPERATION

The PROM-4 board provides up to 4K of addressable Read-Only-Memory, utilizing the Intel 8702-1702 PROM devices. The board contains 256 bytes of memory for each 8702-1702 chip installed.

Address lines A0 through A7 are run directly to all PROM positions to select one of the 256 internal byte positions, while address lines A8 through A11 are used to select and enable one particular PROM position through 8205 decoders. Address lines A12 through A15 are jumper-selected to determine the board's enabling address.

The board is enabled when the 74LS30 NAND (C1) inputs are all high, namely when the selected address appears on the address bus, and the Status line SMEMR is high. The Processor Ready line is controlled by a 74195 shift register via an 8T97. The 74195 provides a user-selected memory read delay, selectable with jumpers in the delay select socket. The 74195 shift register is reset on the rising edge of the inverted Board Enable (BDENA) signal.

When addressed and enabled, an 8702-1702 PROM puts out its data on the D0 through D7 lines. The data output lines of all PROMS are tied to these lines, and these lines are buffered via 8T97 sections to the DI0 through DI7 back plane bus lines.

Power for the card logic is provided by a +5 volt regulator and a -5 volt regulator-4 volt zener combination to yield +5 and -9 volts. Tantalum and disc ceramic by-pass capacitors eliminate noise from the power distribution busses.

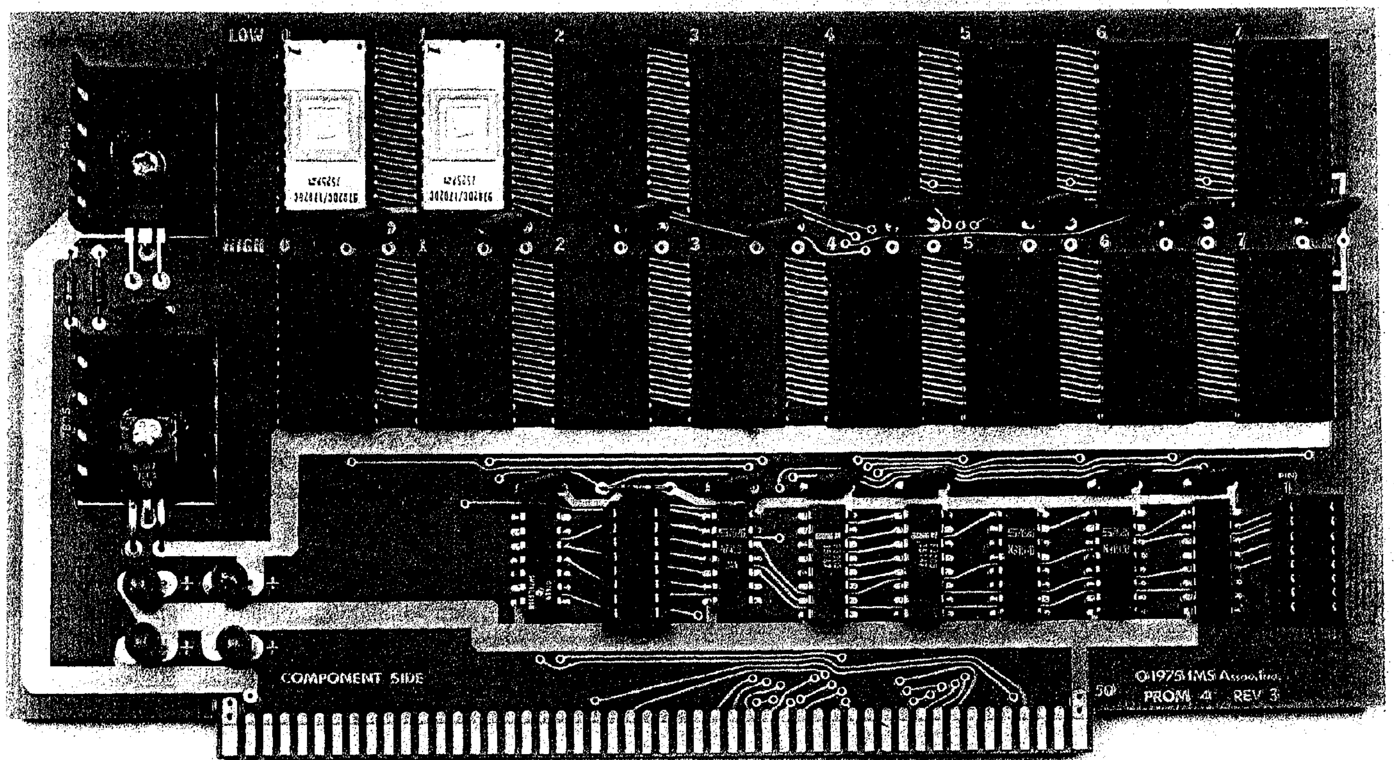
FUNCTIONAL DESCRIPTION

The IMSAI PROM-4 Board supports up to 4K bytes on non-volatile, read-only memory. Designed to utilize the Intel 1702 or 8702 read-only memory devices, the PROM-4 board may be flexibly configured to contain up to 4K bytes in 256 byte increments. The board address can be switch or jumper-selected to any 4K block of the computer's 64K memory space. Tri-state bus drivers and fully-de-coupled on-card voltage regulators provide reliable plug-in compatibility with the IMSAI 8080 (S100).

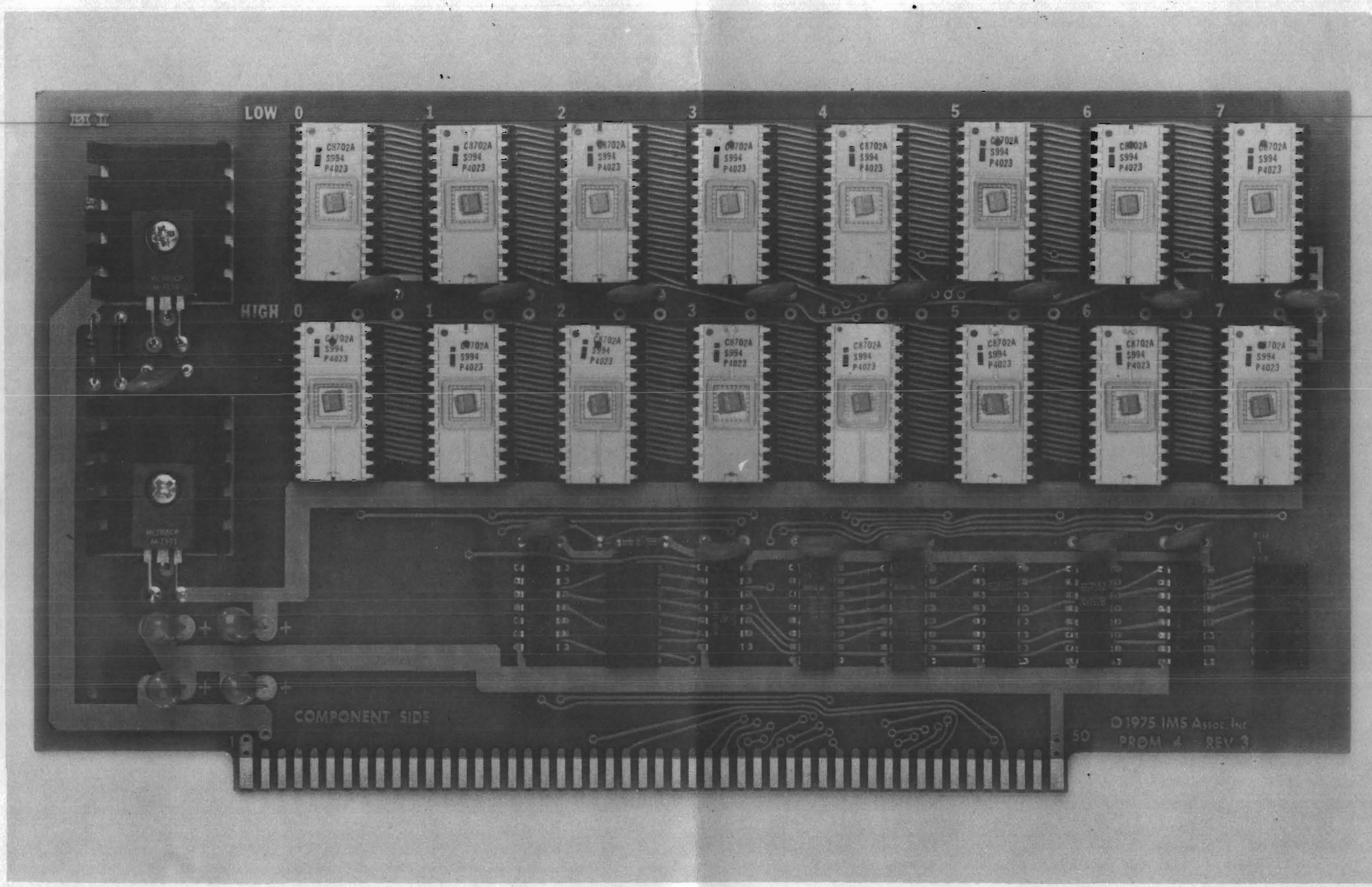
The PROM-4 board provides sockets for 16 1702 or 8702 PROMs. The socket locations are marked for easy selection of PROM addresses. A user-selectable memory read delay allows efficient use of fast or slow PROM devices. (Please consult the User's Guide for additional information about this feature.) Two on-card regulators provide the +5 and -9 volts required by the 1702-8702 chips.

The PROM 4-4 board contains a full 4K block of PROM (8702 or 1702). PROM 4-512 contains 512 bytes of PROM (two 8702's or 1702's) which may be expanded through the use of an expansion module MM702-5. Each expansion module provides an additional 512 bytes of PROM capability.

Physically, the PROM-4 board is G-10 equivalent, 1/16" thick glass fiber reinforced laminate. Plated through-holes eliminate jumpers, and the edge connector contact fingers are gold plate over nickel for reliable contact and long life. The board measures 5" x 10" and uses the standard 100 pin edge connections to the back plane. Discrete components are of the highest quality with tantalum by-pass and ceramic de-coupling capacitors. Both on-card voltage regulators are fully protected against short circuits and thermal overloads.



PROM 4–512 REV 3


LOW
HIGH
0
1
2
3
4
5
6
7
COMPONENT SIDE
50
©1975 IMS Assoc. Inc
PROM 4 REV 3

PROM 4-4 REV. 3

BOARD: PROM 4

ITEM	IMSAI PART #		QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Solder	15-0000001		10'	
Heat Sink	16-0100002		2	Thermalloy Heat Sink/6106B-14
Screw	20-3402001		2	6-32x3/8" Phillips Pan Head Machine Screw
Nut	21-3120001		2	6-32 Hex Nut
Lockwasher	21-3350001		2	#6 Internal Star Lockwasher
Header	23-0400001		2	16 Pin IC Header
Socket	23-0800001		2	16 Pin Solder Tail Socket
Socket	23-0800002		16	24 Pin Solder Tail Socket
Resistor	30-4100362		2	1K Ohm, ¼Watt Resistor/brown, black, red
Capacitor	32-2010010		15	.1uf Disk Capacitor
Capacitor	32-2233070		4	33-25 Tantalum Capacitor (or 22-25)
Diode	35-1000004		1	Zener Diode (brown, violet)/1N748
8T97	36-0089701		2	Hex Tri-State Buffer/N8T97B
74LS04	36-0740402		1	Hex Inverter (Low Power Schottky)/SN74LS04N
74LS30	36-0743002		1	8 Input NAND (Low Power Schottky)/SN74LS30N
7805	36-0780501		1	5 V. Positive Voltage Regulator/MC7805CP
7905	36-0790501		1	5 V. Negative Voltage Regulator/MC7905CP
8205	36-0820501		2	Binary Decoder/8205
1702a	36-0870201	(for 4-4) (for 4-512)	16 2	Programmable Read Only Memory (256x8)/White and Gold Chip /C8702A
74195	36-7419501		1	4-Bit Parallel I/O Shift Register/SN74195N
PC Board	92-0000014		1	Printed Circuit Board/PROM 4

ASSEMBLY INSTRUCTIONS

( ) 1. Unpack your board and check all parts against the parts list enclosed in the package.

( ) 2. If gold contacts on the edge connector appear to be corroded, use pencil eraser to remove any oxidation. NOTE: Do not use Scotchbright or any abrasive material as it will remove the gold plating.

RESISTOR INSTALLATION

( ) 3. Insert and solder each of the two 1K ohm, ¼ watt resistors (brown, black, red) at locations R1 and R2 as shown on the Assembly Diagram.

( ) 4. Insert and solder the 3.9 volt zener diode (brown/violet) at location Z1 as shown on the Assembly Diagram.

DISCRETE COMPONENT INSTALLATION

( ) 5. Insert and solder each of the two 16 pin IC sockets at locations C2 and C9 as shown on the Assembly Diagram. Plug in the jumper headers. See User Guide section for jumper wire connections(s).

( ) 6. Insert and solder each of the fifteen .1uF capacitors at locations C1 and C6 through C19 as shown on the Assembly Diagram.

( ) 7. Insert and solder each of the three 33uF capacitors at locations C2 through C5 as shown on the Assembly Diagram.
NOTE: Observe polarity (+ to +) as shown on the board.

IC INSTALLATION

NOTE: All IC Pin 1's point toward the upper left hand corner as noted on the board.

( ) 8. Insert and solder the one 74LS04 at location C3.

( ) 9. Insert and solder the one 74LS30 at location C1.

( ) 10. Insert and solder the one 74195 at location C8.

( ) 11. Insert and solder the two 8205's at locations C4 and C5.

( ) 12. Insert and solder the two 8T97's at locations C6 and C7 as shown on the Assembly Diagram.

( ) 13. PROM 4-4:
Insert and solder the 16 24-pin sockets at locations A1 through A8 and B1 through B8. Insert the 1702A's (or 8702A's) into their appropriate locations. See the User Guide section, Table 1 for these locations.

PROM 4-512:
Determine the appropriate locations for the 512 bytes of PROM by consulting the User Guide (Table 1). Then insert and solder the two 24-pin sockets into the selected locations. Finally, insert the two 1702A's (or 8702A's) into their respective sockets.

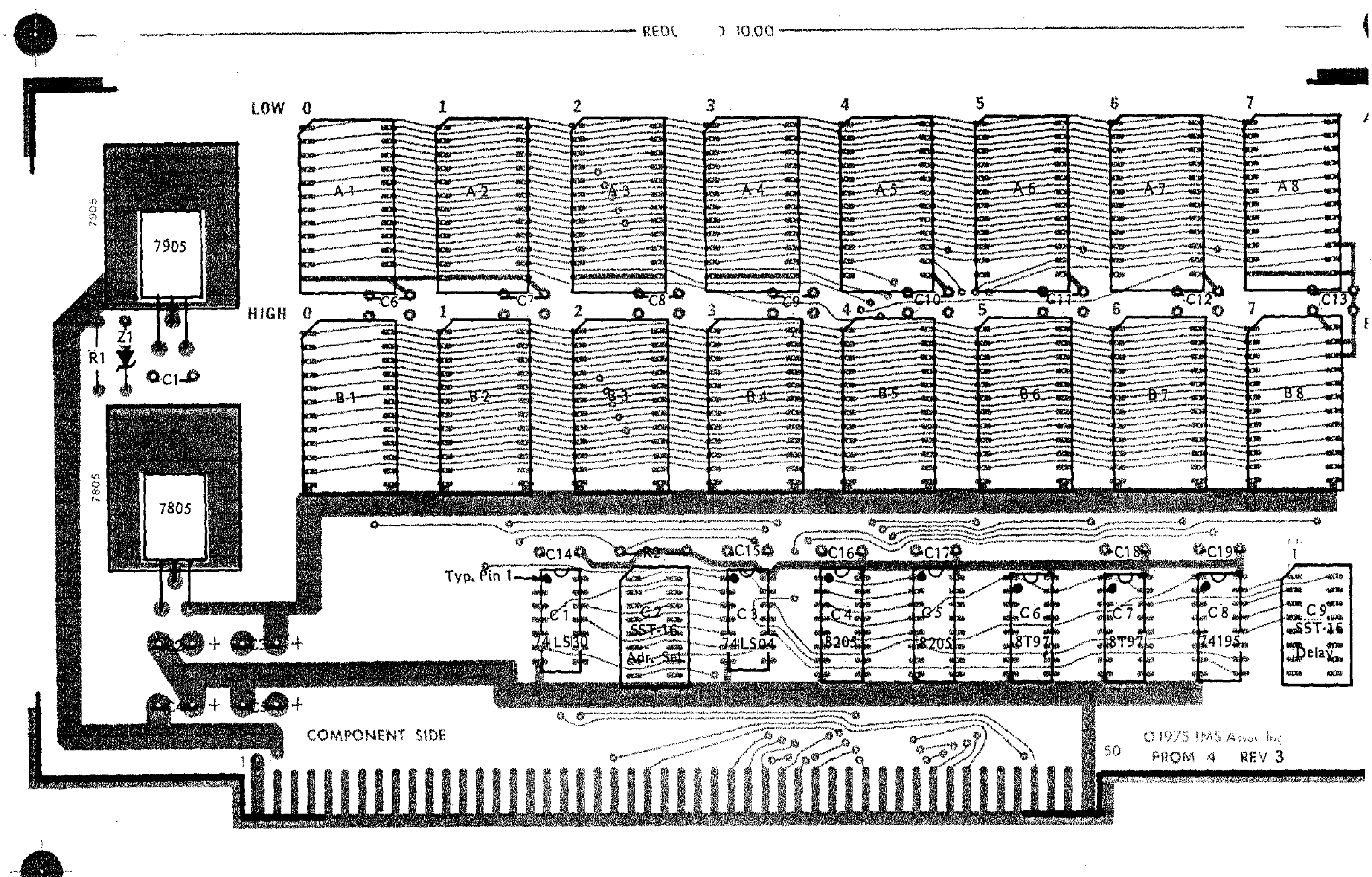
REGULATOR AND HEAT SINK INSTALLATION

CAUTION NOTE: The 7805 and 7905 regulators are physically similar. The identifying number is located immediately below the center hole. Be certain you are using the correct device in each location!

( ) 14. Before installing the heat sink and regulator, bend the 7805 regulator leads at 90 degree angles to facilitate mounting on the heat sink.

( ) 15. Insert the #6 screw through the 7805 regulator and heat sink on the component side of the board and attach through the lockwasher and nut on the circuit side of the board. Tighten the screw carefully to insure proper alignment of the heat sink to prevent shorting to adjacent traces. Solder in the 7805 leads.

( ) 16. In a similar manner install the 7905 regulator and its heat sink, following instructions 14 and 15 above.



IMS Associates Inc.

ASSEMBLY DIAGRAM

PROM 4 REV 3 2/76

2/27/76

USER GUIDE

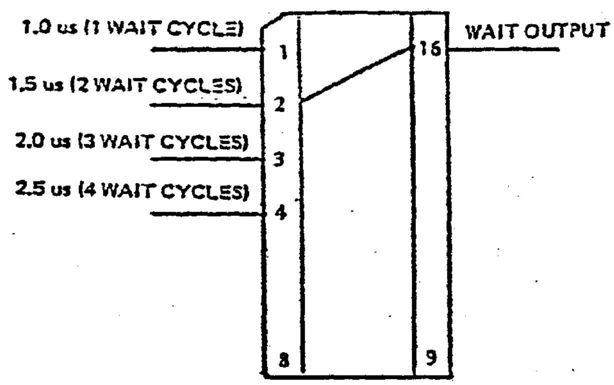
The PROM-4 board uses Intel 8702 or 1702 ROM chips which are structured 256 x 8 bits so that the minimum increment possible in memory space is 256 bytes or 1 8702-1702 chip. The board is designed to contain up to 16 8702-1702 devices, which is the full 4K of PROM. Each of the 16 PROM sockets has its own unique address, and each PROM operates independently of any other PROM. Thus, the user may structure his/her memory space in any way desired merely by placing his/her PROM(s) in the desired location(s).

The PROM-4 board is structured so that the memory address corresponds to a physical location on the board. The PROM sockets are arranged in a 2 x 8 rectangular array, and a particular PROM socket is addressed by address bits A8, A9, A10 and A11. A particular byte in the selected PROM is addressed by address bits A0 through A7. The sockets are labelled LOW 0 through 7 and HIGH 0 through 7. Table 1 should clarify the relationship between address and selected socket.

TABLE 1

ADDRESS				SOCKET
A11	A10	A9	A8	ADDRESSING
0	0	0	0	L0
0	0	0	1	L1
0	0	1	0	L2
0	0	1	1	L3
0	1	0	0	L4
0	1	0	1	L5
0	1	1	0	L6
0	1	1	1	L7
1	0	0	0	H0
1	0	0	1	H1
1	0	1	0	H2
1	0	1	1	H3
1	1	0	0	H4
1	1	0	1	H5
1	1	1	0	H6
1	1	1	1	H7

DELAY SELECTION SOCKET



C9

The delay jumper socket (C9) of the PROM-4 board allows the selection of one of four possible memory read cycle delays. The available delay times are 1, 2, 3, or 4 machine cycles, which translates to 1000, 1500, 2000 and 2500 nanoseconds. This read cycle delay is necessary to insure the data from PROM is correct before transmission to the data bus. Most 1702-8702 chips available are either 1000 or 1500 nanosecond access time chips. The chips provided by IMSAI with the PROM-4 board are 1000 ns access time devices. After determining the access time of the slowest PROM on the board, the user should jumper the delay socket to produce that necessary delay.

Table 2 lists jumper pin numbers for the possible delays. In all cases, jumper the selected pin to pin 16.

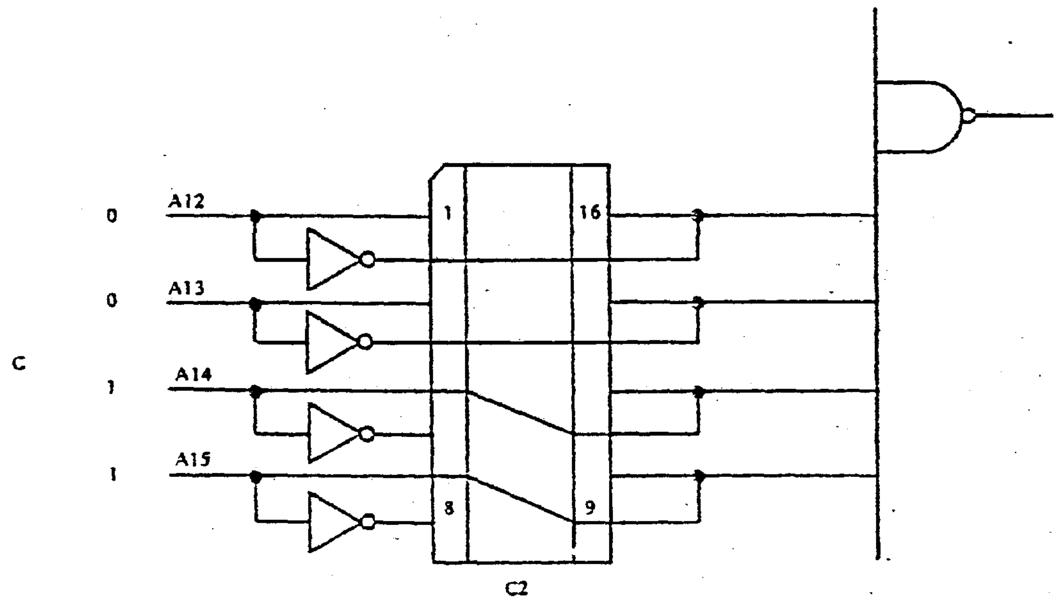
TABLE 2

Delay (ns)	Pin #
1000	1
1500	2
2000	3
2500	4

The example at the top of the page is jumpered for a 1500 ns delay.

Board Addressing

An example jumper for the Address Block beginning with the Address C hex:



The board address select jumper location is C2. It permits any one of the 16 possible 4K blocks of memory space to be jumpered to form the board enable.

The jumper location accepts a standard 16 pin IC socket and the jumpers can be soldered on to a header which can be plugged into the socket and changed easily without any resoldering from the board.

After selecting a board address, the user must properly jumper the socket. Very simply, to enable the board, all address inputs to the NAND gate must be high. Therefore, any address bit not a 1 at the selected address should be inverted before connection to the NAND input.

Address bits 12, 13, 14 and 15 are available on pins 1, 3, 5 and 7 and their respective complements on pins 2, 4, 6 and 8. These signals should be jumpered to the input of the board select circuitry which appears on pins 9 through 16. An 8 position DIP switch similar to that used for write enable may be inserted into this location should very frequent changes of address be desired. For a board whose address is expected to remain the same, jumpers may be inserted directly on the board.

It is suggested that pins 9, 11, 13 and 15 be used to input as desired either a 0 or a 1 from the address bits so that for any address bits desired to be 0, the jumper will extend directly across the header and for any address bits desired to be 1, the jumper will extend diagonally across the header. For instance, if A16 were to be 1, the jumper would extend from pin 7 to pin 9. This makes it easy to visually tell what address the board is jumpered for.

IMSAI

PIO 4

FUNCTIONAL DESCRIPTION

The PIO 4 board provides for up to four input and four output ports of eight bits each parallel input and parallel output. Each input and each output port has its own latch and both input and output latches are provided with handshaking logic for conventional eight bit parallel transfers.

Connection to the input or output ports is made through board edge connectors at the top of the board on .10 inch centers and the fingers will accept the 3M flat cable edge connectors as well as most other .1 inch center-to-center board edge connectors.

The handshake logic on any input or output port will generate an interrupt. The priority level of the interrupt is selectable. The address of the four ports is four sequential addresses, and this block of four addresses may be jumper-selected to be any block of four sequential addresses in the 256 I/O address space. The board may also be addressed with memory-mapped I/O, in which case normal memory read or write instructions are used to read or write data to the Input/Output ports. When using memory-mapped I/O, board addressing is done by selectable jumpers for the lower byte of address and the upper byte of address is hex FF or octal 377.

Provision is made for each of the four output ports to drive eight LED's for a total of 32 on-board LED's.

This feature can be used to provide program-controlled output for dedicated processor applications of the IMSAI 8080 in which case this PIO board would be plugged in where the front panel would normally be mounted and a special photographic mask made to put in front of it with the appropriate labels for the specific purpose the controller is to be used. The front panel can still be used during development by plugging it into an extender card in another slot.

The board is double-sided glass-epoxy-laminate G10-type and all holes are plated through to eliminate the need for any circuit jumpers. The power regulator is provided with a heat sink and has current limiting for protection in case of an overload. The I/O ports utilize the Intel 8212 8-bit latch.

The +5 and ground pins on the input or output port connectors can be used to provide 5 volt power at up to 200 or 300 milliamperes total from the full board. In addition, approximately 100 additional milliamperes of +5 volt power would be available for each 8212 input or output port which is not installed in the PIO 4 board. For example, if four input ports were installed, but only two output ports were installed, the 5 volt power that could be drawn from the connectors would raise from 300 milliamperes to 500 milliamperes.

THEORY OF OPERATION

The board enable is the output of the 74LS30 in position C9. Input to this 8 input NAND gate is the true or complement address bits 2 through 6, according to how they are jumpered. The input and output status bits are logically ORed and the output or its complement is also jumpered to the NAND gate in position C9. These two are used for I/O reference instructions or these two inputs to the NAND gate are taken from the complement of the status input or output instruction and the high address line which comes from the 74LS30 in position C6. This NAND gate in position C6 is active when all the high order of address bits 8 through 15 are true, that is, high. Address 0 and 1 and their complements are fed into a one-of-4 decoder consisting of the 7427 in position and part of the 7402 in position C11 along with one inverter.

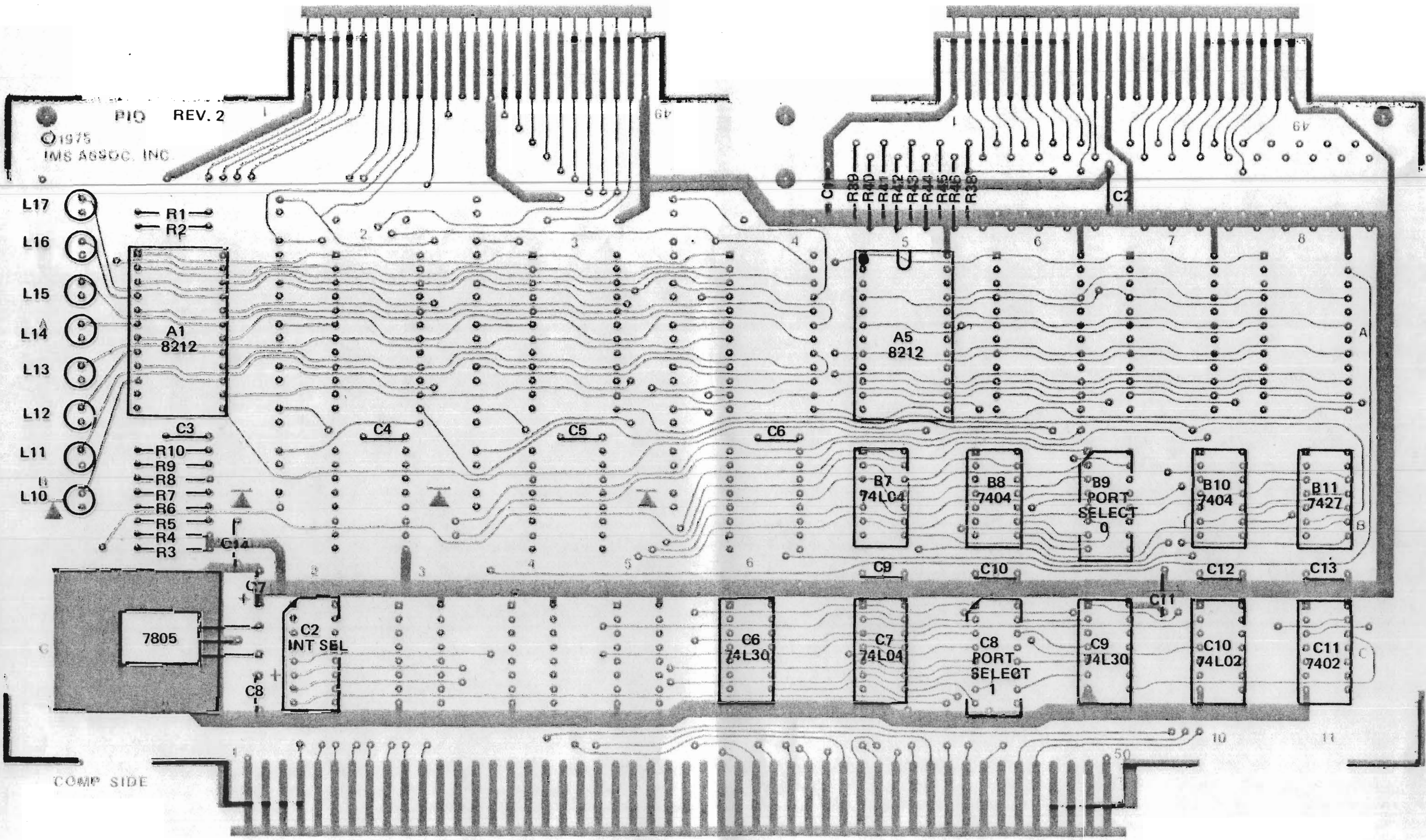
Also as a condition in this one-of-four decoder is the board enable. The outputs of this one-of-four decoder are fed directly to the enable pins on the respective 8212 input or output ports. The DATAIN bus on the IMSAI 8080 system is driven directly from the output of the four input latches. This is a tri-state output and is enabled only when the chip is selected by the one-of-four decoder.

The DATA OUTPUT bus in the IMSAI 8080 goes directly to the four 8212 output ports. The second enable line on each of the input ports is connected to the PROCESSOR DATA BUS-IN signal such that the data is placed on the IMSAI 8080 bus during the time that the processor wishes to read it. The other device select line in output port 8212's is driven by the ORed condition of the PROCESSOR WRITE STROBE or FRONT PANEL WRITE STROBE, these coming from pins 77 and 68 on the IMSAI 8080 back plane respectively. The PROCESSOR DATA BUS-IN signal appears on pin 78 of the IMSAI 8080 back plane.

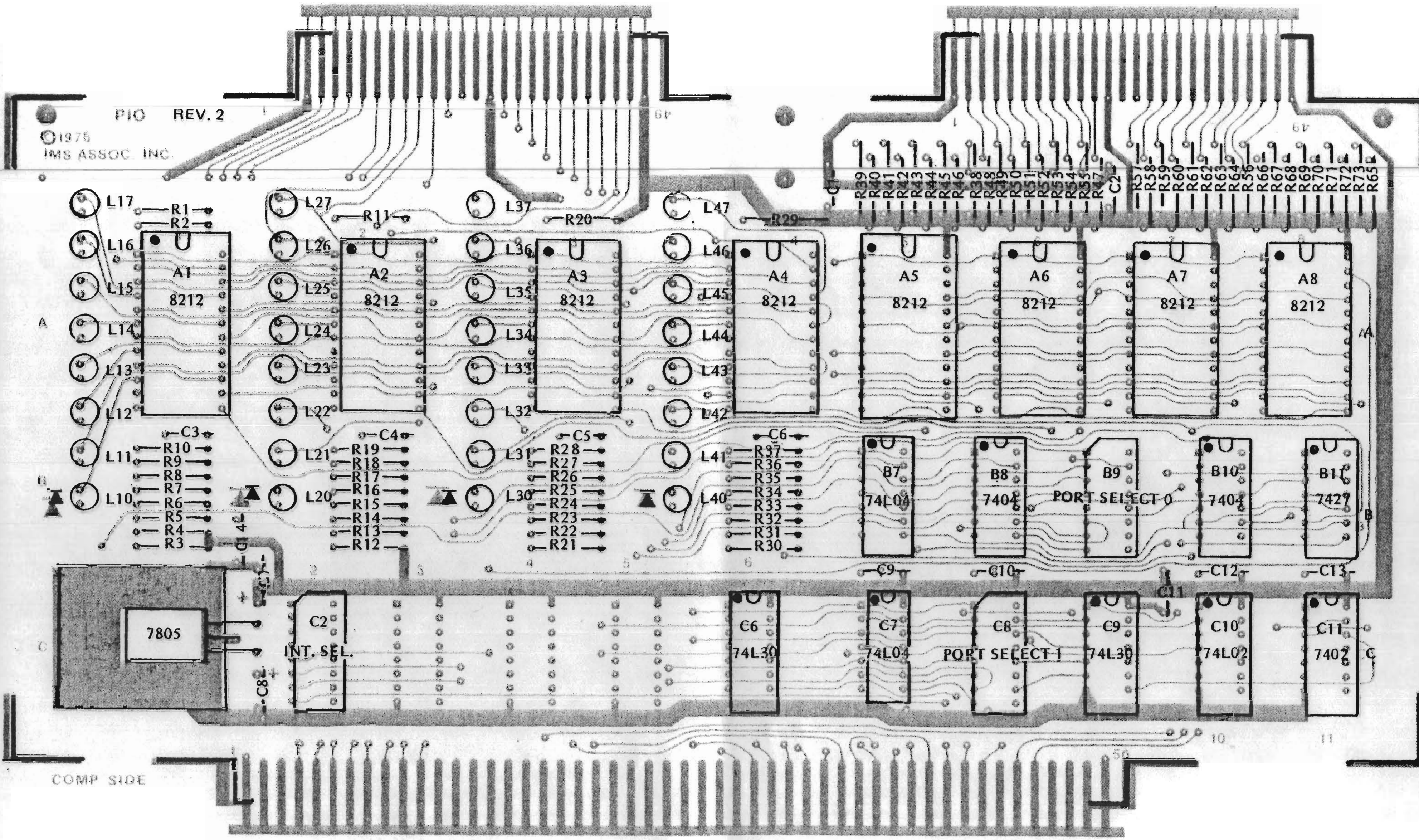
Handling the interrupt levels from the four input and four output ports requires only the interrupt select jumper socket in position 2 so that the appropriate interrupt levels which are already originated by the 8212 chips can be connected as desired to the proper priority interrupt line on the IMSAI 8080 back plane. The remainder of the interrupt function is affected by the PIC-8 board, the Priority Interrupt/Clock board.

The LED's on the output ports are driven through the current-limiting resistor to +5 volts, so that when the output bit is low the LED is on. This orientation was chosen because the 8212's have a greater ability to sink current than they do to source current.

The strobe line into each 8212 input or output port is tied through a 1K resistor to +5 volts so that if the strobe line is not used, it will remain high and noise will not trigger signals on the input or output ports or the interrupt lines. All of the input lines have a 1K resistor to +5 volts, so that when the lines are not connected they exist in a defined state.



IMS ASSOCIATES INC.
ASSEMBLY DIAGRAM
PIO 4–1 REV. 2 5/76

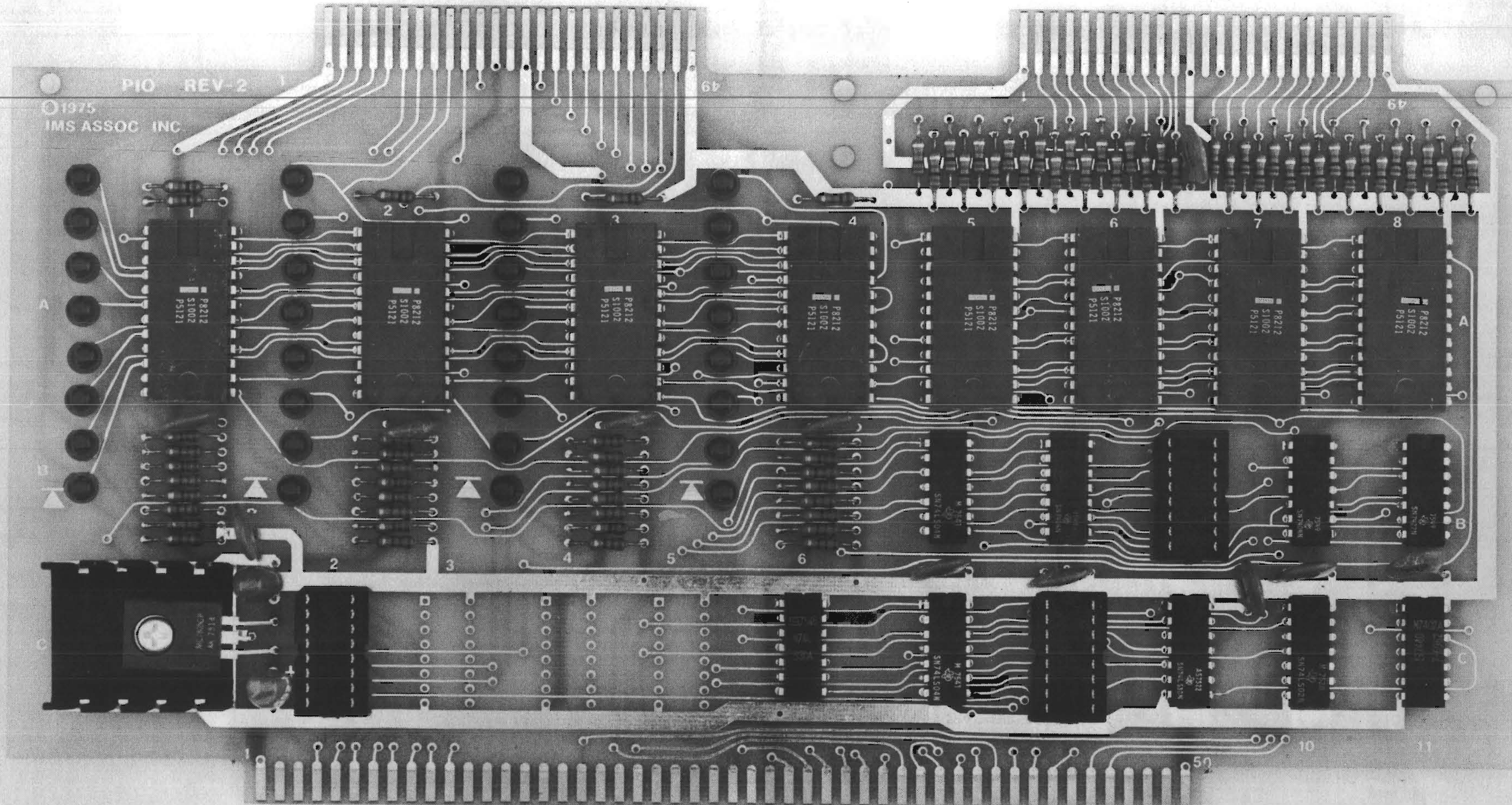


IMS Associates Inc.
ASSEMBLY DIAGRAM
PIO 4–4 REV 2 1/76

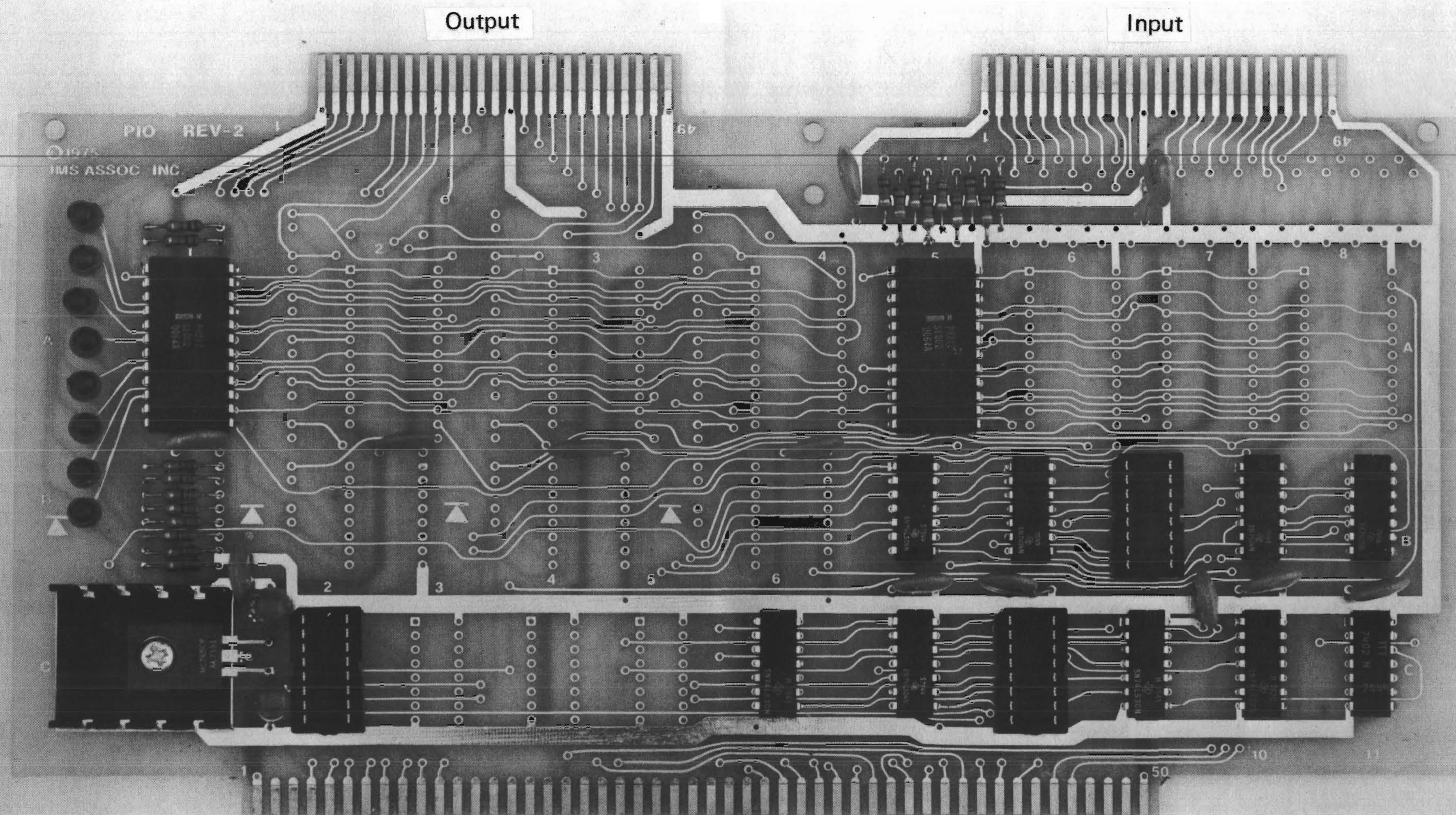
3/4/76

Output

Input



PIO 4 REV 2



PIO 4-1 REV 2

BOARD: PIO 4

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Solder	15-0000001	10'	
Heat Sink	16-0100002	1	Thermalloy Heat Sink/6106B-14
Screw	20-3303001	1	6-32x3/8" Phillips Pan Head Machine
Nut	21-3120001	1	6-32 Hex Nut
Lockwasher	21-3350001	1	#6 Internal Star Lockwasher
Header	23-0400001	3	16 Pin IC Header
Socket	23-0800001	3	16 Pin Solder Tail Socket
Resistor	30-3220362	8 32	(For PIO 4-1) 220 Ohm, ¼ Watt/ (For PIO 4-4) Red, red, brown
Resistor	30-4100362	41	1K Ohm, ¼ Watt/brown, black, red
Capacitor	32-2010010	12	.luF Disk Ceramic
Capacitor	32-2233070	2	33-25 Tantalum
LED	35-3000001	8 32	(For PIO 4-1) Red Light Emitting Diode (For PIO 4-4)
7402	36-0740201	1	Quad 2 Input NOR/SN7402N
74LS02	36-0740202	1	Quad 2 Input NOR (Low Power Schottky)/ SN74LS02N
7404	36-0740401	2	Hex Inverter/SN7404N
74LS04	36-0740402	2	Hex Inverter (LPS)/SN74LS04N
7427	36-0742701	1	Triple 3 Input NOR/SN7427N
74LS30	36-0743002	2	8 Input NAND (LPS)/SN74LS30N
7805	36-0780501	1	5 Volt Positive Voltage Regulator/ MC7805CP
8212	36-0821201	2 8	(For PIO 4-1) 8 Bit I/O Port/D8212 (For PIO 4-4)
PC Board	92-0000013	1	PIO 4, Rev. 2

ASSEMBLY INSTRUCTIONS

( ) 1. Unpack your board and check all parts against the parts lists enclosed in the package.

( ) 2. If gold contacts on the edge connector appear to be corroded, use pencil eraser to remove any oxidation. NOTE: Do not use Scotchbright or any abrasive material as it will remove the gold plating.

RESISTOR INSTALLATION

( ) 3. Insert and solder the thirty-six 1K ohm, ¼ watt resistors (brown, black, red) at locations R38 through R73 in the upper right hand corner of the board as shown on the Assembly Diagram for PIO 4-4 and the 9 1K, ¼ watt resistors at locations R38 through R46 for PIO 4-1.

( ) 4. Insert and solder the other five 1K ohm, ¼ watt resistors at locations R1, R2, R11, R20 and R29 above the output port locations as shown on the Assembly Diagram for PIO 4-4; and the other two 1K ohm, ¼ watt resistors at locations R1 and R2 at output port location as shown on the PIO 4-1 Assembly Diagram.

IC INSTALLATION

NOTE: All Pin 1's are toward the upper left hand edge and away from the 100 pin connector.

( ) 5. Insert and solder the one 7402 at location C11.

( ) 6. Insert and solder the one 74LS02 at location C10.

( ) 7. Insert and solder the two 7404's at locations B8 and B10.

( ) 8. Insert and solder the two 74LS04's at locations B7 and C7.

( ) 9. Insert and solder the one 7426 at location B11.

( ) 10. Insert and solder the two 74LS20's at locations C6 and C9.

( ) 11. Insert and solder the eight 8212's at locations A1 through A8 for PIO 4-4; the two 8212's at locations A1 and A5 for PIO 4-1. NOTE: If fewer than

4 output or 4 input ports are being installed, they would normally be installed from left to right as the address of the output ports; for instance: address 0 for position A3 and address 3 for position A4. The input port chips are arranged in a similar fashion from low to high address left to right, starting with position A5 and going to position A8. The choice of positions to be filled first can be made according to which addresses desired on the input or output ports.

( ) 12. Insert and solder the twelve .luF capacitors at locations C1 through C6 and C9 through C14 as shown on the Assembly Diagram. Be sure the capacitors are pulled in close to the board. In applications where the PIO board will be used in place of a front panel for programmed output in dedicated processor uses, care should be taken to lay the .1 disk ceramic bypass capacitors below the lowest LED position. This is to insure that nothing in the upper half of the board extends higher off the board than the LED's. If LED's are to be mounted, they can be mounted at this time.

( ) 13. Insert and solder the two 33uF tantalum capacitors at locations C7 and C8 as shown on the Assembly Diagram. NOTE: Observe polarity (+ to +) as shown on the board.

( ) 14. Insert and solder the 32 LED's at locations L10 through L17, L20 through L27, L30 through L37, and L40 through L47 as shown on the Assembly Diagram for PIO 4-4; the 7 LED's at locations L10 through L17 for PIO 4-1 as shown on the Assembly Diagram. Take a piece of cardboard, plastic or aluminum and cut it to approximately 3/8" wide for a strip about 3/4 inches long and then cut up the center. Don't cut all the way, so that the center saw cut can be placed on either side of the LED's leads, and the LED's can then be pushed into the holes and against the 1/8 inch spacer and soldered. The tool is then removed, and the LED should have a uniform spacing of 1/8 inch above the board. The 220 ohm or 230 ohm LED resistors should also be installed at this time if the LED's are installed. These are user supplied resistors and are not provided by IMSAI.

( ) 15. Insert and solder the 3 16 pin solder tail sockets at locations B9 and C8, to provide for addressing jumpers, and at location C2 to provide for priority interrupt jumpers.

REGULATOR AND HEAT SINK INSTALLATION

( ) 16. Before installing the regulator and heat sink, bend the 7805 regulator leads at 90 degree angles to facilitate mounting on the heat sink.

( ) 17. Insert the #6 screw through the regulator and heat sink on the component side of the board and attach through the lockwasher and nut on the circuit side. Tighten the screw carefully to insure proper alignment of the heat sink. Solder the 7805 leads.

USER GUIDE

The PIO 4 Board has four input ports and four output ports. Each port has an eight bit latch associated with it. These ports may be addressed in one of two different ways: First, addressed as an input/output port with input or output instructions; second, they may be addressed with memory reference instructions. The type of addressing is selectable by jumpers and the board cannot have both types of addressing at the same time. The four input ports form a block of addresses that are four sequential addresses and the four output ports form a block of four sequential addresses which are the same four addresses as the input port. In other words, the same address used with an input instruction to input on port number 0 is the same address used to output on port number 0.

When the board is being used with memory-mapped I/O, any 8080 instruction which either reads or writes a byte from memory can be used to either read or write respectively a byte from an input or output port on the I/O board. That is, a load accumulator, from the address that this board is jumper-selected to respond to, will load the accumulator with the data from the input port addressed. Each of the four input and each of the four output latches are equipped with data strobe lines. Each port has both an interrupt line and a strobe line which can be used as hand-shake signals for conventional parallel data transfers. In the case of the output ports, a low pulse on the strobe line will set the interrupt line low. The interrupt line changes on the falling edge of the strobe line and the strobe line would normally be kept high.

The interrupt line is made high again upon the trailing edge of the WRITE strobe of the processor which is writing a new eight bits of data into the output port. Thus, the strobe line would be the input hand-shaking line and the interrupt line would be the output hand-shaking line. The interrupt line may also be jumpered to one of the IMSAI 8080 priority interrupt lines on the back plane to effect an interrupt to the processor when it goes low, that is, when the strobe line has been pulsed low to indicate it has been taken by the peripheral device.

If it is not desired to use hand-shaking lines, it is not necessary to jumper them or take any other action. Successive bits may be put out to the output ports with no further action by any other device. In this case, the strobe line would remain high from the on-board pull-up resistor and the interrupt line would remain high for lack of any strobe signal to affect it.

The input ports also have one strobe line and one interrupt line each. Each of the strobe lines for the input ports also has an on-board pull-up resistor. If the strobe line is not connected or if it is driven high, the data in the latch will follow the input lines. The program can read input from the input lines. The program can read input from the input lines and it will read the data that is present at the instant that the input instruction is executed. When the strobe line is made low the data that is present on the input lines at the falling edge of the strobe lines is latched into the input latch and remains there as long as the strobe line is held low. As soon as the strobe line is raised, the data in the latch will again follow the input lines. On the falling edge of the strobe lines the interrupt line will change from high to low.

This can be jumpered to the IMSAI 8080 priority interrupt lines to create an interrupt to the processor, and/or it may be used as an indication that the processor has not yet read the latched data. If, while the strobe line is being held low, the processor reads data from the input port, then the interrupt line will return high at the trailing edge of the read strobe, thus indicating to the peripheral device that the processor has read that data and the latch is available for latching the next data byte into it. Each input and each output port has its own strobe and interrupt line. They may be driven together or separately.

All four of the output port strobe, interrupt and data lines appear on the 50 pin connector on the upper left edge of the board, and all four of the input port strobe, interrupt and data lines appear on the 50 pin connector on the upper right hand edge of the board.

Also appearing on these connectors is ground and +5 volts. See the schematic diagram for pin numbers.

Both the input and the output connectors are designed to accept 3M-type flat cable board edge connectors. The flat cable may be run directly from the peripheral in through the flat cable clamp at the top back of the IMSAI 8080 Cabinet and directly to the edge connector which slips onto the top of the PIO 4 board.

Each of the data input lines on the input ports is tied to +5 volts through a 1K resistor so that unused lines will be read as high data level or true data level.

As an alternative, two EIA type connectors, 25 pins each, may be connected by way of flat cable and the 3M flat cable system to the board, so that the board connections can be taken to the EIA connectors with no hand wiring. This is true of the input port connector and the output port connector.

The connectors are arranged so that if the EIA 25 pin connectors are used, either two output ports or two input ports appear on the pins of the EIA connector along with both ground and +5 volts. If there are two devices, each needing only one output port, they may be connected without hand wiring both to the EIA 25 pin connectors by passing the flat cable through the first connector into a second so that two 25 pin connectors can be connected to the same flat cable; then only one port would be used in the first connector and the other port would be used in the other connector. This would permit separate plugs from separate peripheral devices without requiring any hand wiring of cables. The same, of course, is true for the input ports.

If desired for custom program display front panel-type use or just debugging use, the user can assemble the output ports with 8 LED's and current limiting resistors of either 220 or 330 ohms, so that the user has a visual indication fo the status of the output bits.

If the PIO 4 board is to be used in place of the front panel for custom program output, care should be taken during assembly to make sure that the disk ceramic bypass capacitors are not stood up but rather are laid over on top of the resistors they are next to so that they do not extend higher than the LED's.

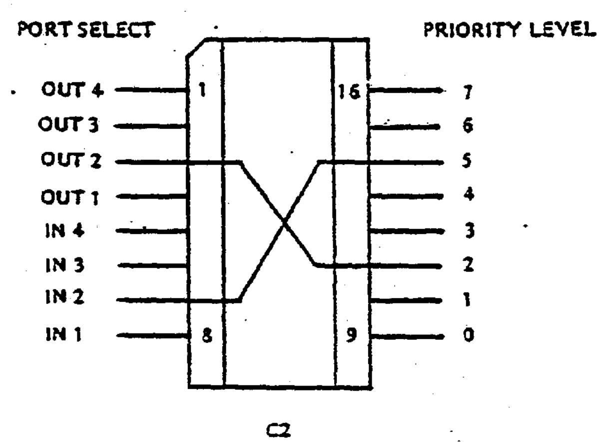
The LED's on the output ports are turned on when the data bit is written as a 0. This was done because it was felt it was a more cost-effective solution for the user to put a complement instruction in his/her firmware than it was for IMSAI to put inverters in the hardware.

The +5 and ground pins on the input and output port connectors can be used to provide 5 volt power at up to 200 or 300 milliamperes total from the full board. In addition, approximately 100 additional milliamperes of +5 volt power would be available for each 8212 input or output port which is not installed in the PIO 4 board. For example, if four input ports were installed, the 5 volt power that could be drawn from the connectors would raise from 300 to 500 milliamperes.

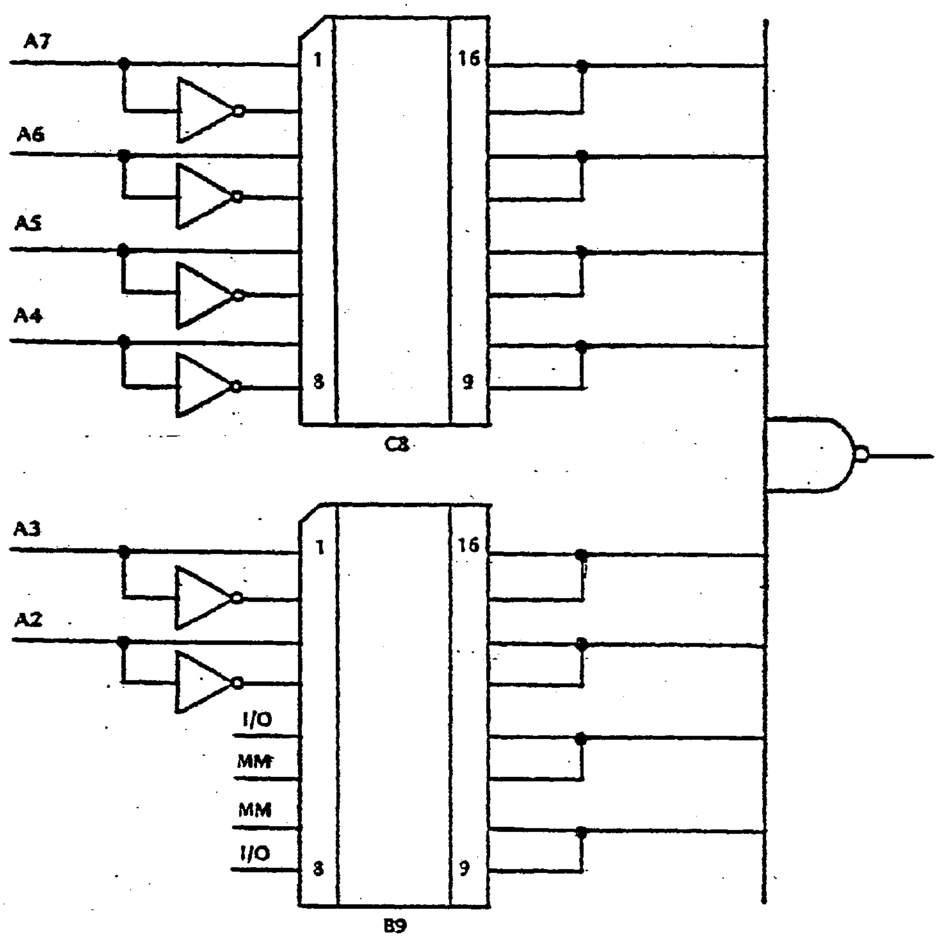
INTERRUPT SELECT JUMPER SOCKET

Position C2 on the PIO 4 board is the interrupt select jumper socket. Appearing at the pins of this socket are all eight of the priority interrupt lines for the IMSAI 8080, the four input interrupt lines and the four output interrupt lines of the PIO 4 board. Thus, any interrupt line desired to be used may be jumpered from the appropriate pin as noted on the diagram below.

If an interrupt is desired to be used, the jumper may be put between the interrupt line from the desired input or output port to the desired priority interrupt on the IMSAI 8080 back plane. The IMSAI PIC-8 board must be used to monitor these interrupt lines and originate the interrupt to the processor according to which line is requesting an interrupt. If more than one line is requesting an interrupt at the same time, the higher priority line rules. An example is shown below for connecting the interrupt line from input port 2 to level 5 priority and the interrupt line from output port 2 to level 2 priority interrupt.



BOARD ADDRESS SELECTION JUMPER SOCKETS



The board address is selected by jumpers or a DIP switch in locations C8 and B9. There are two cases for which this board may be jumpered: 1) to respond to input/output instructions and 2) to respond to memory access instructions. The case of input/output instructions will be treated first.

In selection location B9, pins 8 and 9 must be jumpered together and pins 5 and 12 must be jumpered together. Address bits 0 and 1 determine which of the four input or output ports will be addressed. Port address bits 2 and 3 are also selected on location B9 with jumpers. If, for instance, address bit 2 is desired to be a 0 when the board responds, then pins 4 and 13 would be jumpered together. If address bit A2 was desired to be a 1, then either pins 3 or 13 may be jumperd together, since 13 and 14 are tied to the common address selection input.

It is suggested, however, that when jumpers are being used, pins 3 and 13 be connected together to provide an easy visual indication of whether the address bit is a 1 or a 0 since that will correspond to whether the jumpers are slanted or straight across the jumper socket. Pins 13 and 14 were tied together so that an 8 position DIP switch can be inserted in this location and used to select the address.

Address bits 3, 4, 5, 6 and 7 are jumpered in a similar manner. Address bit 3 is also on location B9; address 4, 5, 6 and 7 are jumpered on position C8. See the diagram on the previous page for pin numbers for each address bit.

If it is desired to use the board in a memory-mapped I/O capacity, then in position B9 the jumpers between pins 8 and 9 and 5 and 12 must be removed and two jumpers inserted between pins 7 and 10 and between 6 and 11. The remaining jumpers for bits 2 through 7 function exactly the same and affect the lower eight bits of the memory address. The upper eight bits of the address will always be all ones, that is hex FF or octal 377.

When used as a memory-mapped I/O board, all instructions that normally affect the memory will operate on the I/O ports. For example, an increment memory instruction would read the data from the addressed input port, increment that data by one and output it on the same address output port.

IMSAI

PIO 6

FUNCTIONAL DESCRIPTION

The PIO6 Board provides the IMSAI 8080 Microcomputer System with two sets of 24 parallel programmable I/O lines and the capability of extending its own standard I/O bus to peripheral devices. Each set of 24 parallel I/O lines is derived from Intel's 8255 programmable peripheral interface integrated circuit. The function of the PIO6 Board is that of a general purpose I/O component to interface peripheral equipment to the IMSAI 8080 Microcomputer System bus.

The functional configuration of the PIO6 Board is controlled by the microprocessor software. The twenty-four 8255 I/O pins are programmable in two groups of 12 each and used in three basic modes of operation:

Mode 0 - Each group of 12 I/O pins may be programmed in sets of four to be input or output.

Mode 1 - Each group of 12 may be programmed to have eight lines of input or output. The four remaining lines in each group are left for "handshaking" and interrupt control signals.

Mode 2 - Bidirectional bus mode which uses eight lines for a bidirectional bus, and five lines, borrowing one from the other group, for "handshaking".

The control lines of the 8255 chips are also brought to PIO6 connector pins (in J2 and J3) for additional control by the peripheral device.

Enough microprocessor control and bus signals are brought through the PIO6 Board connectors J2 and J3 so that the "Isolated I/O" technique directly from the processor can be implemented in the peripheral device. With the addition of J4 to the PIO6 Board, the "Memory Mapped I/O" technique directly from the microprocessor can be implemented in the peripheral device. J4 also provides interrupt lines from the microprocessor. Pads are provided on the PIO6 board so that interrupt lines from the 8255's can be directly jumpered to the processor's vectored interrupts.

Connection to the input or output ports is made through board edge connectors at the top of the board on 0.10 inch centers, and the fingers will accept 3M flat cable edge connectors as well as most other 0.1 inch center to center board edge connectors.

The board uses a 7805 integrated circuit regulator for the +5 volt power and tantalum capacitors before and after the regulator with ample ceramic disk capacitor bypassing throughout the board.

The board size is 4.7 inches by 10 inches with a 100 pin edge connector on the bottom, dual 50's on 0.125 inch centers. On the top of the board are two 50 pin edge connectors, dual 25's on 0.10 inch centers and a 26 pin edge connector, dual 13's also on 0.10 inch centers.

Connector fingers on all edge connectors are gold plated over nickel for reliable contact and long life. The remainder of the board circuitry is tin-lead plated for appearance and reliable solder connections.

The board is double sided glass-epoxy-laminate G10-type, and all holes are plated through to eliminate the need for any circuit jumpers. The power regulator is provided with a heat sink and has current limiting for protection in case of an overload.

The +8, +18, and -18 volt lines provided at the PIO6 Board top edge connectors are bussed directly from the microprocessor motherboard and are unregulated. Before using these voltages, the user should regulate these lines with, for example, a 7805 integrated circuit regulator to obtain +5 volts.

THEORY OF OPERATION

The PIO6 Board is enabled by having its address (as jumpered in the Address Jumper Sockets) appear on the microprocessor address lines during an input or output instruction. When this occurs, the outputs of the Address Jumper Socket will all be high, causing the output of the 74LS30 at location C7 to be low which is the true state of the board enable signal "/BDENA". This signal is buffered by an 8T97 and passed on to connectors J2 and J3 as "/HIADD". Also, AND'ed into the board enable signal is the OR of the microprocessor status bits "SINP" and "SOUT" which are the strobes used to indicate input instruction and output instruction respectively.

The board enable signal is further AND'ed with address lines 2 and 3 to create the 8255 chip selects "/CSO" and "/CS1". This is done in three section of the 74LS32 in location C8. To get either of the chip selects true, address line 3 must be low. Address line 2 low allows /CSO to be true while address line 2 high allows /CS1 to be true.

Once an 8255 is selected, it becomes responsive to the I/O command to transfer data to it or from it. The various functions of which the 8255's are capable are selected by the appropriate combination of address bits 0 and 1 which are bussed to 8255 port address inputs A0 and A1 (pins 9 and 8).

After buffering through 8T97's, the processor's read (PDBIN) and write (/PWR) strobes are wired to the read and write lines of the 8255's. The read signal is inverted before reaching the 8255 to create the proper polarity read signal.

External clear (/EXT CLR) from the processor front panel switch and power on clear (/POC) from the processor are OR'ed together to provide the reset signal to the 8255's and the J2 and J3 connectors.

Four 8216's are utilized to provide two levels of buffering for the data bus. The first pair buffer between the microprocessor and the 8255's. The second pair buffer between the 8255's and the J2 and J3 connectors. The board enable signal (/BDENA) allows the microprocessor data bus to be connected to the 8255's data bus. In addition, if address line 3 (A3) is high, the 8255 bus is connected to the J2 and J3 connectors. The external board enable signal available to the J2 and J3 connectors (/BDEN) enables both pairs of 8216's which allow the microprocessor data bus to be bussed to the J2 and J3 connectors.

All other processor signals brought to the J2, J3 and J4 connectors are buffered by 8T97's. The processor voltages +8, +18 and -18 volts and GROUND are bussed directly from the J1 connector (which plugs into the processor motherboard) to the J2 and J3 connectors.

Traces from the 8255 lines "PC0" and "PC3" are brought from each 8255 to the bottom of the board where they may be jumpered to inverters (to get the right polarity) and then jumpered to either the processor's "/INT" line or one of the vectored interrupts.

TABLE 1

SIGNAL DESCRIPTION J2 & J3 CONNECTORS

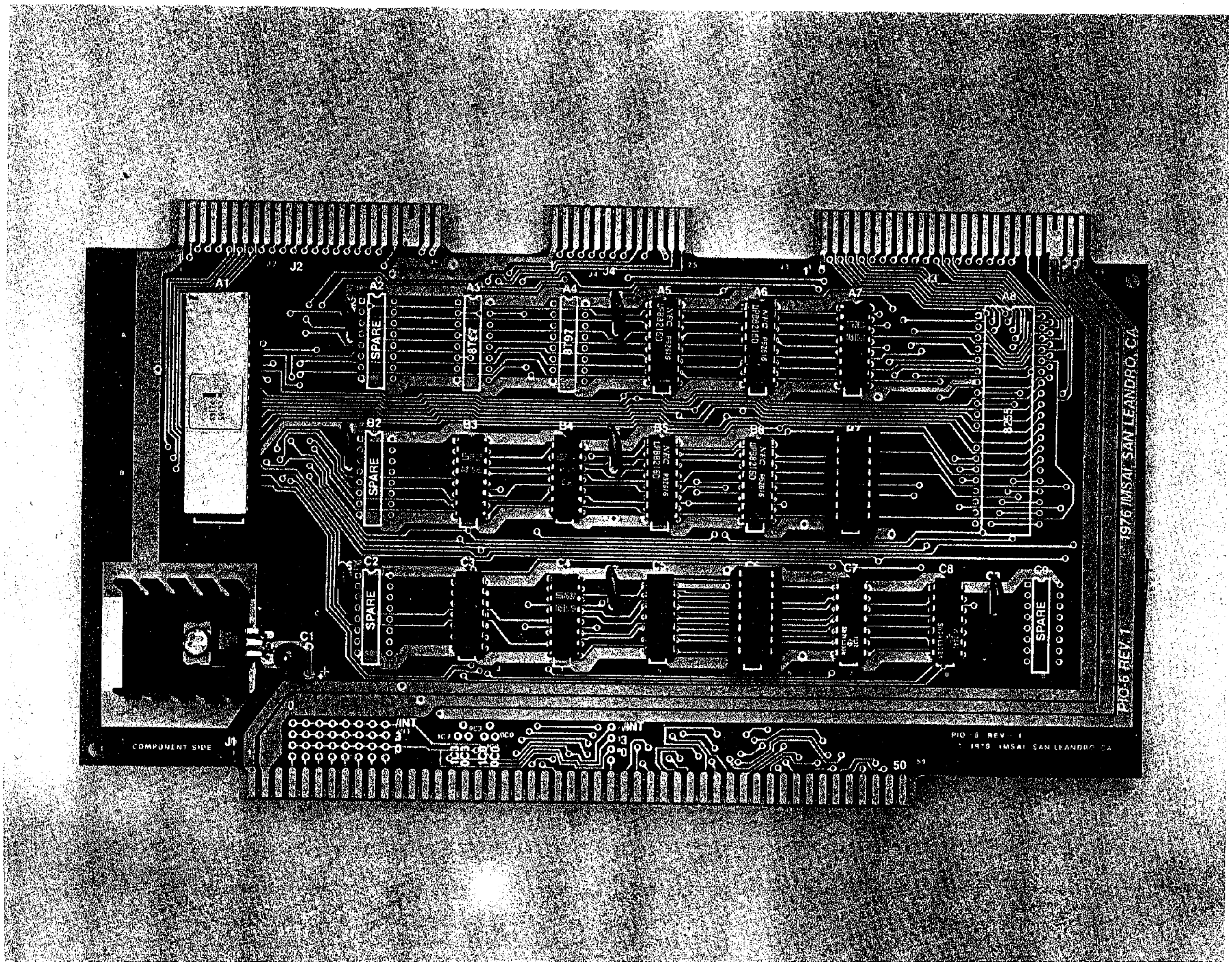
Pin #	Name	Description
1	GROUND	
2	PB0	
3	PB1	
4	PB2	
5	PB3	Port B lines brought directly from
6	PB4	8255 to connector.
7	PB5	
8	PB6	
9	PB7	
10	PC0	
11	PC1	
12	PC2	
13	PC3	Port C lines brought directly from
14	PC4	8255 to connector.
15	PC5	
16	PC6	
17	PC7	
18	PA0	
19	PA1	
20	PA2	
21	PA3	Port A lines brought directly from
22	PA4	8255 to connector.
23	PA5	
24	PA6	
25	PA7	
26	A0	Buffered low order address bits from
27	A1	microprocessor.
28	A2	
29	A3	
30	DB0	
31	DB1	
32	DB2	
33	DB3	Bidirectional data bus from micro-
34	DB4	processor buffered through two 8216's.
35	DB5	
36	DB6	
37	DB7	
38	/HIADD	High Address. Same as internal PIO-6 signal "/BDENA" except buffered through an 8T97. This signal is the decode of the PIO-6 port address. Address lines

TABLE 1

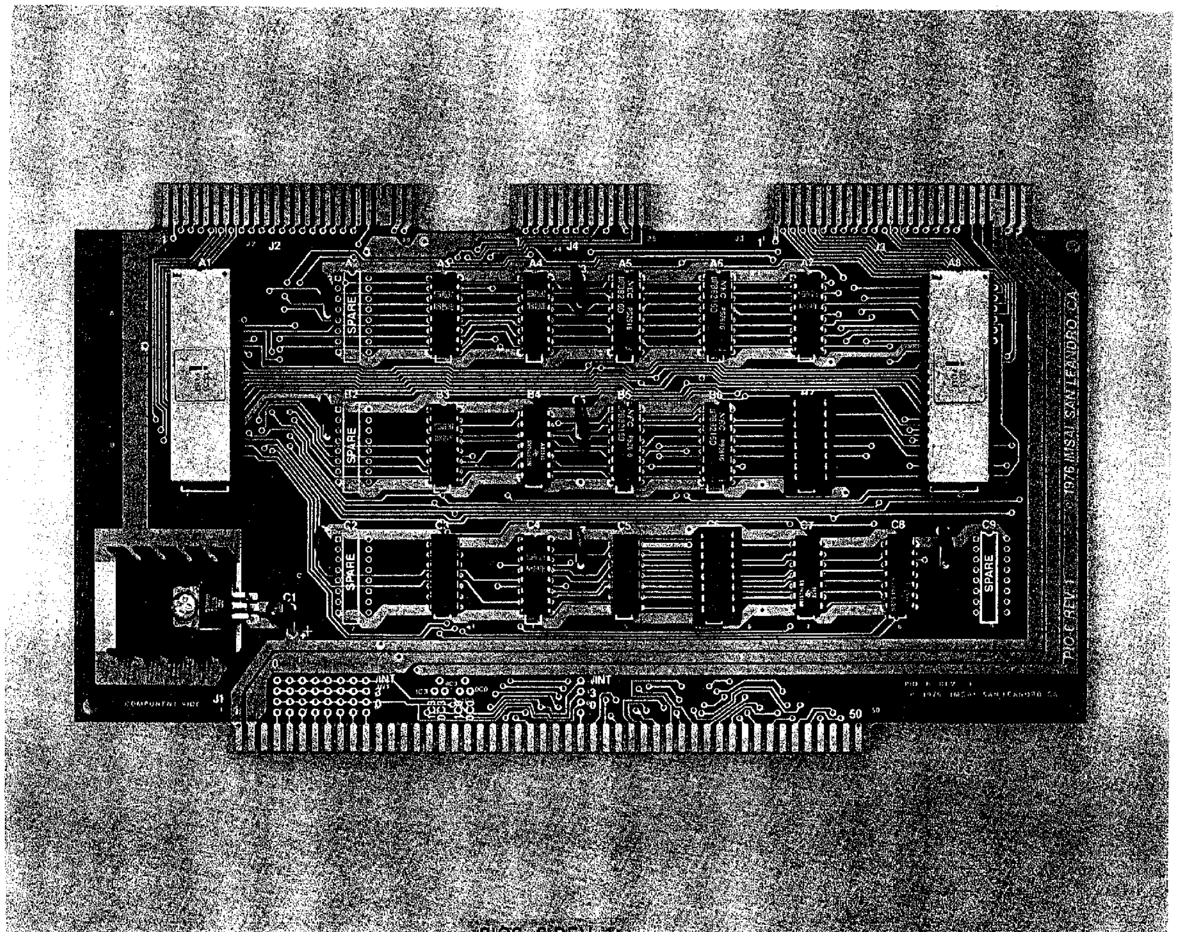
Pin #	Name	Description
38	/HIADD (Cont.)	"A2" through "A7" are decoded. "A2" and "A3" may be jumpered to "DON'T CARE".
39	/RESET	Buffered external clear signal from microprocessor front panel switch.
40	DBIN	Buffered "PDBIN" signal from micro-processor. Processor command/control output signal indicating to external circuits that the data bus is in the input mode.
41	/WR	Buffer /PWR" from microprocessor. Processor command/control output used for memory write or 1/O output con-trol. Data on the data bus is stable while the "/PWR" is active.
42	GROUND	
43	+8V	
44	+8V	+8 volts unregulated.
45	+8V	
46	+18V	+18 volts unregulated.
47	+18V	
48	-18V	
49	-18V	-18 volts unregulated.
50	GROUND	

TABLE 2

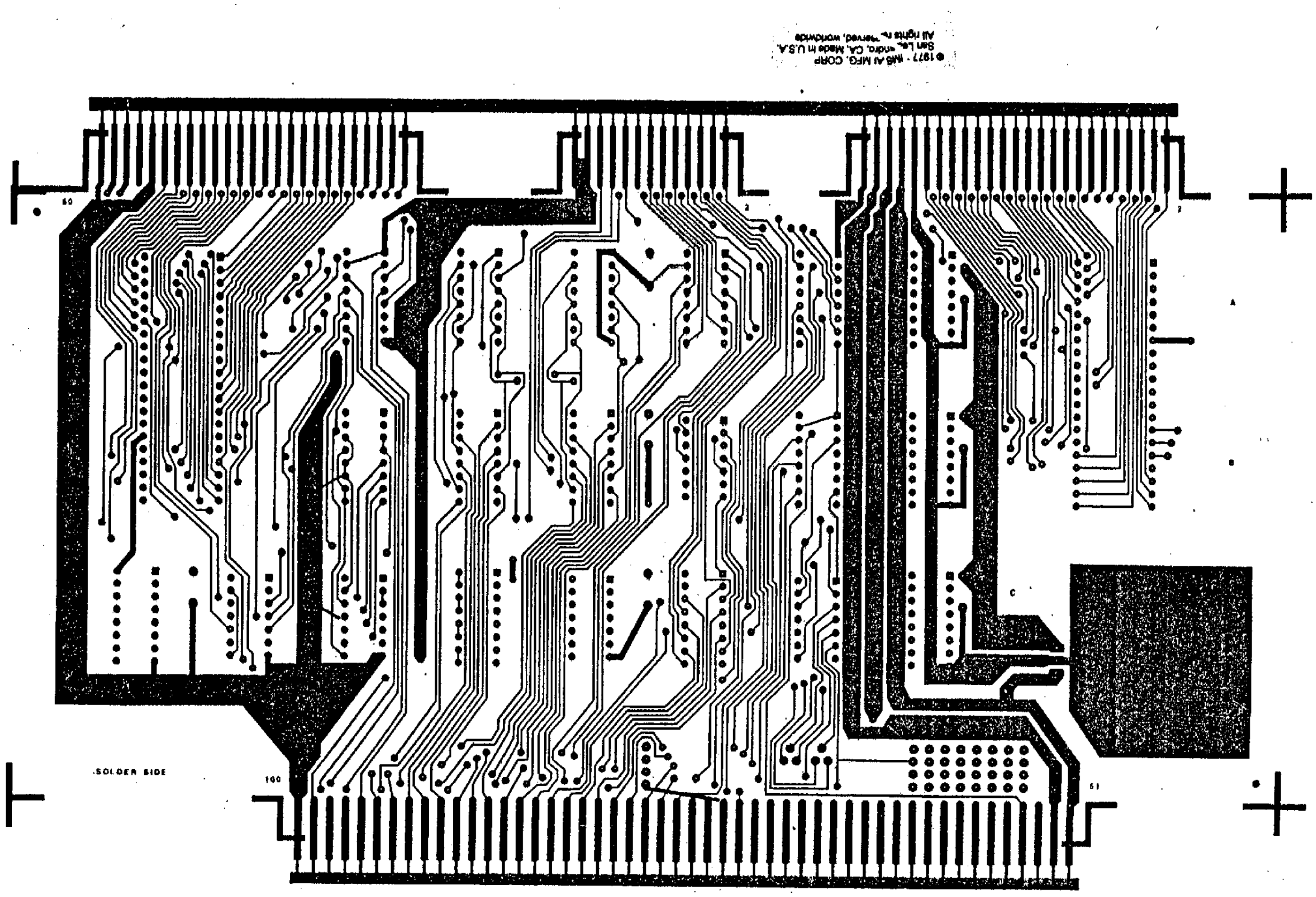
Pin #	Name	Description
20	INP	Buffered status output signal from the microprocessor which indicates that the address bus contains the address of an input device and the input data should be placed on the data bus when "PDBIN" is active.
21	GROUND	
22	READY	Buffered "PRDY" signal to the microprocessor. Processor command/control input that controls the RUN state of the processor. If the line is pulled low, the processor will enter a WAIT state until the line is released.
23	SYNC	Buffered "PSYNC" from the microprocessor. Processor command/control output indicating the beginning of each machine cycle.
24	GROUND	
25	Ø2	Buffered phase 2 clock from the microprocessor.
26	GROUND	


PIO-6 REV. 1
1976 IMSAI SAN LEANDRO, CA
COMPONENT SIDE
SPARE
SPARE
SPARE
SPARE
8T97
8T97
8255

PIO6—3 REV. 1


PIO 6 REV 1
1976 IMSAI SAN LEANDRO, CA
SPARE
SPARE
SPARE
SPARE
COMPONENT SIDE
J1
J2
J4
50

PIO6-6 REV. 1



PIO6 REV. 1 (CUIT SIDE

Item	IMSAI Part #	Quantity	Description/Identifying Marks
Heat Sink	16-0100002	1	Thermalloy, 5 Prong/6106-B-14
Screw	20-3302001	1	6-32 x 5/15" Phillips Pan Head Machine
Nut	21-3120001	1	6-32 x ¼" (OD)
Lockwasher	21-3350001	1	#6 Internal Star
Wire	22-0030001	1'	Wire, Wire Wrap, 30 GA., Kynar, Blue
Header **	23-0400001	2	16 Pin
Socket **	23-0800001	2	16 Pin, Solder Tail Low-Profile IC Socket TIN C831602
Socket	23-0800004	1	40 Pin, Solder Tail Low-Profile IC Socket TIC 834002
Capacitor	32-2010010	7	.luF Disk Ceramic Capacitor
Capacitor	32-2233070	1	33uF, 25V Tantalum Capacitor
8T97	36-0089701	3	Tri-State Buffer/N8T97B
74LS04*	36-0740402	2	Hex Inverter/DM74LS04N
74LS08*	36-0740802	1	Quad 2 Input AND/SN74LS08N
74LS30*	36-0743002	1	8 Input NAND/SN74LS30N
74LS32*	36-0743202	1	Quad 2 Input OR Gate/SN74LS32N
7805	36-0780501	1	5 Volt Positive Voltage Regulator/7805CU
P8216	36-0821601	4	Bi-directional Bus Driver/μPB8216D
8255	36-0825501	1	Programmable Peripheral Interface/C8255
PC Board	92-0000038	1	PIO6-3

* NOTE: Regulator version of all chips is OK.

** Replaceable by 2, 8-position DIP Switches.

Item	IMSAI Part #	Quantity	Description/Identifying Marks
Heat Sink	16-0100002	1	Thermalloy 5 Prong/6106-B-14
Screw	20-3302001	1	6-32 x 5/16" Phillips Pan Head Machine
Nut	21-3120001	1	6-32 x ¼" (OD)
Lockwasher	21-3350001	1	#6 Internal Star
Wire	22-0030001	1'	Wire, Wire Wrap, 30 GA., Kynar, Blue
Header **	23-0400001	2	16 Pin
Socket **	23-0800001	2	16 Pin, Solder Tail Low-Profile IC Socket TIN C831602
Socket	23-0800004	2	40 Pin, Solder Tail Low-Profile IC Socket TIC 834002
Capacitor	32-2010010	7	.1uF Disk Ceramic Capacitor
Capacitor	32-2233070	1	33uF, 25V Tantalum Capacitor
8T97	36-0089701	5	Hex Tri-State Buffer/N8T97B
74LS04*	36-0740402	2	Hex Inverter/DM74LS04N
74LS08*	36-0740802	1	Quad 2 Input AND/SN74LS08N
74LS30*	36-0743002	1	8 Input NAND/SN74LS30N
74LS32*	36-0743202	1	Quad 2 Input OR Gate/SN74LS32N
7805	36-0780501	1	5 Volt, Positive Voltage Regulator/7805CU
P8216	36-0821602	4	Bi-directional Bus Driver/µPB8216D
8255	36-0825501	2	Programmable Peripheral Interface/C8255
PC Board	92-0000038	1	PIO6-6

* NOTE: Regular version of <u>all</u> LS chips is OK.

** Replaceable by 2, 8-Position DIP Switches.

Item	IMSAI Part #	Quantity	Description/Identifying Marks
Socket	23-0800004	1	40 Pin, Solder Tail Low-Profile IC Socket TIC 834002
8T97	36-0089701	2	Tri-State Buffer/N8T97B
8255	36-0825501	1	Programmable Peripheral Interface/C8255

1. Unpack your board and check all parts against the parts list enclosed.

2. If gold contacts on the edge connectors appear to be tarnished, use pencil eraser to remove any oxidation.
NOTE: Do not use Scotchbright or any abrasive material as it may remove the gold plating.

IC INSTALLATION

NOTE: Pin 1's are located toward the upper left hand edge of the printed circuit board and the three edge connectors.

3. Insert and solder the two 74LS04's at locations C3 and C5 as shown on the Assembly Diagram.

4. Insert and solder the one 74LS08 at location B4 as shown on the Assembly Diagram.

5. Insert and solder the one 74LS30 at location C7 as shown on the Assembly Diagram.

6. Insert and solder the one 74LS32 at location C8 as shown on the Assembly Diagram.

7. Insert and solder the five 8T97's at locations A3, A4, A7, B3 and C4 (three for PIO6-3 at locations A7, B3 and C4) as shown on the Assembly Diagram.

8. Insert and solder the four 8216's at locations A5, A6, B5 and B6 as shown on the Assembly Diagram.

9. Insert and solder the two 40 pin sockets at locations A1 and A8 as shown on the Assembly Diagram (one for PIO6-3 at location A1).

10. Insert and solder the two 16 pin sockets at locations B7 and C6 as shown on the Assembly Diagram.

DISCRETE COMPONENT INSTALLATION

11. Insert and solder the one 33uF, 25V tantalum capacitor at location C1 as shown on the Assembly Diagram.

12. Insert and solder the seven .1uF disk ceramic capacitors at locations C2 through C8 as shown on the Assembly Diagram.

HEAT SINK AND REGULATOR INSTALLATION

13. Bend the leads of the 7805 regulator at 90 degree angles approximately ¼" from the bottom edge of the regulator to facilitate insertion.

14. Place the heat sink on the board and insert the regulator leads into the holes. Use the #6 screw and insert from the top side through the regulator and heat sink and tighten with lockwasher and nut on back

side of the board. Solder the 7805 leads. NOTE: Be sure to place in proper position to prevent shorting.

15. Insert the two 8255's at this time into their proper sockets (one for PIO6-3) at locations A1 and A8.

16. Insert the 16 pin headers at this time at locations B7 and C6.

17. On the solder side of the board, cut the trace going to pin 13 of IC-C7 (74LS30). Solder an insulated jumper between pin 11 of IC-C7 (74LS30) and pin 11 of B7 (address jumper select). This may be done most easily on the back of the board (solder side).

NOTE: Step 18 is only necessary if direct interrupt to the $\overline{\text{PINT}}$ bus is used. It is not necessary if either no interrupts from this board are used or vectored interrupts are used.

18. Cut the trace coming from pin 28 of J1 on the component side of the board. Solder an insulated jumper between the top of pin 73 of J1 and the pad with the large diameter hole immediately above pin 28 of J1. This is the pad that was cut free from pin 28. Be careful not to allow solder to flow down the edge connector at pin 73. This jumper is on the solder side of the board.

USER GUIDE

The basic purpose of the PIO6 Board is to provide the microprocessor with two programmable 24 line parallel I/O ports and to extend a fully buffered microprocessor I/O interface to a peripheral device.

The address of the PIO6 Board is jumpered using the "Address Jumper Select" headers in locations C6 and B7. These sockets are also shown in Figure One. The address byte is divided in half. Location C6 contains the four high order address bits. To make an address bit high, the odd numbered pin on the left must be jumpered to the corresponding pin on the right. For example, the dotted lines in Figure One represent an address of:

A7	A6	A5	A4
0	1	0	1

In the low order half of the address, bits 2 and 3 are selected by location B7. These bits can be jumpered as above using pins 5, 6, 7, 8, 9, 10, 11, and 12. Bits 2 and/or 3 can be jumpered as "DON't CARE" bits by jumpering pin 3 to 14 or 4 to 13 respectively.

When addressing the 8255's bit 3 must be low. Bit 2 high, addresses 8255-1 and bit 2 low addresses 8255-0. Address bits 0 and 1 are used for addressing the particular functions within a given 8255. For example, when writing to an 8255 with address bits 0 and 1 both high, this causes the 8255 to interpret the incoming data byte as a control word. A summary of the address scheme is shown in Figure Two.

The 8255's operate completely independent of one another. The functional configuration of the 8255's is under program control by the software residing in the microprocessor. A control word is output to an 8255 setting up a desired functional configuration, while other I/O commands operate on the particular configuration. Refer to the Intel 8080 Microcomputer Systems User's Manual for the details of operation of the 8255's.

Connectors J2 and J3 contain all the signals necessary for operation of and interfacing the 8255's to peripheral devices. Both J2 and J3 are 50 pin edge connectors which accept a 3M type flat cable edge connector. Table One is a list and description of the signals contained in connectors J2 and J3 (see PIO6 Theory of Operation).

By using either J2 or J3 combined with J4, the microprocessor I/O bus can be extended to a peripheral device so the user may implement I/O logic direct from the microprocessor. Both Intel's "Isolated I/O" and "Memory Mapped I/O" techniques may be utilized. The "Isolated I/O" technique utilizes input and output commands between a particular address port and the A register. "Memory Mapped I/O" allows the user to use any memory reference instruction for I/O. It treats I/O ports as memory locations. For details on these techniques and the microprocessor I/O bus, the user is referred to Intel's user's guide again and the microprocessor user's guide.

The J4 connector has 26 pins and also accepts the 3M type flat cable edge connectors.

Signals to and from the 8255's are direct with no buffering. 8255 outputs have the ability to source 1 ma of current at 1.5 volts. This allows Darlington transistors to be directly driven for applications, such as printers and high voltage displays.

The +18, -18 and +8 volt power supplied to J2 and J3 is unregulated. The +5 volts is regulated and is output from the 7805 on the PIO6 Board. The +5 volts supplies the card and does not appear on any of the edge connectors.

The PIO6 Board plugs into one slot of the microprocessor motherboard. Cables may be run directly from the PIO6 Board or from the PIO6 Board to 25 pin EIA connectors at the rear of the microprocessor and from here to the peripheral device.

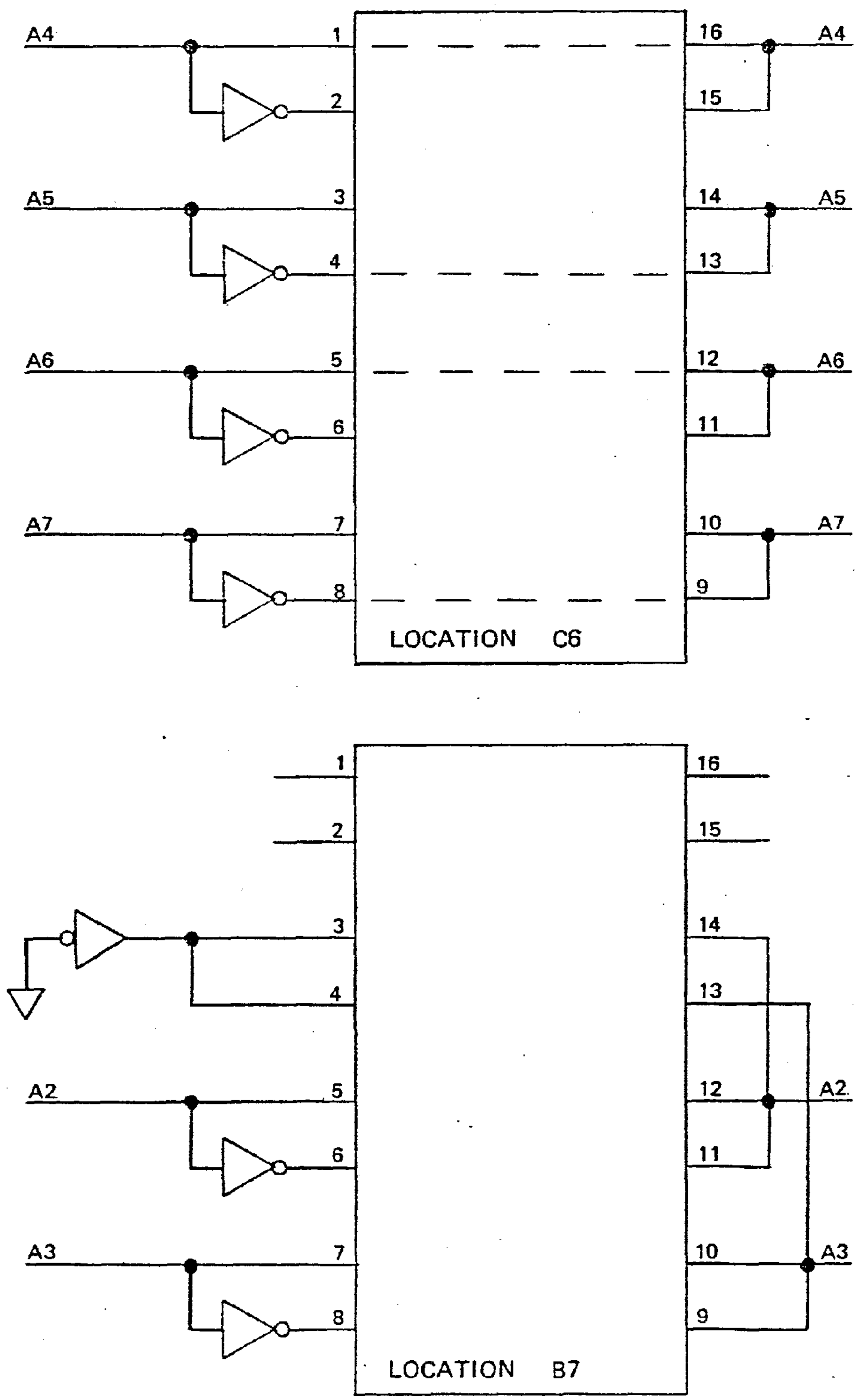
The PIO6 Board is available in two configurations: PIO6-3 and PIO6-6. The PIO6-3 contains one 8255 and the PIO6-6 utilizes two 8255's. J2, J3 and J4 cables are available separately for either configuration.

The 8255 I.C.'s can be programmed to generate a signal to be used as an interrupt request (see <u>Intel 8080 Microcomputer Systems User's Manual</u> for details).

These signals, PC0 and PC3 are brought down to the bottom of the board where they can be jumpered for use. They are labelled "0C0" and "0C3" from 8255-0 and "1C0" and "1C3" from 8255-1. Any two of these can be jumpered to an adjacent inverter input. The inverter outputs labelled "3" and "0" can be jumpered to "/INT" at the pad provided which connects to J1-73 or to the vectored interrupt pads connecting to J1-4 through J1-11. "/INT" is also connected to J4-2 so that an external interrupt request may be generated and brought to either the MPU "/INT" or the vectored interrupts.

A dual 25 card edge connector to the dual D connectors, Cable B, is available for providing interface connections at the back of the chassis.

FIGURE ONE ADDRESS SELECTION



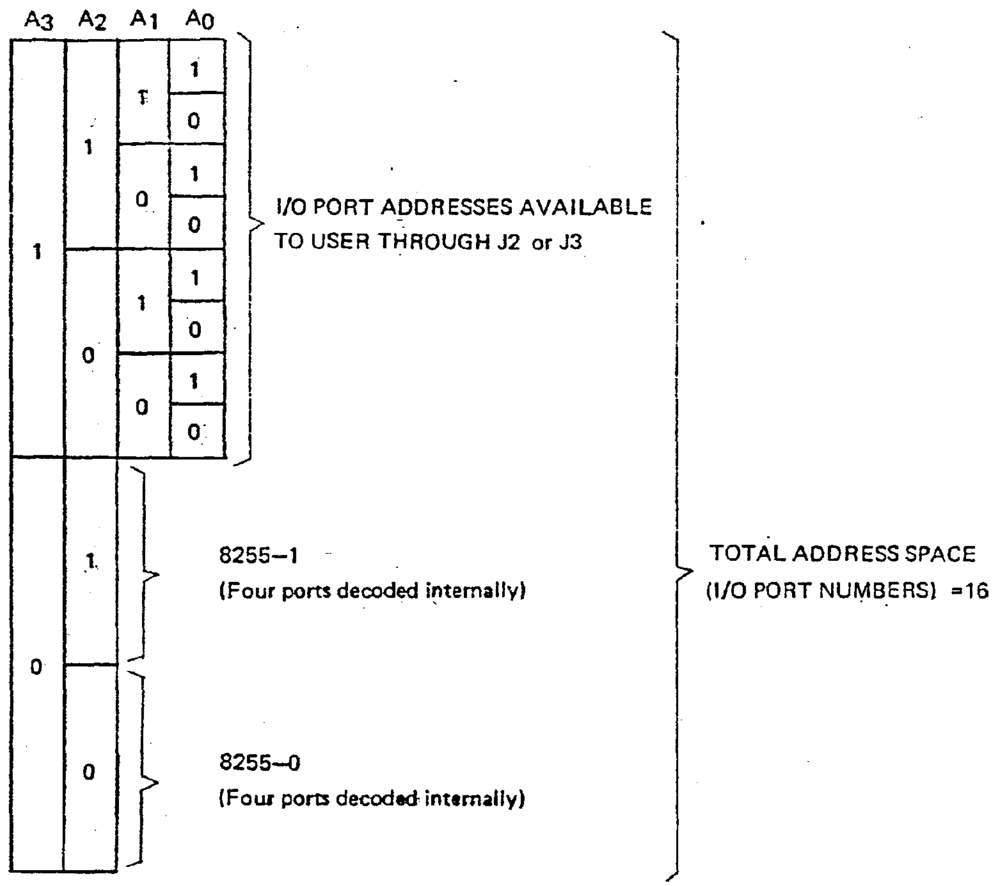


FIGURE TWO ADDRESS SPACE SUMMARY

IMSAI

SIO 2

SIO 2 Board

FUNCTIONAL DESCRIPTION

The SIO Board provides a serial input/output capability for the IMSAI 8080 System. It contains two serial I/O ports, providing two complete RS232 full duplex data lines with all control signals. Data lines for both channels are provided in RS 232, TTL Level and current loop formats. Asynchronous or synchronous lines utilizing full or half duplex can be run with this board at any rate up to 9600 baud in the Asynchronous mode and 56,000 baud in the Synchronous mode.

The SIO Board may be jumper-selected to respond either to input and output instructions from the IMSAI 8080 System or to memory reference instructions for memory-mapped I/O.

Operation of the board requires 16 I/O port or address locations, which are selected by address bits 0 through 3. When the board is used with input and output instructions, address bits 4 through 7 form the remainder of the board address and are jumper selectable. When the board is used as memory-mapped I/O, the lower byte of address is jumper selected exactly the same as an I/O port address and the upper byte of address is hex FE or octal 376.

The SIO Board is structured around a pair of Intel 8251 USART (Universal Synchronous-Asynchronous Receiver-Transmitter) devices.

The 8251 chips provide for extensive program control of the input/output functions including the RS232 Control Line and sync character selection in the Synchronous mode and error condition sense and recovery. The board provides interrupt generation for received characters, empty transmitter buffers, and sync characters detected with provision for jumper selecting the priority of the interrupt. The interrupt works in conjunction with the Priority Interrupt/Clock board (PIC-8).

All functions may also be program controlled so that the full capability of the board is available to the machine without the use of interrupts. All RS232 level drivers and receivers necessary for two complete RS232 lines are included on the board.

Control lines included are DSR, DTR, RTS, CTS, and Carrier Detect. RS232 level drivers and receivers are also provided for receive and transmit clocks for use in Synchronous Mode. Jumper options permit the SIO board to be used either as the receiving (terminal) end of an RS232 line, or as the originating (computer) end.

Jumper options are available so that the two serial I/O ports may be used together so that the control lines are connected together on the two ports and the data lines are received and originated by the 8251 USARTS.

This configuration permits breaking an existing RS232 line and inserting the IMSAI 8080 System between the ends so that the control signals pass straight through and the IMSAI 8080 System intercepts, processes, and retransmits the data. This configuration is extremely useful where format adaptation or other changes must be made to data travelling on RS232 Systems.

Jumper-selectable baud rates are provided on the board for standard asynchronous and synchronous rates up to 9600 baud asynchronous and up to 38,400 baud synchronous. Other rates may be obtained through the use of the SIOC board which contains a jumper-programmable divider which mounts directly onto the SIO Board.

The two output connectors on the top of the board are designed to use the 3M flat cable system to connect directly to 25 pin EIA connectors so that no hand wiring is required to either receive or originate an RS232 line.

TTL and current loop serial input and output are connected to unused pins on the input/output connector. TTL levels are available on the connector for DTR, DATAIN, and DATAOUT, to provide maximum flexibility and utility. A current source is available on the connector for use with current loops. Current loop driving is done through opto-isolators for complete isolation of current loop lines.

Integrated circuit power regulation is provided with high quality tantalum and disc ceramic by-pass capacitors. The board is made on G10-type, 1/16 inch laminate with contact fingers gold-plated over nickel for reliable contact and long life. The remainder of the circuitry is tin-plated for good appearance and reliable solder connections.

Plated through-holes eliminate the need for any circuit jumpers. All jumper options are provided in 16 pin dual in-line package patterns so that jumpers may be installed on headers plugged into IC sockets for convenient and quick changing.

SIO THEORY OF OPERATION

To enable the SIO board, it must be properly addressed. In the I/O port addressed mode, address bits A4 through A7 are jumpered to the 74LS30 (8 input NAND) in C8. The status bits SINP and SOUT are NORed, this intermediate value inverted, and applied (via jumper on D6) to another of the NAND inputs. Remaining NAND inputs in this mode are jumpered (via D6) to a +5 volt level. Thus, when the selected address appears on A4-A7, and the MPU sends a SINP or SOUT pulse, the NAND output goes low and the board is enabled. See schematic.

In the memory-mapped I/O mode, the jumpering in socket C7 still selects an address. The high-order address is interpreted in another 8 input NAND (D8), and hard-wired to respond to the hex value FE. The jumper in socket D6 should be wired to put the inverted output of D8 into an input of C8, and the NORed output of the status bits SINP and SOUT directly connected to the (C8) NAND's input.

The +5 volt tie line jumper in D6 should not be connected for memory-mapped I/O. In this mode, when the corrected high and low order bits are on A4 through A15, and the MPU does not send a SINP or SOUT pulse, the board is enabled. See Diagram.

The SIO board has a bi-directional data bus on the board which connects to the 8251 chips and to the input and output portion of the SIO board control port. The bi-directional bus is connected to the DATA IN and DATA OUT busses on the IMSAI 8080 back plane through 8216 bi-directional bus driver chips. The board enable signal selects these bi-directional bus driving chips and the processor's data bus in signal (DBIN) is used to determine the direction of driving of the bi-directional chips.

8T97's are used to gate the control port data on the bi-directional data bus on the board. They are enabled by the DBIN strobe from the processor and address bit 3.

The 4 output bits of the control port on the SIO board are latched into the 74177 which is clocked by a combination of board enable and address bit 3 and the write strobe either from the processor or from the front panel.

The 8251 chips are selected by address bits 1 and 2, respectively, with address bit 0 determining whether the chip is in control or data mode. The read and write strobes are supplied to complete the control, enabling the chip to read data or write data onto the bi-directional data bus on the board.

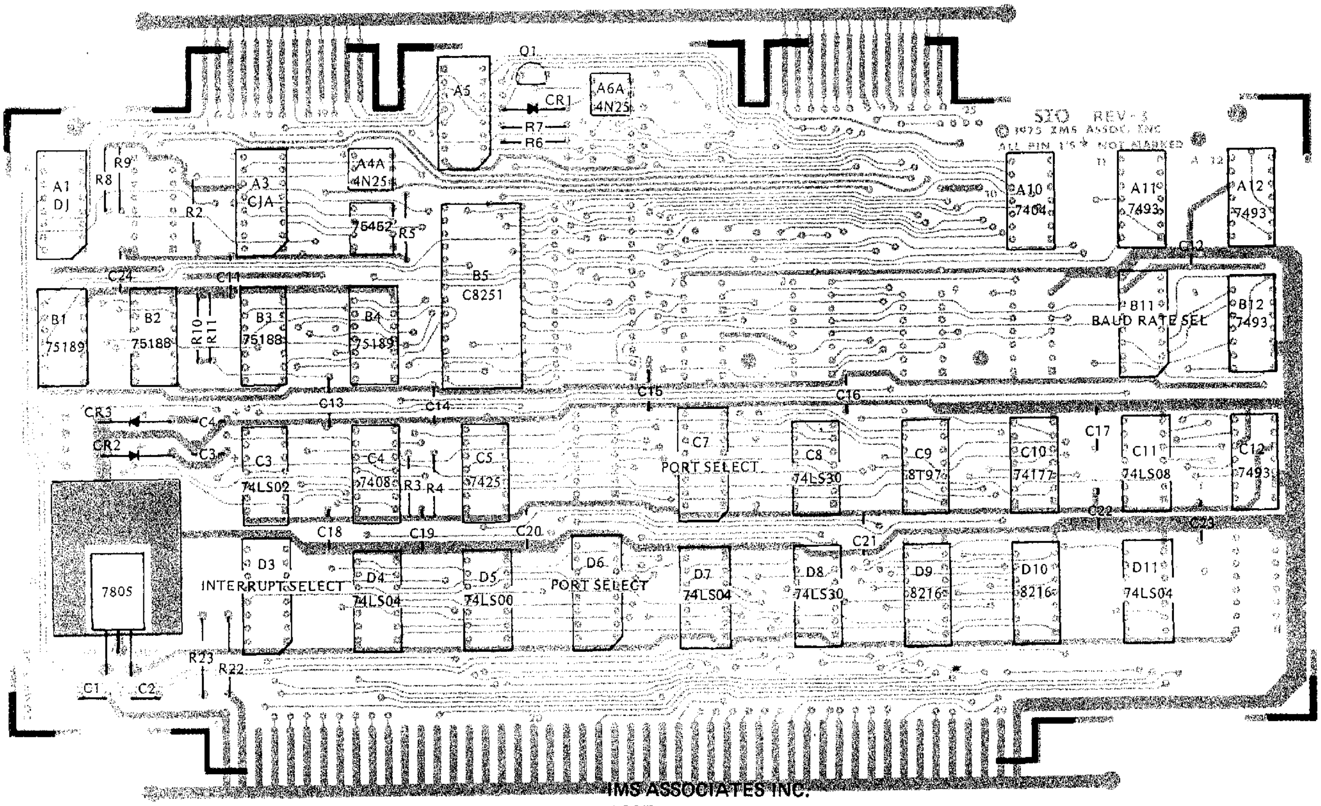
The four control lines desired for interrupt generation are ORed through 7425 and the resultant value supplied to an interrupt select jumper socket (D3). The 7425 OR gate may be disabled by two of the output port bits (IEA or IEB) when interrupts are not desired.

The two megacycle system clock phase II is divided to provide the standard baud rates for jumper selection to channel A and B. It is first divided by 13 through the use of a 7493 with external gating. This produces a rate extremely close to 16 times 9600 baud.

Further division of two are made by 7493's to provide most of the other standard baud rates. 110 baud for a standard teletype is achieved by a divide by 11 from the 2400 baud line which is then divided by 2 to create a symmetrical output and supplied to the jumper socket for 110 baud.

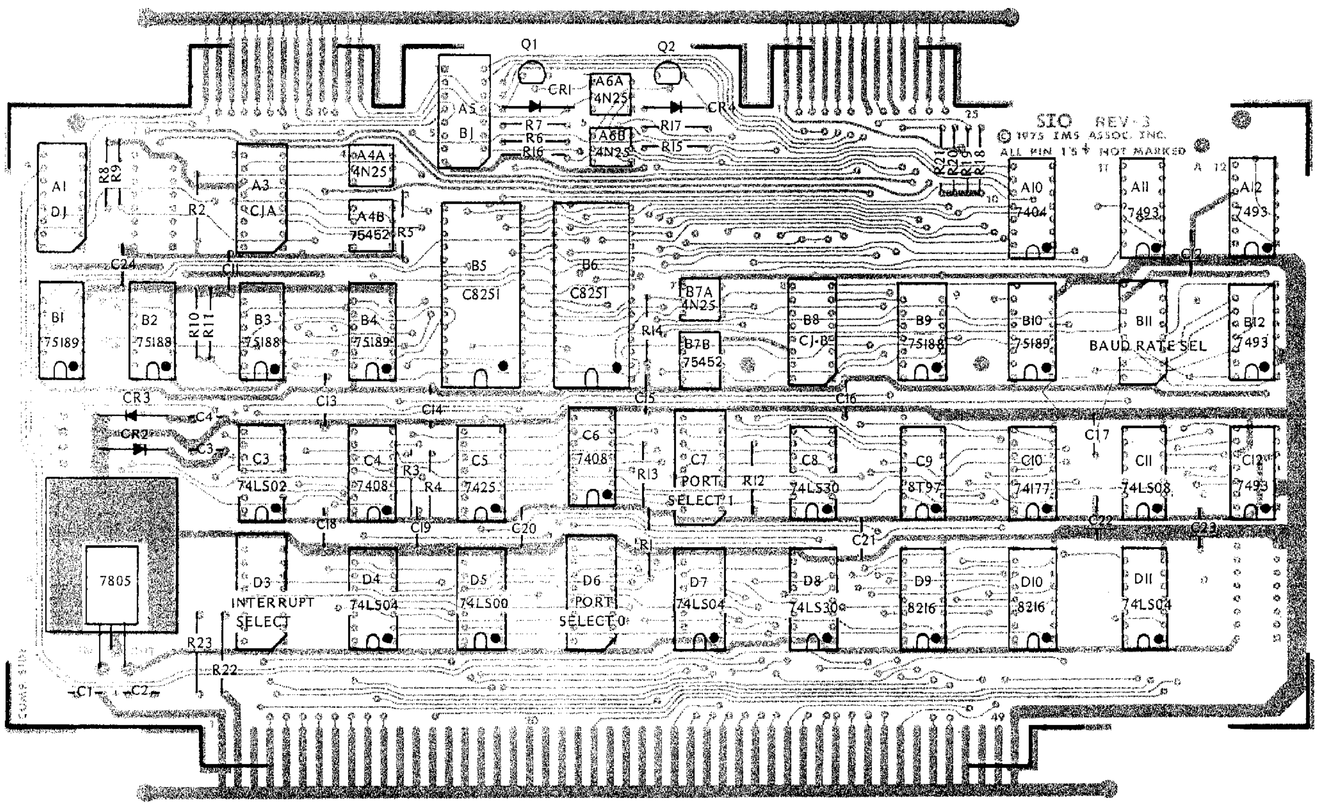
The phase II clock, +5 volts and ground are also supplied to the data rate select socket for use by the SIOC board which connects to the SIO board through the data rate select socket (B11) to provide a jumper-selectable baud rate generator for special rates.

The data and control outputs of the 8251 chips are driven or received through 1488 or 1489 TTL to RS232 level converters as appropriate to the functions. The TTL levels for data and control are driven through open-collector peripheral drivers and a 220 ohm pull-up to +5 volts. The current loop input and output are driven through opto-isolators and are designed to work adequately with either 20 or 60 miliampere current loops.



IMS ASSOCIATES INC.
ASSEMBLY DIAGRAM
SIO 2-1 REV 3 2/76

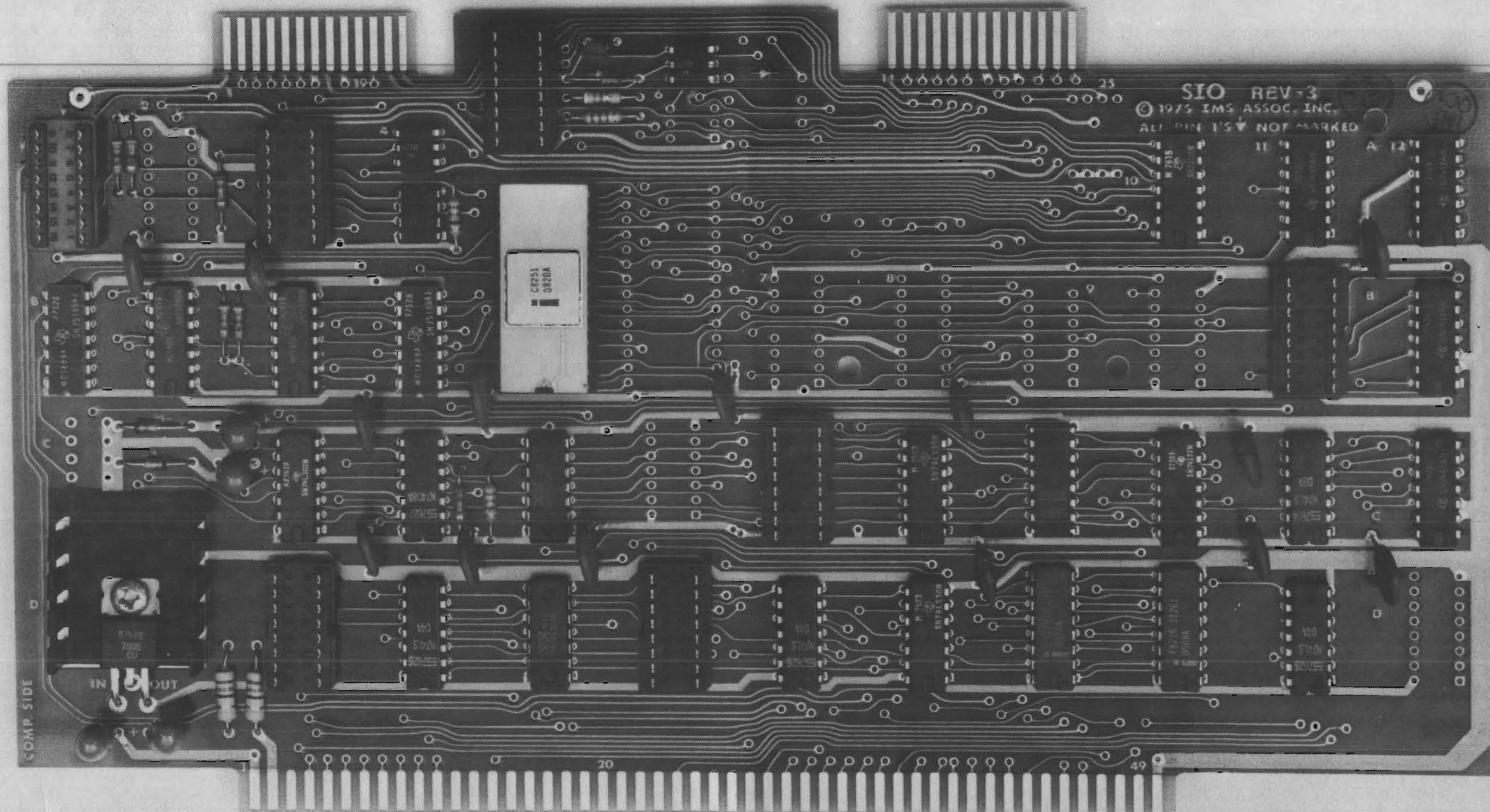
5/18/76



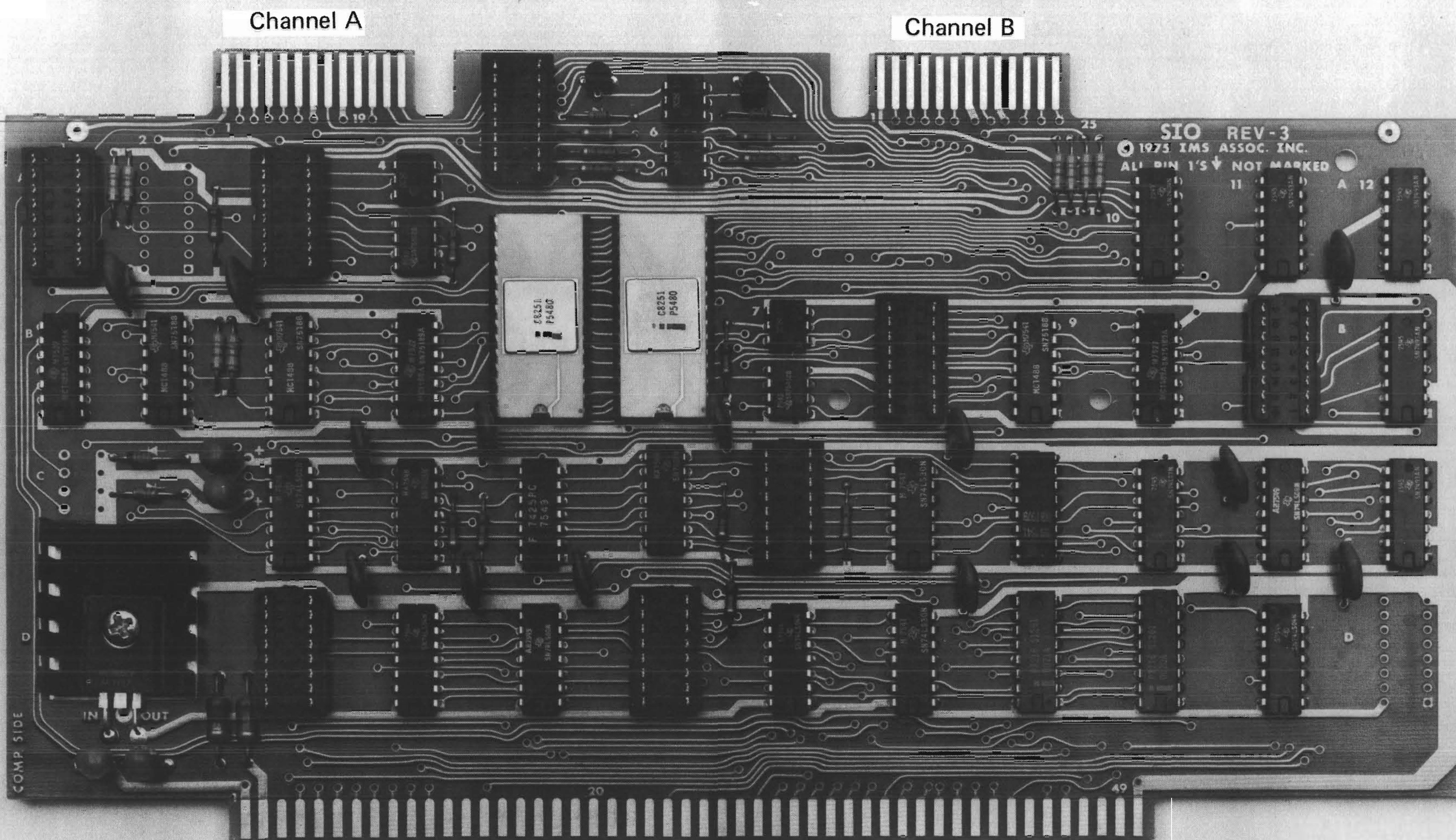
IMS Associates Inc.

ASSEMBLY DIAGRAM

SIO 2-2 REV 3 2/76


SIO REV-3
© 1975 IMS ASSOC. INC.
ALL PIN 1'S ▼ NOT MARKED
COMP. SIDE
IN
OUT

SIO 2–1 REV 3


Channel A
Channel B
SIO REV-3
© 1975 IMS ASSOC. INC.
ALL PIN 1'S NOT MARKED
COMP SIDE

SIO 2-2 REV. 3

BOARD: SIO 2

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Solder	15-0000001	5'	
Heat Sink	16-0100002	1	Thermalloy/6106B-14
Screw	20-3402001	1	6-32x3/8" Phillips Pan Head Machine
Nut	21-3120001	1	6-32 Hex Nut
Lockwasher	21-3350001	1	#6 Internal Star Lockwasher
Header	23-0400001	7	16 Pin IC Header
Socket	23-0800001	7	16 Pin Solder Tail Socket
Socket	23-0800003	1	28 Pin Solder Tail Socket
Resistor	30-2560462	2	56 Ohm, ½ Watt/green. blue, black
Resistor	30-3220362	3	220 Ohm, ¼ Watt/red, red, brown
Resistor	30-3470362	4	470 Ohm, ¼ Watt/yellow, violet, brown
Resistor	30-4100362	4	1K Ohm, ¼ Watt/brown, black, red
Resistor	30-4470362	1	4.7K Ohm, ¼ Watt/yellow, violet, red
Capacitor	32-2010010	14	.luF Disk Ceramic
Capacitor	32-2233070	4	33-25 Tantalum Capacitor (or 22-25)
Diode	35-1000006	1	1N914 Zener Diode
Diode	35-1000009	2	1N4742 Zener Diode
Transistor	35-2000002	1	2N3904 Transistor
Isolator	36-0042501	1	Opto Isolator/4N25
8T97	36-0089701	1	Hex Tri-State Buffer/N8T97B
74LS00	36-074002	1	Quad 2 Input NAND(Low Power Schottky)/ SN74LS00N
74LS02	36-0740202	1	Quad 2 Input NOR (LPS)/SN74LS02N

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
7404	36-0740401	1	Hex Inverter/SN7404N
74LS04	36-0740402	3	Hex Inverter (LPS)/SN74LS04A
7408	36-0740801	1	Quad 2 Input AND/SN7408N
74LS08	36-0740802	1	Quad 2 Input AND (LPS)/SN74LS08N
7425	36-0742501	1	Dual 4 Input NOR with Strobe/SN7425N
74LS30	36-0743002	2	8 Input NAND (LPS)/SN74LS30N
7493	36-0749301	4	4 Bit Binary Counter/SN7493N
7805	36-0780501	1	5V Positive Voltage Regulator/MC7805CP
8216	36-0821601	2	Bi-Directional Bus Driver/D8216/S1261
8251	36-0825101	1	Programmable Comminication Interface/ C8251
74177	36-7417701	1	4 Bit Binary Counter, 35MHz/SN74177N
75188	36-7518801	2	RD 232 Driver/SN74188
74189	36-7418901	2	RS 232 Receiver/SN75189A
75452	36-7545201	1	Dual Peripheral Driver/SN75452BD
PC Board	92-0000018	1	SIO Rev. 3

BOARD: SIO 2

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Solder	15-0000001	5'	
Heat Sink	16-0100002	1	Thermalloy /6106B-14
Screw	20-3402001	1	6-32x3/8" Phillips Pan Head Machine
Nut	21-3120001	1	6-32 Hex Nut
Lockwasher	21-3350001	1	#6 Internal Star Lockwasher
Header	23-0400001	8	16 Pin IC Header
Socket	23-0800001	8	16 Pin Solder Tail Socket
Socket	23-0800003	2	28 Pin Solder Tail Socket
Resistor	30-2560462	2	56 Ohm, ½ Watt/green, blue. black
Resistor	30-3220362	6	220 Ohm, ¼ Watt/red, red, brown
Resistor	30-3470362	8	470 Ohm, ¼ Watt/yellow, violet; red
Resistor	30-4100362	7	1K Ohm, ¼ Watt/brown, black, red
Resistor	30-4470362	2	4.7K Ohm, ¼ Watt/yellow, violet, red
Capacitor	32-2010010	14	.luF Disk Ceramic
Capacitor	32-2233070	4	33-25 Tantalum Capacitor (or 22-25)
Diode	35-1000006	2	1N914 Zener Diode
Diode	35-1000009	2	1N4742 Zener Diode
Transistor	35-2000002	2	2N3904 Transistor
Isolator	36-0042501	4	Opto Isolator/4N25
8T97	36-0089701	1	Hex Tri-State Buffer/N8T97B
74LS00	36-0740002	1	Quad 2 Input NAND (Low Power Schottky)/ SN74LS00N
74LS02	36-0740202	1	Quad 2 Input NOR (LPS)/SN74LS02N

SIO 2-2 Rev. 3
Parts List

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
7404	36-0740401	1	Hex Inverter/SN7404N
74LS04	36-0740402	3	Hex Inverter (LPS)/SN74LS04A
7408	36-0740801	2	Quad 2 Input AND/SN7408N
74LS08	36-0740802	1	Quad 2 Input AND (LPS)/SN74LS08N
7425	36-0742501	1	Dual 4 Input NOR with Strobe/SN7425N
74LS30	36-0743002	2	8 Input NAND (LPS)/SN74LS30N
7493	36-0749301	4	4 Bit Binary Counter/SN7493N
7805	36-0780501	1	5V Positive Voltage Regulator/MC7805CP
8216	36-0821601	2	Bi-Directional Bus Driver/D8216/S1261
8251	36-0825101	2	Programmable Communication Interface/ C8251
74177	36-7417701	1	4 Bit Binary Counter, 35 MHz/SN74177N
75188	36-7518801	3	RD 232 Driver/SN75188
75189	36-7518901	3	RS 232 Receiver/SN75189A
75452	36-7545201	2	Dual Peripheral Driver/SN75452BD
PC Board	92-0000018	1	SIO Rev. 3

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
7408	36-0740801	1	Quad 2 Input AND/SN7408N
75188	36-7518801	1	RD 232 Driver/SN75188
75189	36-7518901	1	RS 232 Receiver/SN75189A
75452	36-7545201	1	Dual Peripheral Driver/SN75452BD
8251	36-0825101	1	Programmable Communication Interface/ C8251
Diode	35-1000006	1	1N914 Zener Diode
Isolator	36-0042501	2	Opto Isolator/4N25
Header	23-0400001	1	16 Pin IC Header
Transistor	35-2000002	1	2N3904 Transistor
Resistor	30-3220362	3	220 Ohm, ¼ Watt/red, red, brown
Resistor	30-3470362	4	470 Ohm, ¼ Watt/yellow, violet, brown
Resistor	30-4100362	3	1K Ohm, ¼ Watt/brown, black, red
Resistor	30-4470362	1	4.7K Ohm, ¼ Watt/yellow, violet, red
Socket	23-0800001	1	16 Pin Solder Tail Socket
Socket	23-0800003	1	28 Pin Solder Tail Socket
Solder	15-0000001	5'	

SIO Assembly Instructions

1) Unpack your board and check all parts against the parts list enclosed in the package.

2) If gold contacts on the edge connector appear to be corroded, use pencil eraser to remove any oxidation. NOTE: Do not use Scotchbright or any abrasive material as it will remove the gold plating.

RESISTOR INSTALLATION

3) Insert and solder the two 56 ohm ¼ watt resistors (green/blue/black) at locations R22 and R23 as shown on the Assembly Diagram.

4) Insert and solder the six 220 ohm ¼ watt resistors (red/red/brown) at locations R4 through R6, R12, R15, and R16 as shown on the Assembly Diagram for SIO 2-2; or the three 220 ohm ¼ watt resistors (red/red/brown) R4 through R6 for SIO 2-1.

5) Insert and solder the eight 470 ohm ¼ watt resistors (yellow/violet/brown) at locations R8 through R11 and R18 through R21 for SIO 2-2; the four 470 ohm ¼ watt resistors (yellow/violet/brown) at locations R8 through R11 for SIO 2-1 as shown on the Assembly Diagrams.

6) Insert and solder the 1K ohm ¼ watt resistors (brown/black/red at locations R1 through R3, R13, and R14 for SIO 2-2; the 4 1K ohm ¼ watt resistors R2, R3, R13 and R14 for SIO 2-1 as shown on the Assembly Diagrams. The remaining two 1K ohm resistors for the SIO 2-2 board are used as jumper selects on A3 and B8. See User Guide section.

7) Insert and solder the two 4.7K ohm ¼ watt resistors (yellow/violet/red) at locations R7 and R17 shown on the SIO 2-2 Assembly Diagram; one 4.7K ohm ¼ watt resistor at location R7 as shown on the SIO 2-1 Assembly Diagram.

IC INSTALLATION

All Pin 1's are toward the lower right hand edge of the PC board and the 100 pin connector.

8) Insert and solder the one 74LS00 at location D5.

9) Insert and solder the one 74LS00 at locaiton D5.

10) Insert and solder the one 74LS02 at location C3.

11) Insert and solder the one 7404 at location A10.

12) Insert and solder the three 74LS04's at locations D4, D7, and D11.

13) Insert and solder the two 7408's at locations C4 and C6 on SIO 2-2; or the one 7408 at location C4 on SIO 2-1.

14) Insert and solder the one 74LS08 at location C11.

15) Insert and solder the one 7425 at location C5.

16) Insert and solder the two 74LS20's at locations C3 and D8.

17) Insert and solder the four 7493's at locations A11, A12, B12, and C12.

18) Insert and solder the one 74177 at location C9.

19) Insert and solder the three 75188's at locations B2, B3, and B9 on SIO 2-2; or the two 75188's at locations B2 and B3 on SIO 2-1.

20) Insert and solder the three 75189's at locations B1, B4, and B10 on SIO 2-2; or the two 75189's at locations B1 and B4.

21) Insert and solder the two 75452's at locations A4B and B7B on SIO 2-2; or the one 75452's at location A4B on SIO 2-1.

22) Insert and solder the two 8216's at locations D9 and D10.

23) Insert and solder the two 28 pin solder tail sockets at locations B5 and B6 on SIO 2-2; or the one 28 pin solder tail socket at location B5.

24) Insert and solder the four 4N25's at locations A4A, A6A, A6B, and B7A for SIO 2-2; or the two 4N25's at locations A4A and A6A for SIO 2-1.

25) Insert the one 8T97 at location C9.

DISCRETE COMPONENT INSTALLATION

26) Insert and solder the fourteen .1 uf disk capacitors at locations C11 through C24 as shown on the Assembly Diagram.

27) Insert and solder the four 33 uf tantalum capacitors at locations C1 through C4 as shown on the Assembly Diagram. NOTE: Observe polarity (+ to +) as shown on the board.

28) Insert and solder the two 1N914 zener diodes at locations CR1 and CR4 as shown on the Assembly Diagram.

29) Insert and solder the two 1N742 zener diodes at locations CR2 and CR3 as shown on the Assembly Diagram.

30) Insert and solder the two 2N3904 transistors at locations Q1 and Q2 as shown on the Assembly Diagram for SIO 2-2; or the one 2N3904 transistor at location Q1 as shown on the Assembly Diagram for SIO 2-1.

31) Insert and solder the eight 16 pin sockets at locations A1, A3, A5, B8, B11, D3, and D6 for SIO 2-2; or the seven 16 pin sockets at locations A1, A3, A5, B11, D3, and D6 for SIO 2-1 as shown on the Assembly Diagram.

REGULATOR AND HEAT SINK INSTALLATION

32) Before installing the heat sink and regulator, bend the 7805 regulator leads at 90 degree angles to facilitate mounting of the heat sink.

33) Insert the #6 screw through the 7805 regulator and heat sink on the component side of the board and attach through the lockwasher and nut on the circuit side of the board. Tighten the screw carefully to insure proper alignment of the heat sink to prevent shorting to adjacent traces Solder in the 7805 leads.

34) Finally, the 8251 chips should be inserted in their sockets with Pin 1 down toward the 100 pin edge connector at the bottom of the board. Addressing and baud rate jumpers should be installed and other option jumpers installed as required. The board is ready for use.

USER GUIDE

The IMSAI SIO Board provides 2 independent channels of serial data input and output. Utilizing the Intel 8251 USART devices, the SIO Board provides 2 channels of RS232, TTL, and current loop data lines with complete control signals.

The SIO Board also includes all logic necessary to control the 8251 devices from the IMSAI 8080 Back Plane.

For reference information on the programming and operation of the 8251 chip, the user should refer to the Intel 8080 Microcomputer Systems User's Manual.

The User's Guide is intended to cover the information beyond that contained in the Intel Data Book necessary to make full use of the SIO board.

Both the memory-mapped and jumper-wired I/O configurations use the lower 4 bits of the address bytes (A1 through A3) to select and control the board's functions. Bit 4 through 7 of the board address (A4 - A7) are jumper-selected as described on another page. If the board is jumper-selected to run as an input and output port type board, then A0 - A7 form a complete address. If the board is jumper-selected to respond to memory-mapped I/O, then A0 - A7 form the lower byte of address and the upper byte of address is hex FF or octal 376.

Address bits 1 and 2 select serial I/O channel A or channel B respectively. That is, when address bit 1 (A1) is high, serial I/O channel A is enabled. When address bit 2 (A2) is on, serial I/O channel B is enabled.

Address bit 0 determines whether the I/O channel selected will respond to the current byte as a control byte or a data byte. If address bit 0 is a 1, the control functions are selected, and if address bit 0 is a 0, the byte is assumed to be data. Thus, to write a control byte into serial I/O channel A, the lower 4 bits of address would normally contain hex 3 or octal 03, while the normal address

SIO BOARD ADDRESSING

Address Bit	Function		
0	C/$\overline{D}$ on 8251's	1 = CONTROL	0 = DATA
1	SELECT CHANNEL A	1 = SELECT	
2	SELECT CHANNEL B	1 = SELECT	
3	SELECT CONTROL I/O	1 = SELECT	
4	CARD ADDRESS Jumperable to any one of 16 addresses		
5			
6			
7			

This byte is I/O port address to run SIO card from INP & OUT instructions.
If SIO card is to be run from memory reference instructions (memory mapped I/O), the above byte is the low order address byte; the high order address byte is FE_{hex} (376_{octal}) (1111 1110 $_{binary}$)

SIO CONTROL I/O BIT DEFINITIONS

Bit	Input Byte	Output Byte
0	always 1	Interrupt Enable chan. A
1	always 1	Carrier Detect chan. A
2	Carrier Detect chan. A	non - functional
3	Clear To Send chan. A	non - functional
4	always 1	Interrupt Enable chan. B
5	always 1	Carrier Detect chan. B
6	Carrier Detect chan. B	non - functional
7	Clear To Send chan. B	non - functional

Carrier detects need option jumper to select originate/receive
Interrupts occur on TxRDY, TxEMTY, RxRDY, and SYNDET

TxRDY and RxRDY interrupts are removed if the respective functions (transmit and receive) are disabled by software command byte. TxEMTY interrupt is removed only by filling transmit buffer with a byte. This may be done while the transmit function is disabled .if desie for channel B control bytes would be hex 5 or octal 05. Address bit 3 (A3) selects the board control I/O port. When address bit 3 (A3) is high, the control port will be enabled. Thus, when use is being made of the control port, the lower 4 bits of address would normally be hex 8 or octal 10.

The control I/O byte selected by address bit 3 is divided into the upper 4 bits and the lower 4 bits. The lower 4 bits, 0 through 3, serve the channel A serial I/O circuit. The upper four bits, 4 through 7, serve the second I/O channel B functions. Bits 0 and 4, for channel A and B respectively, control the interrupt enable separately for each channel. When this bit is a 1, the interrupts are enabled and the processor will receive and interrupt whenever any one of the following 4 lines are active: the transmitter ready line, the transmitter empty line, the receiver ready line, and the sync detect line.

If bits 0 or 4 (as appropriate to channel A or B) are made 0, then no interrupts will be generated from the affected channel. Bits 1 and 5 serve channel A and B, respectively, to output the carrier detect signal. This is operative only when the jumper in jumper socket BJ has selected the board to act as the originator of the carrier detect line.

Bits 2, 3, and 6, and 7 are not functional in the output mode for the SIO control byte. When an input is read from the SIO control byte, bits 0, 1, 4 and 5 are not functional. These 4 bits will always be read as a 1.

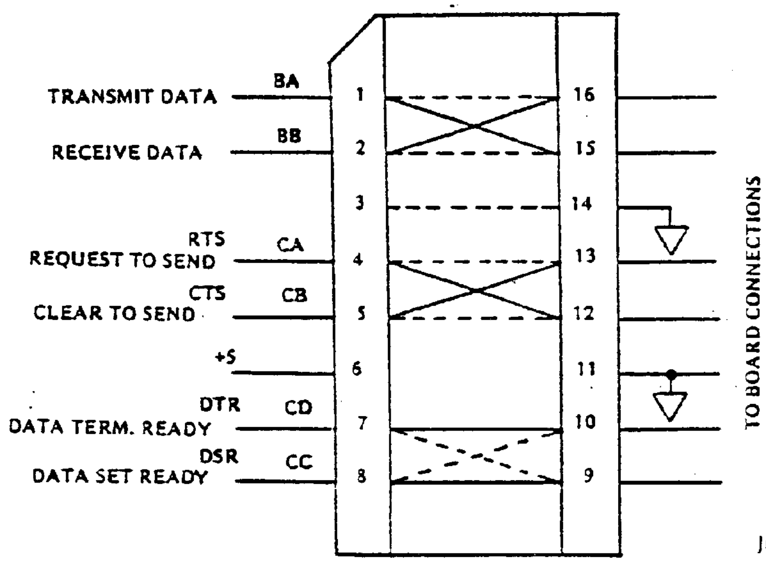
Bits 2 and 6 read the condition of the carrier detect receiver for channels A and B, respectively. The signal is operative only when jumper socket BJ is jumpered to read the condition of the carrier detect line.

Bits 3 and 7 serve channel A and B, respectively, to read the condition of the clear-to-send (CTS) control signal. This is provided because it is not possible to read the condition of CTS through programmed input from the 8251.

SIO BOARD I/O PIN DEFINITIONS

EIA 25 pin connector	26 pin edge connector	RS232 LEVELS	TTL LEVELS	CURRENT LOOP
1	1	AA chassis ground		
2	3	BA Trans. Data		
3	5	BB Rec. Data		
4	7	CA Req. to Send		
5	9	CB Clr. to Send		
6	11	CC Data Set Rdy.		
7	13	AB signal ground		
8	15	CF Carrier Det.		
9	17	+ V		+ V + Current Source
10	19			
11	21			In Loop +
12	23			Out Loop +
13	25			Out Loop −
14	2		$\overline{\text{Data Term. Rdy.}}$	
15	4	DB Trans. Clk.		
16	6		$\overline{\text{Data Set Rdy.}}$	
17	8	DD Rec. Clk.		
18	10		$\overline{\text{Data Out}}$	
19	12		$\overline{\text{Data In}}$	
20	14	CD Data Term. Rdy.		
21	16			Current sink 1
22	18			
23	20			Current sink 2
24	22			
25	24			In Loop −

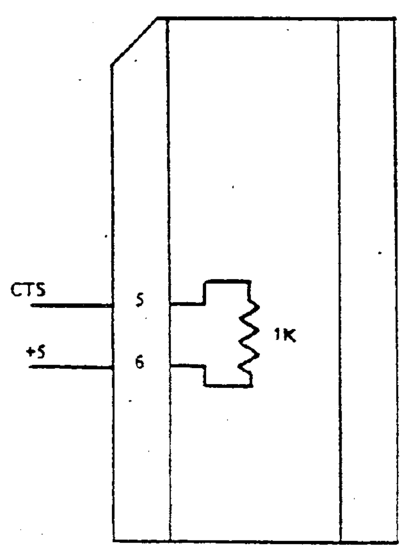
RS 232 OPTION



LOCATIONS A3 CHANNEL A
B8 CHANNEL B

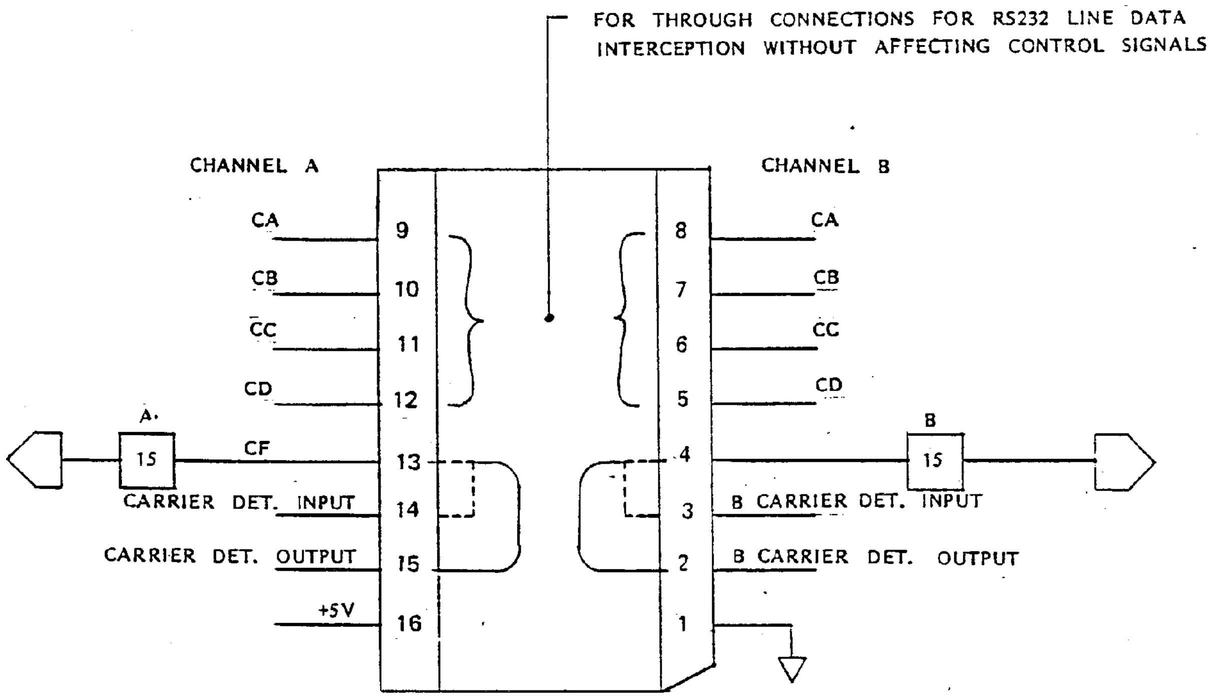
Jumpers shown for connection as terminal or computer end of an RS232 line.
Jumper connection 3 to 14 is always to be made.

TERMINAL ------
COMPUTER ———



CURRENT LOOP OPTION

SIO RS232 INTERCHANNEL CONTROL JUMPERS and CARRIER DETECT

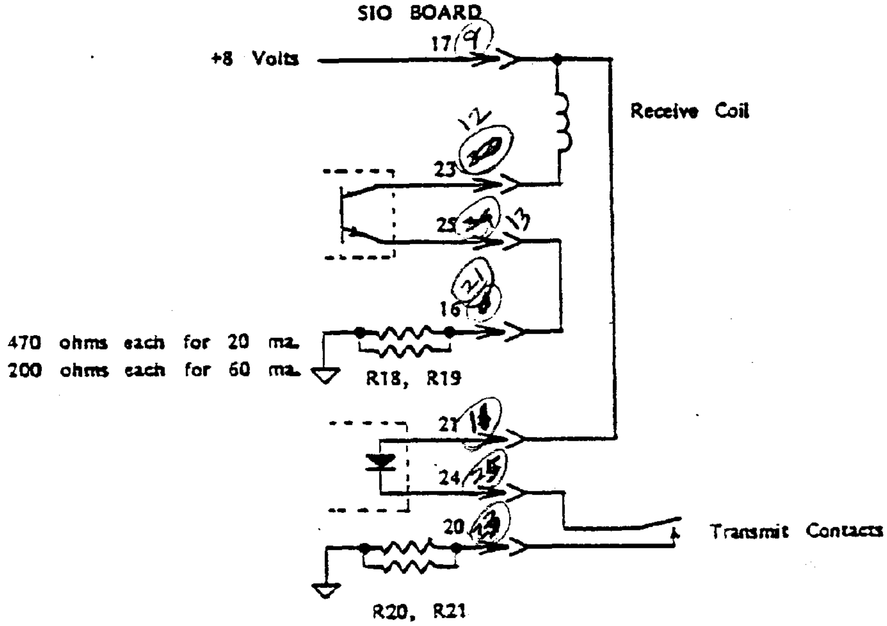


To receive carrier detect - - - - - - - TERM.
To originate carrier detect ———— COMP.

CURRENT LOOP CONNECTIONS

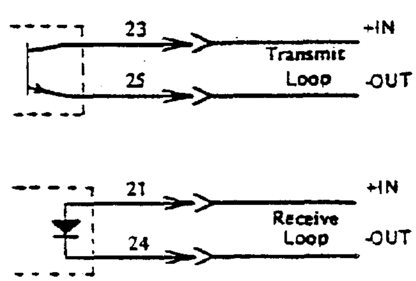
WITH CURRENT SOURCE

SIO BOARD



WITHOUT CURRENT SOURCE

SIO BOARD



The TTL output levels are driven by a 75452 dual peripheral driver, with open collector outputs, and a 220 ohm pull-up to +5 volts. The TTL data inputs drive 1TTL input load and a 1K pull-up to +5 volts.

When the TTL inputs are not being used, they should be left open or held high so as not to affect data input from other sources.

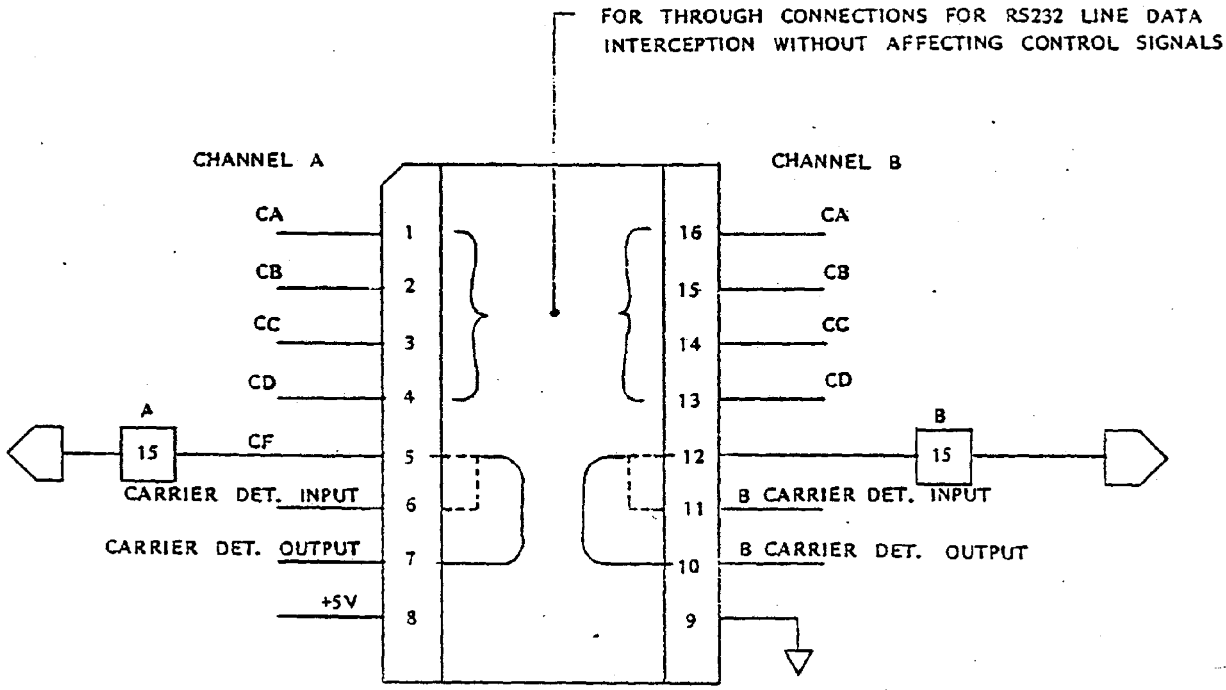
The TTL Data Input line must be left open and not held high when the current loop inputs are used. The current loop input drives opto-isolators and will respond to either 20 or 30 milliamperes. In applications where a significant reverse voltage may be experienced, such as when inductive circuits (i.e., relays) are coupled to the data line, a protective diode should be put across the line such that any reverse voltage spikes will cause the diode to conduct and thus protect the LED in the opto-isolator from too large a reverse voltage.

The current loop output is switched by an isolated transistor through an opto-isolator and is provided with a transient-shunting diode across the output transistor so that it may be used to drive relays without risk of damage to the output circuit. Typical wiring connections are diagramed on another page, both with and without the current source being used.

Setting the baud rate for serial I/O channels A and B is done on the jumper select socket RJ in position B11. The baud rates designated on the detailed sheet for rate select are correct when the 8251 is programmed for a 16X asynchronous clock rate and a 1X synchronous clock rate.

The details of selecting the desired baud rates are located on the schematic.

SIO RS232 INTERCHANNEL CONTROL JUMPERS and CARRIER DETECT



To receive carrier detect - - - - - - - TERM.

To originate carrier detect ———— COMP.

Jumper CJ-A or CJ-B

The jumper selection socket in A3 serves serial I/O channel A and the jumper selection socket in B8 serves serial I/O circuit B. Their functions are the same for their respective channels. The function of this jumper socket is to permit the serial I/O port RS232 levels to be wired so as to either serve as the terminal end of a 232 line or the computer end of a 232 line with no special cable wiring required off the Serial I/O board.

With pins 1, 2, 4, 5, 7 and 8 wired directly across the jumper socket as shown in the diagram for the terminal end, the function of the lines correspond one to one with the names of the RS232 control lines referred to in the 8251 specifications.

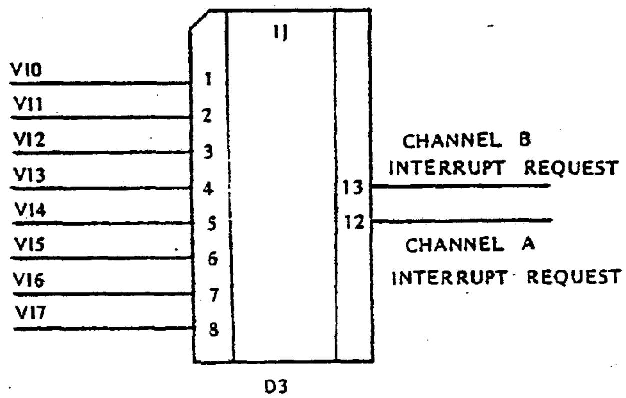
The inputs and outputs are arranged as appropriate for the SIO board to serve as the terminal end of an RS232 line. Should it be desired for the SIO board to serve as the computer end of a standard RS232 line, use jumpers connected as shown in the diagram. The 3 pairs of lines are reversed so that TRANSMIT DATA is now driving what is received data for the terminal and RECEIVE DATA is receiving what is transmit data from the terminal, and similarly, REQUEST TO SEND and CLEAR TO SEND are reversed and DATA SET READY and DATA TERMINAL READY are reversed.

Ground and +5 volts are available on the socket for providing permanent mark or space levels to any of the control lines if CLEAR TO SEND is not driven by an external source. It should be wired to pin 6 to provide a constant enable for the transmitter section of the USART.

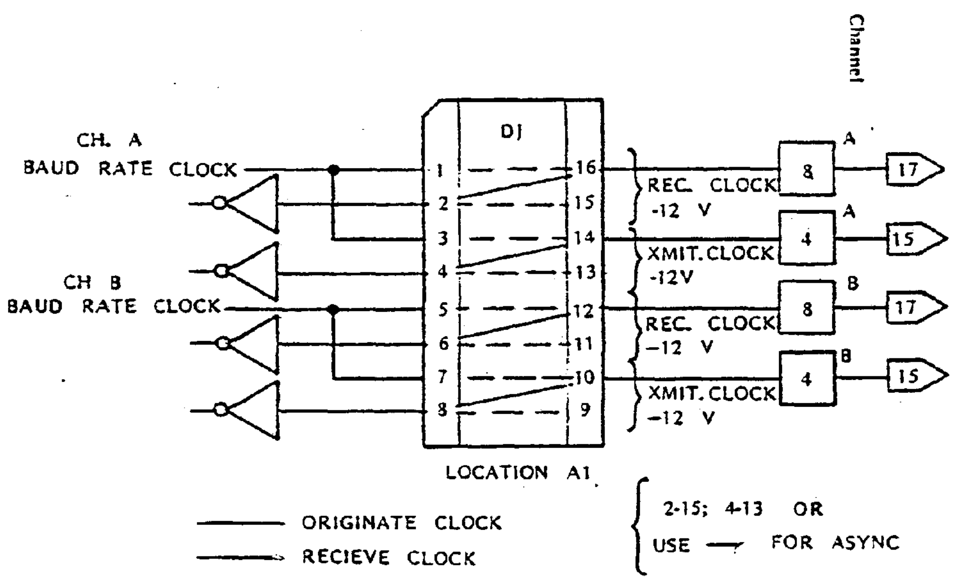
Jumper Socket BJ

Socket BJ serves both to determine whether CARRIER DETECT is being originated or received by the SIO board. It is also used to jumper the control lines between channel A and channel B for applications where the control lines are desired to be passed through and data intercepted and

SIO INTERRUPT SELECT SOCKET



SIO 232 CLOCKS JUMPER OPTIONS



Program 8251 for x16 for asynchronous operation,
x1 for synchronous.

handled. The four primary control lines for both channel A and channel B appear in this jumper socket, and can be jumper-wired straight across as desired.

It should be remembered that only one source should be driving an RS232 line at a time. If the control lines are jumpered straight across so that the modem and data terminal are driving the lines, then appropriate jumpers in jumper socket locations A3 or B8 should be removed so that the SIO board will not be attempting to drive these lines at the same time. If it is desired to detect the DATA TERMINAL READY line, then a jumper needs to be placed as shown between pins 5 and 6 for channel A, or between pins 11 and 12 for channel B.

If it is desired to originate the CARRIER DETECT line, a jumper should be placed instead between pins 5 and 7 for channel A, for 10 and 12 for channel B.

Ground and +5 volts are available in this jumper socket for providing a permanent mark or space level to any of these control lines.

The interrupt line from channel A and channel B both appear on the interrupt select socket in position D3. All 8 of the IMSAI 8080 system priority interrupt lines on the back plane, also appear on the interrupt select socket. A jumper may be placed between the appropriate channel's interrupt line and any one of the priority interrupt system lines to provide an interrupt of the desired priority.

Jumper Location DJ, Located in A1

The jumper select socket in A1 provides facilities for originating and receiving clock signals for receive or transmit for use in the synchronous mode of communication. One-half of the socket controls lines for Channel A and the other half is dedicated to Channel B. Pins 1, 2, 3, 4, and 13, 14, 15 and 16 serve the channel A jumper functions. The remainder of the pins have the identical function for Channel B.

When it is desired to originate the clock signal the pins for that channel should be jumpered straight across, as shown in the diagram, so that the clock signal from the SIO board is driven through converters to RS232 levels onto the DD and DB lines.

The inputs to the data clock receive circuits are tied to -12 volts to provide an inactive output to the OR-gate supplying the receive clock to the USART chip.

When it is desired instead to receive the clock from the RS232 cable, then these jumpers are removed and the RS232 lines DD and DB are jumpered to the input of the clock-receive circuits as shown in the diagram.

When this is done, the data rate select socket for the appropriate channel must be jumpered so that the clock line from this jumper select socket is held at ground or low in order to avoid interference between the onboard clock circuit and the incoming clock from the RS232 line.

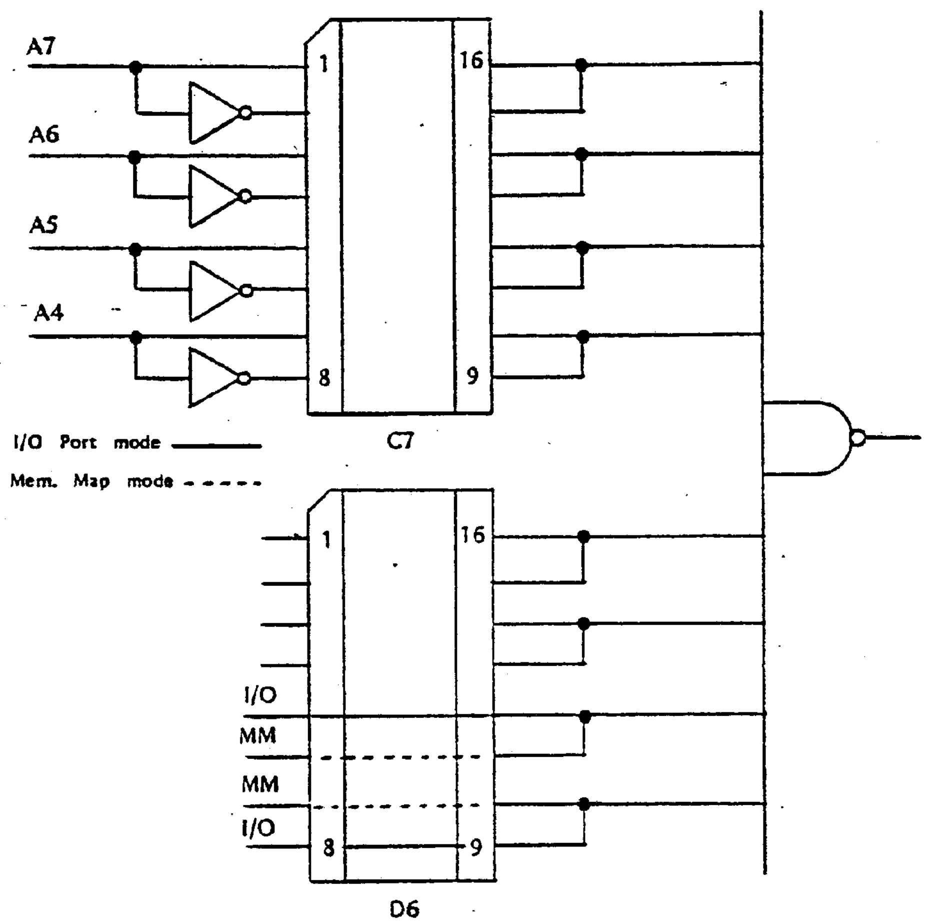
Data Rate Select Socket

The jumper socket in position B11 provides for selecting different baud rates for both Channel A and Channel B from the set of standard rates proided by the SIO board. The pin numbers and baud rates are indicated in the diagram.

The clock lines for Channel A and Channel B are completely independent and may be jumpered to the same rate or different rates.

When the chip is being used in the synchronous mode, the chip is running at a 1X clock rate rather than 16 X rate as in the asynchronous mode. Thus, the baud rates are 16 times as great for the same jumper location when used in the synchronous mode.

Board Address Selection Jumper Sockets



The board address is selected by jumpers or a DIP switch in locations C7 and D6. There are two cases for which this board may be jumpered: 1) to respond to input/output instructions and 2) to respond to memory access instructions. The case of input/output instructions will be treated first.

In selection location D6 pins 8 and 9 must be jumpered together and pins 5 and 12 must be jumpered together. The user must jumper socket C7 so when the desired I/O Port Address appears on the Address lines, the inputs to the NAND gate from bits A4 through A7 are high. If, for instance, address bit 6 is desired to be a 0 when the board responds, then pins 4 and 13 would be jumpered together. If address bit A6 was desired to be a 1, then either pins 3 and 14 may be jumpered together or 3 and 13 may be jumpered together, since 13 and 14 are tied to the common address selection input.

It is suggested, however, that when jumpers are being used, pins 3 and 13 be connected together to provide an easy visual indication of whether the address bit is a 1 or a 0 since that will correspond to whether the jumpers are slanted or straight across the jumper socket. Pins 13 and 14 were tied together so that an 8 position DIP switch can be inserted in this location and used to select the address. Address bits 4, 5, and 7 are jumpered in a similar manner on position C7.

See the diagram on the previous page for pin numbers for each address bit.

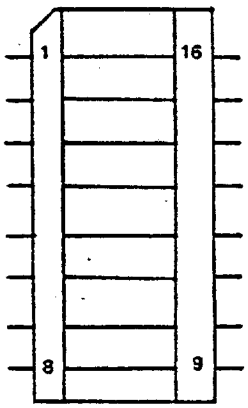
If it is desired to use the board in a memory-mapped I/O capacity, then in position D6 the jumpers between pins 8 and 9 and 5 and 12 must be removed and two jumpers inserted between pins 7 and 10 and between 6 and 11. The remaining jumpers for bits 4 through 7 function exactly the same and affect the lower eight bits of the memory address. The upper eight bits of the address will always be all ones, that is hex FE or octal 376.

When used as a memory-mapped I/O board, all instructions that normally affect the memory will operate on the I/O ports. For example, an increment memory instruction would read the data from the addressed input port, increment that data by one and output it on the same address output port.

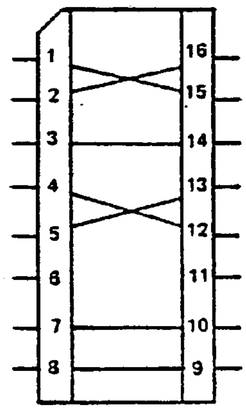
Example Jumpers -

To use the SIO Board in its simplest form, non-interrupted input/output instruction controlled, create jumpers as shown.

A1

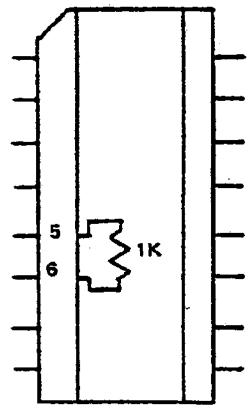


RS232 or Current Loop



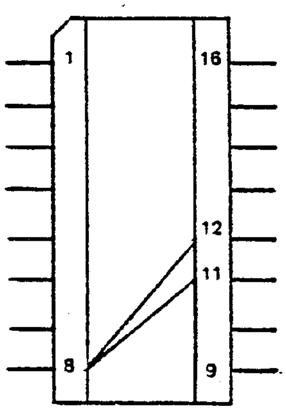
RS232

A3 (B8)

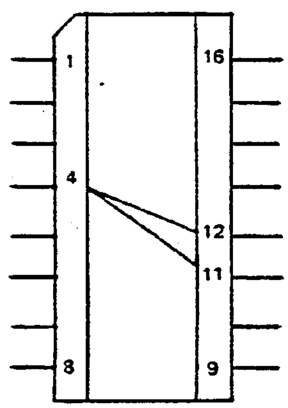


Current Loop

B11

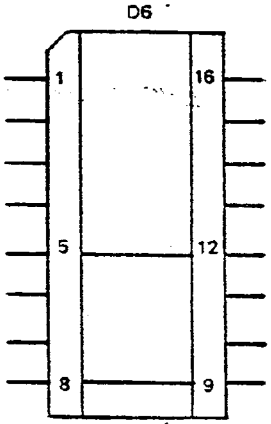


110 BAUD

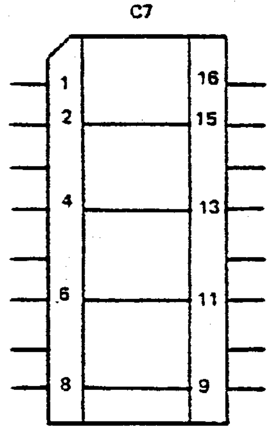


1200 BAUD

D6



C7



Select appropriate address-
Shown is 0- as required by SCS

Sample sequence to set up SIO for teletype and echo from keyboard to printer:

Format used is 2 stop bits, no parity, and 7 data bits. Reset IMSAI 8080 before running. Address and constants are in hexadecimal.

```
LIST
0010        MVI A,0CAH                MODE BYTE
0020        OUT 03
0030        MVI A,27H                 COMMAND BYTE
0040        OUT 03
0050 LOOP   IN 03                     READ CHAN A STATUS
0060        ANI 02                    MASK OUT ALL BUT RECEIVER READY
0070        JZ LOOP                   IF NOT READY LOOP
0080        IN 02                     READ CHAR
0090        OUT 02                    WRITE CHAR
0100        JMP LOOP
```

```
ASSM 3700

3700 3E CA        0010        MVI A,0CAH          MODE BYTE
3702 D3 03        0020        OUT 03
3704 3E 27        0030        MVI A,27H           COMMAND BYTE
3706 D3 03        0040        OUT 03
3708 DB 03        0050 LOOP   IN 03               READ CHAN A STATUS
370A E6 02        0060        ANI 02              MASK OUT ALL BUT RECEIV
370C CA 08 37     0070        JZ LOOP             IF NOT READY LOOP
370F DB 02        0080        IN 02               READ CHAR
3711 D3 02        0090        OUT 02              WRITE CHAR
3713 C3 08 37     0100        JMP LOOP
```

IMSAI

SIOC

SIOC BOARD

FUNCTIONAL DESCRIPTION

The IMSAI SIOC Board is a small optional board used with the Serial Interface (SIO Board). The SIOC provides user selection of any USART clock frequency from 15 Hz to 56 KHz.

The generated clock frequency is determined by a binary value set in two 16-pin jumper sockets. An additional jumper socket allows selection of either the SIOC or the standard SIO USART clocks to channels A and/or B.

Physically, the SIOC Board measures 5.2 X 2.2", and piggy-back mounts to a standard SIO Board. Mounting hardware and decoupling capacitors are provided with the SIOC Board.

SIOC BOARD

THEORY OF OPERATION

The SIOC board is a modulo -N clock divider, where N is user selectable. The SIOC divides down the 2MHz 8080 $\emptyset_2$ clock to a rate appropriate for the 8251 USART devices.

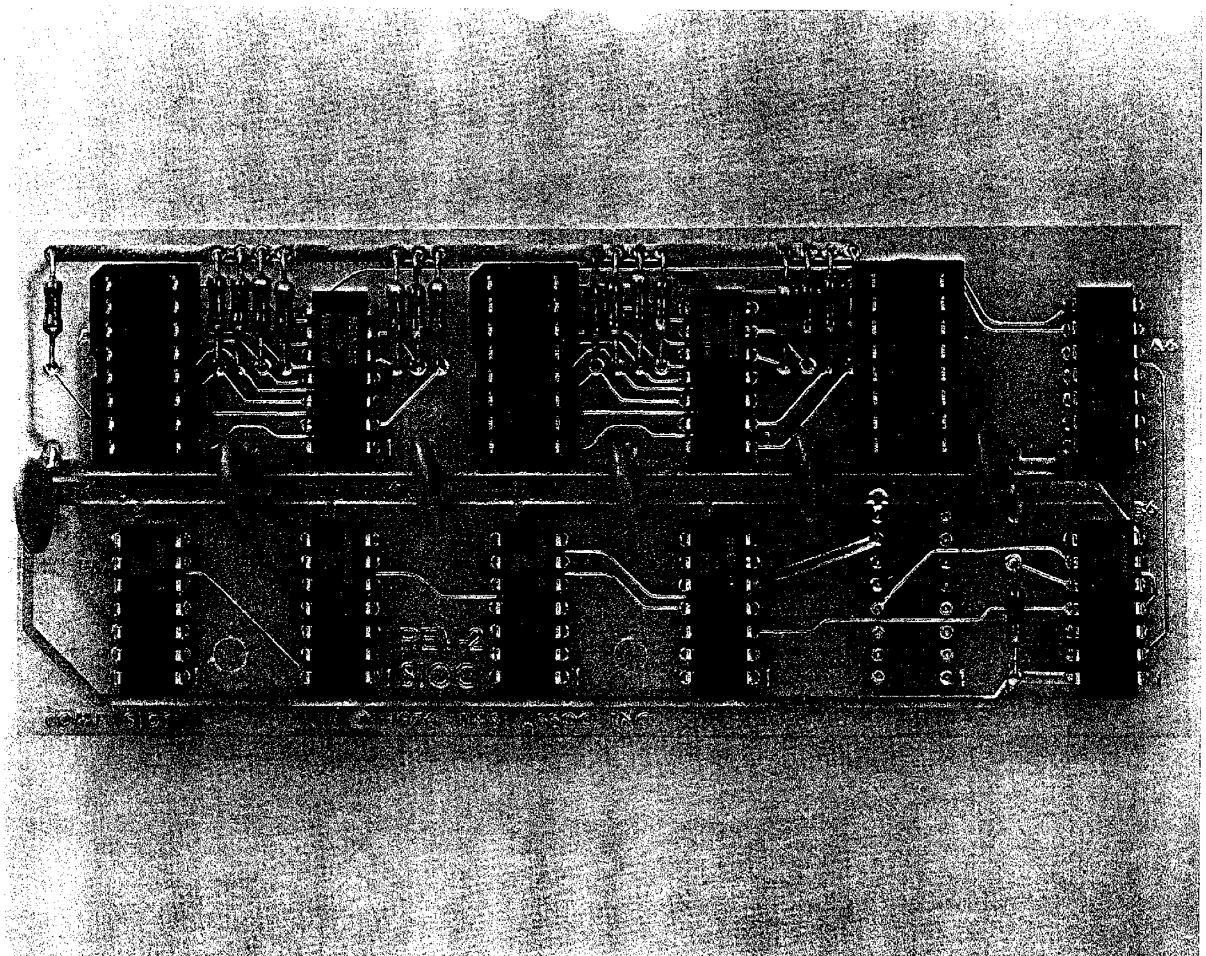
The four 7493 counters are arranged as a 16 bit ripple counter, clocked by the $\emptyset_2$ clock. Jumper sockets, A1 and A3 are jumpered so that when the selected binary value N (where N is selected to produce a desired final clock rate) is reached by the counter, the counter is reset. The 7430's and 7402 create the reset signal, while the 7474 is used to gate the reset pulse and create a symmetrical square wave output.

The final outclock frequency can be determined by:

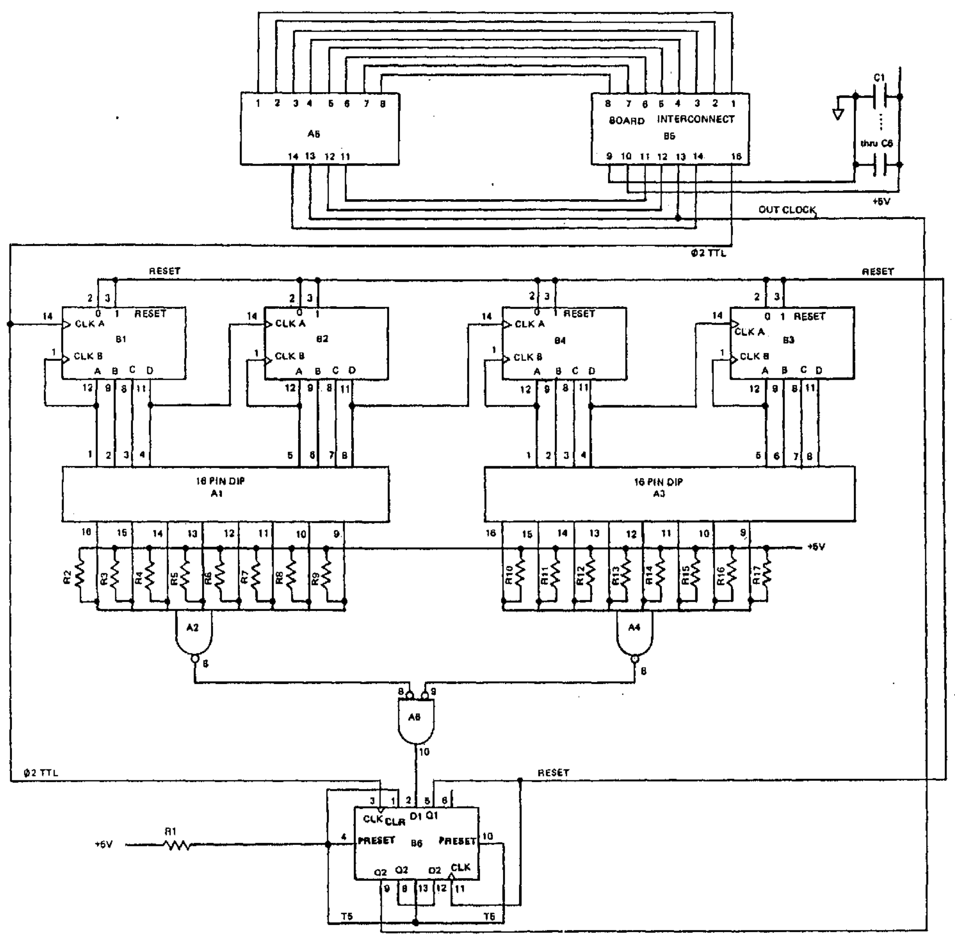
$$\text{Outclock (Hz)} = \frac{1}{2} \cdot \frac{2 \times 10^{6}\ \text{Hz}}{N}$$

Where N is the binary value in jumpers A1 and A3, and the factor of 1/2 is the result of the final ÷ 2 for the symmetrical output.

Decoupling capacitors are provided to eliminate the high frequency noise generated by the digital currents.



SIOC REV 2



A1, A3, A5	16 PIN HEADER SOCKET
A2, A4	7430
A6	7402
B1, B2, B3, B4	7493
B5	16 PIN BOARD INTERCONNECT
B6	7474
C1 thru C6	.1 mF
R1 thru R17	1K ¼W

IMS ASSOCIATES,
SCHEMATIC DIAG
SIOC REV 2 4/

BOARD: SIOC

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Solder	15-0000001	5'	
Washers	21-3390001	3	#6 Fiber Washers
Screw	20-3702001	3	6-32x3/4" Phillips Pan Head Machine
Nut	21-3120001	3	#6 Hex Nut
Spacer	21-3600002		#6, 7/16" Spacer
Socket	23-0800001	3	16 Pin Solder Tail Socket
Connector	23-0400009	1	16 Pin Board Interconnection
Header	23-0400001	3	16 Pin Integrated Circuit Header
Resistor	30-4100362	17	1K Ohm, ¼ Watt/brown, black, red
Capacitor	32-3010010	6	.1uF Disk Ceramic Capacitor
7402	36-0740201	1	Quad 2 Input NOR/SN7402N
7430	36-0743001	2	8 Input NAND/SN7430N
7474	36-0747401	1	Dual D Flip-Flop (Preset & Clear)/ SN7474N
7493	36-0749301	4	4 Bit Binary Counter/SN7493AN
PC Board	92-0000019	1	SIOC Rev. 2

ASSEMBLY INSTRUCTIONS

1) Unpack your board and check all parts against the parts list enclosed in the package.

2) If gold contacts on the edge connector appear to be corroded, use pencil eraser to remove any oxidation. NOTE: Do not use Scotchbright or any abrasive material as it will remove the gold plating.

3) Insert and solder each of the seventeen 1K Ohm, ¼ Watt resistors at locations R1 through R17 as shown on the Assembly Diagram.

IC INSTALLATION

NOTE: All IC's and 16 pin IC sockets orient pin 1 as indicated by the number 1 etched on the board.

4) Insert and solder one 7402 IC at location A6 as shown on the Assembly Diagram.

5) Insert and solder each of the two 7430 IC's at locations A2 and A4 as shown on the Assembly Diagram

6) Insert and solder one 7474 IC at location B6 as shown on the Assembly Diagram.

7) Insert and solder each of the four 7493 IC's at locations B1 through B4 as shown on the Assembly Diagram.

DISCRETE COMPONENTS INSTALLATION

8) Insert and solder each of the three 16 pin IC sockets at locations A1, A3, and A5 as shown in the Assembly Diagram. Plug in the three 16 pin headers.

9) Insert and solder each of the six .1uF disk capacitors at locations C1 through C6 as shown on the Assembly Diagram.

10) Insert and solder on the circuit side of the board one 16 pin board interconnection header B5 as shown on the Assembly Diagram (dotted lines).

ASSEMBLY INSTRUCTIONS (cont.)

TO MOUNT SIOC ONTO SIO BOARD

The SIOC mounting is accomplished by placing the fiber washers into each of the three holes in the SIOC on the component side and each of the three holes in the SIO board on the circuit side. Then insert the #6 screws through the fiber washer and spacers into the fiber washer of the SIO board. Plug in the board interconnect header during alignment before tightening screws.

SIOC BOARD

USERS GUIDE

The SIOC board allows the selection of any USART clock rate between 15 Hz and 56 KHz, allowing data transfer rates of .23 baud to 56K baud.

Designed to piggyback mount on the SIO board, the SIOC allows the user to select either the standard clock rates provided by the SIO board or the user-generated SIOC rate.

The SIOC may be easily mounted on the front of the SIO board, or the user may save one card slot by mounting the SIOC on the back of the SIO, and inserting the combination in the last slot of the mother board. To mount the SIOC on the back of the SIO board, the board rate select socket (B-11) must be soldered to the back of the SIO board, and the SIOC's binary divisor and clock jumper sockets should be mounted on the back of the SIOC board.

Jumper socket A-5 allows the selection of 8251 clock rates. Either the SIO standard rates or the SIOC generated rates may be jumpered to channels A or B. This jumper is arranged identically with B11 on the SIO board.

	9	8	110 A
	10	7	150 A
Chan. B Clk	11	6	300 A
Chan. A Clk	12	5	600 A
SIOC OUTCLK	13	4	1200 A
75 A	14	3	2400 A
	15	2	4800 A
ϕ_2	16	1	9600 A

As with socket B11 on the SIO board, jumpers should be placed to connect the appropriate clock to the proper channel clock input.

To select a binary divisor, determine the desired USART clock rate, remembering that this rate may be 1, 16 or 64 times the desired final baud rate, as program selected by the mode byte output to the USART.

The binary divisor, N, can be determined by

$$N = \frac{1}{2} \cdot \frac{2 \times 10^6 \text{ Hz}}{\text{Clock} \quad \text{Hz}}$$

This value should be converted to binary, and the jumpers (or switches) in A1 and A3 set so that the NAND input for every bit that should be a 1 is connected to the output bit of the counter. All other inputs (i.e., those desired to be a zero) should be left unconnected and pulled high. The selected clock rate will be on pin 13 of A5.

For example, to generate a 45 baud data rate for channel A (with a X16 USART clock), and to use channel B for a 1200 baud data rate, do as follows:

$$45 \text{ Hz} \times 16 = 720 \text{ Hz (Desired clock rate)}$$

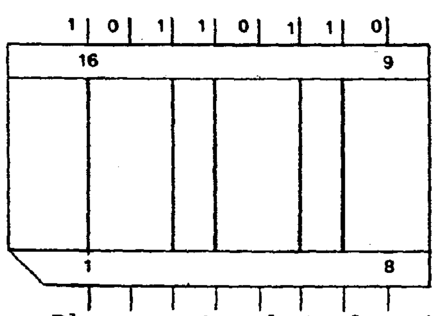
$$N = \frac{1}{2} \cdot \frac{2 \times 10^6 \text{ Hz}}{720 \quad \text{Hz}}$$

$$N = 1389_{10} \qquad N = 10101101101_2$$

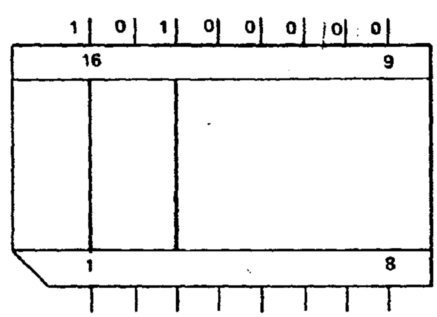
(to convert a decimal number to binary, see the attached note)

The jumpers should be jumpered as:

A1



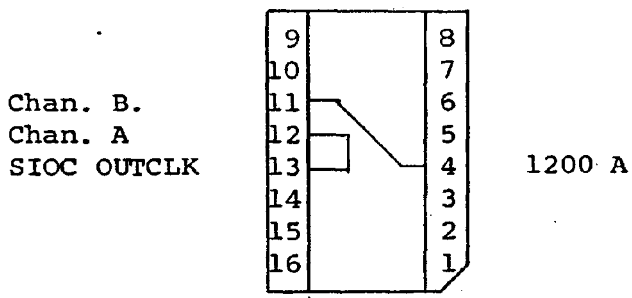
A3



Please note that the binary number appears as:

	A1	A3
	1 2 3 4 5 6 7 8	1 2 3 4 5 6 7 8
	1 0 1 1 0 1 1 0	1 0 1 0 0 0 0 0
	LSB	MSB

Jumper socket A5 as:



Decimal to binary number conversion:

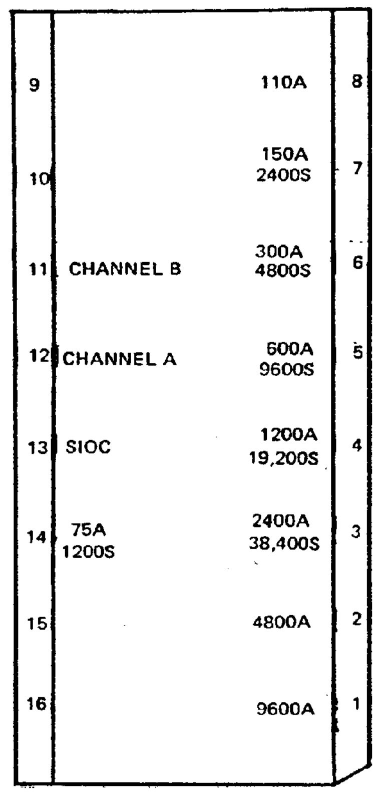
The simplest method to convert a number to binary is to divide it repeatedly by 2, recording the remainder for each step.

To convert the value 1389_{10} to binary,

Value	Value/2	Remainder	
1389	694	1	LSB (Least Significant Bit)
694	347	0	
347	173	1	
173	86	1	
86	43	0	
43	21	1	
21	10	1	
10	5	0	
5	2	1	
2	1	0	
1	0	1	MSB (Most Significant Bit)

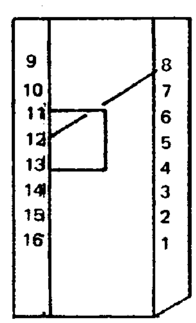
Note that the answer appears LSB first, thus, $1389_{10} = 10101101101_2$

Check: 1 + 4 + 8 + 32 + 64 + 256 + 1024 = 1389



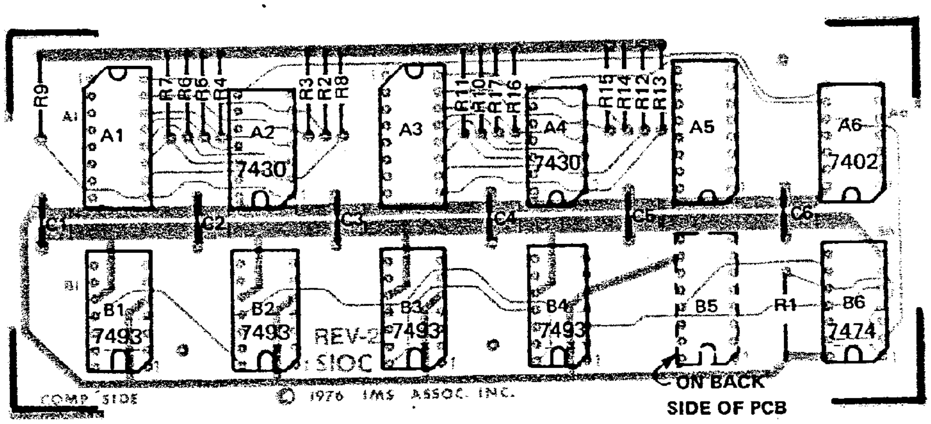
ASYNCHRONOUS RATES INDICATED "A"
ASSUME x16 CLOCK SELECTED IN 8251
SYNCHRONOUS RATES INDICATED BY "

SOCKET POSITON A5



EXAMPLE SHOWING CHANNEL A SET TO 110 BAUD
FOR A TELETYPE, AND CHANNEL B SET TO A
NON–STANDARD RATE SET BY SIOC JUMPERS IN A. . . A3

SIOC BAUD RATE SELECT JUMPERS



IMS ASSOCIATES, INC.
ASSEMBLY DIAGRAM
SIOC REV 2 4/76

IMSAI

PIC 8

PIC-8

FUNCTIONAL DESCRIPTION

The PIC-8 Priority Interrupt-Programmable Clock board provides the IMSAI 8080 Microcomputer System an eight level Priority Interrupt capability and a software-controlled interval clock.

The Priority Interrupt system utilizes the Intel 8214 Priority interrupt control unit and monitors the 8 Priority Interrupt lines on the 8080 back plane. The PIC-8 has the capability to service either single or multiple interrupt requests. When enabled and receiving an interrupt request, the Pic-8 determines if the request priority is higher than the software-controlled current priority, and if necessary issues a restart instruction that directs the 8080 system to one of eight priority controlled restart locations. For multiple interrupt requests, the 8214 determines the highest priority request, and processes it normally. It should be noted that the system does not store inactive requests, and that a peripheral device must hold an interrupt request until it is serviced by the microprocessor.

The current priority status register may be software set to any value desired to prevent low priority interrupts from being generated until the priority status register is reset to a lower value. The status register may be set to 0 if it is desired for all levels of interrupt to always occur.

The PIC-8 board also includes a clock circuit which provides programmed control at intervals ranging from .1 millisecond to 1 second. The program can select from among 3 jumper selected interval rates, or it can turn all three off. The 3 rates are jumper-selectable to any of the following values: .1 ms, .2 ms, 1 ms, 2 ms, 10 ms, 20 ms, 100 ms, 200 ms, or 1000 ms. Additionally, one bit of the DATA OUTPUT port is connected to a transistor and jumper pads for a special-purpose programmer-controlled output. Room is provided on the circuit board for a small speaker or other user-supplied circuitry. Also provided are 5 16-pin IC hole patterns with power and ground decoupling for special purpose user circuits. These hole patterns are drilled to accept wire wrap sockets.

Power on the board is regulated by an integrated circuit power regulator with current limiting protection. Tantalum ceramic bypass capacitors are supplied with the board. The board is G10-type double-sided laminate with plated through holes and contact fingers are gold-plated over nickel for reliable contact and long life.

PIC-8

THEORY OF OPERATION

Program control of the PIC-8 board is done entirely through one output port location. The address of this output port is jumper-selected in socket positions E4 and E5, and forms the input to the 8 input NAND gate (74ls30). The output of this address select is ANDed with the Processor Write Strobe and Phase II clock and provides an output strobe which is used to latch the lower 4 bits of output data into the 8214 priority interrupt chip, and the upper 4 bits into the 7475 4 bit latch.

When the 8214 is ENABLED and one of the priority request lines is low the 8214 sets the output of a 2 GATE Flip-Flop low to request an interrupt from the 8080 processor. When the processor acknowledges the interrupt the Flip-Flop reset and 3 buffer drivers of the 8T98 are enabled to put interrupt request address on bits 3, 4, and 5 of the DATA IN bus. The remaining bits of the DATA are not driven, and remain high via pullup resistors on the MPU Board. The byte thus formed on the DATA IN bus is a restart instruction. with bits 3, 4, and 5 disecting the processor to one of eight restart locators.

Further details on the theory of operation of the 8214 chip can be found in the Intel Data Book.

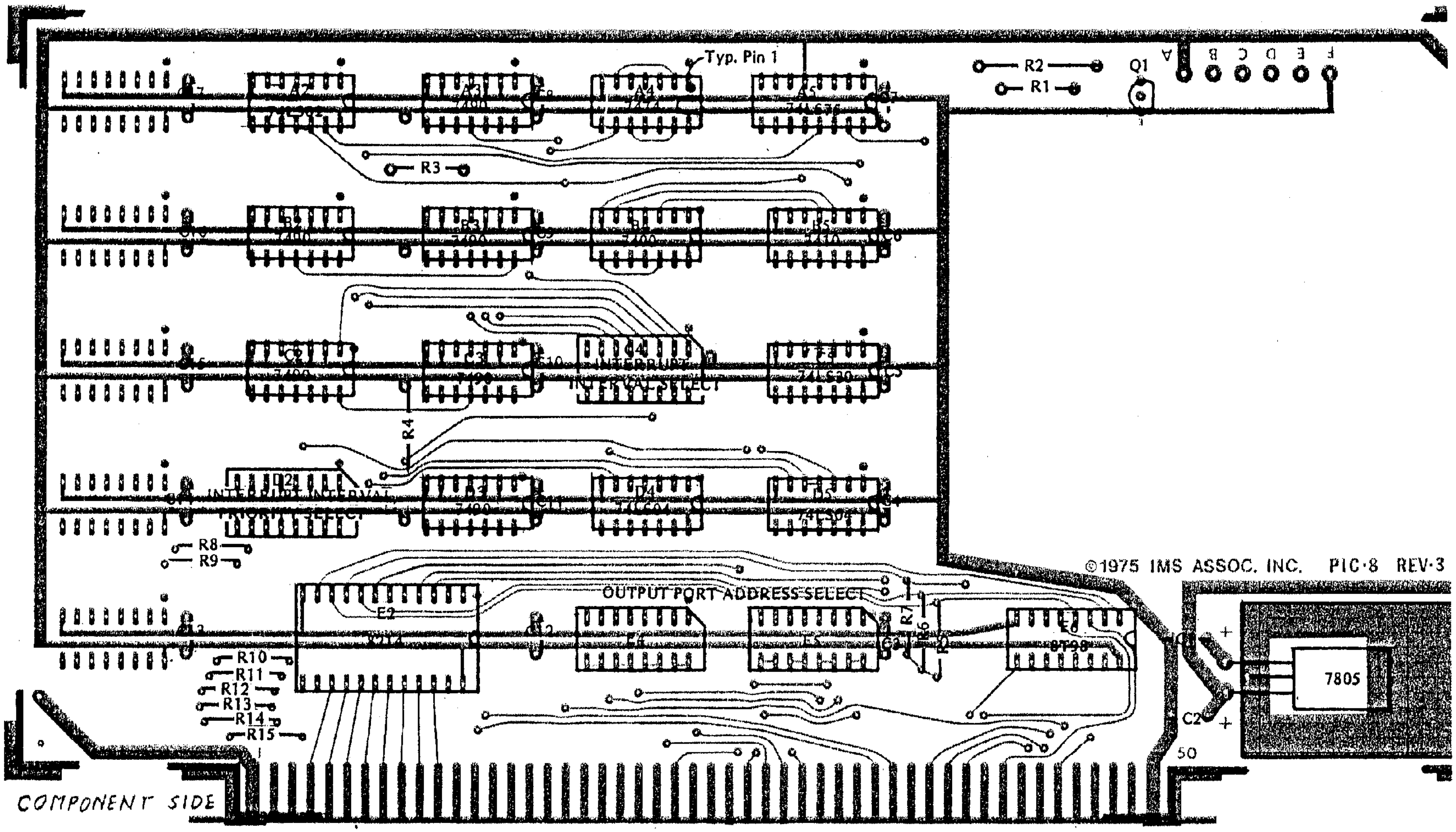
The PIC-8 board also includes a software controlled interval clock. The clock circuit takes the Phase II clock running at two megahertz and divides it by 200 using a divide-by-two (7474) followed by two divide-by-10 sections (7490) to provide the .1 millisecond intervals.

Four consecutive divide-by-10 7490's are then used to produce the other interval rates up to the longest rate of one second. Jumper selection is made from among these rates and ANDed with the output port bits 4, 5 and 6 and the output from the AND gate is used to drive the clock on the other half of the 7474 D type flip-flop. This section of the flip-flop is connected so that on successive clocks it will shift states and thus alternately request and remove the request for an interrupt.

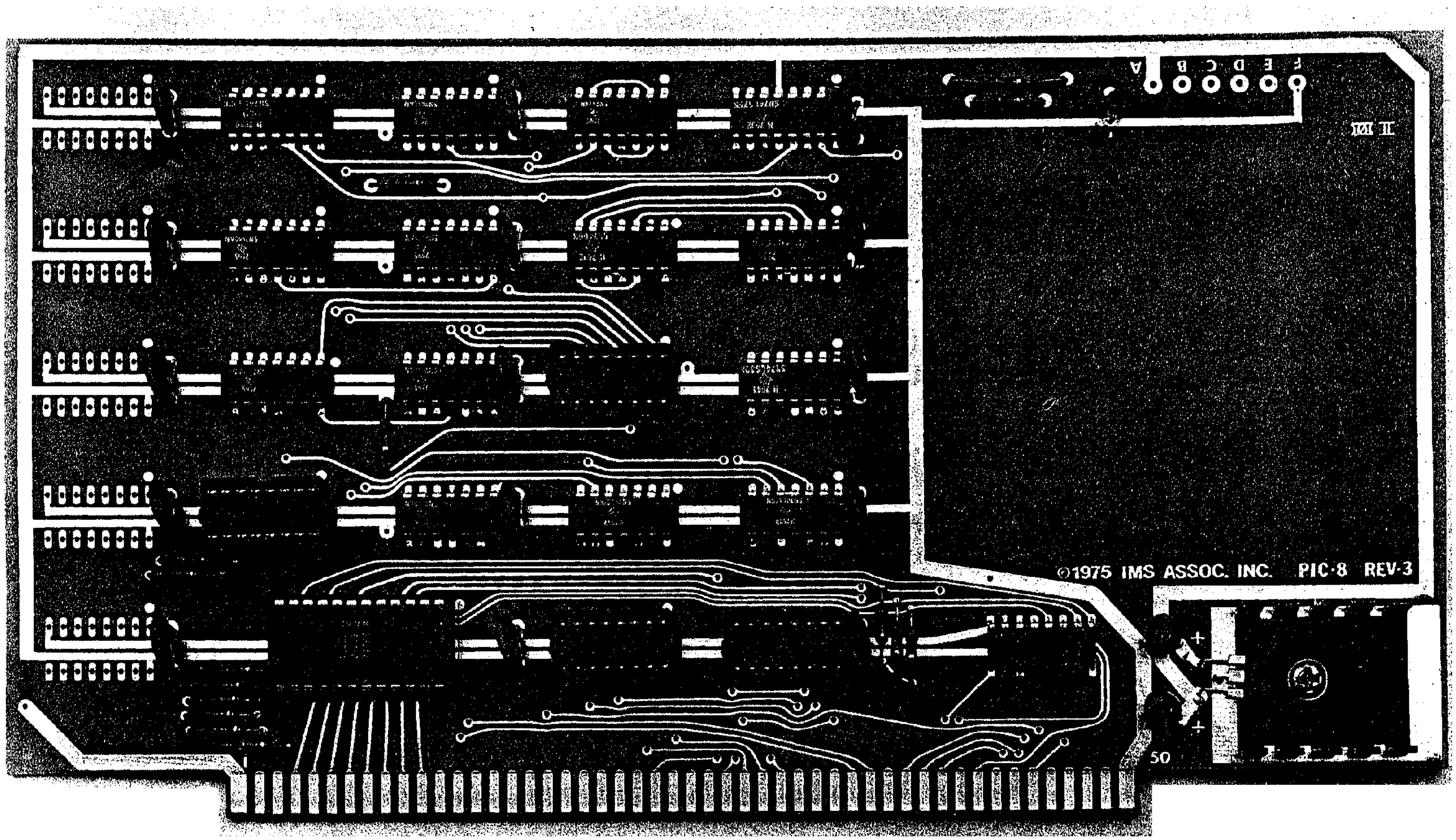
When the processor system is running, and replying to the interrupts, shortly after the request is issued, the interrupt acknowledge line will become active in the low state and set this flip-flop to remove the interrupt request so that the next time the clock line rises, the flip flop is again reset to request another interrupt. The interrupt request from this circuit is jumper-connected to any one of the priority interrupt lines and is handled by the 8214 circuitry exactly the same as any other peripheral board requesting an interrupt through the back plane would be.

Output bit 7 is used to drive the base of the transistor through a 1K resistor for current limiting, and the user supplied circuit to be driven is connected between the positive voltage and the collector current limiting resistor. Should just a voltage level be desired, as an output from this circuit, a resistor from 220 ohms to 1K ohm can be inserted in the collector circuit in the holes provided and a jumper placed between pads A and C to connect the top of the resistor to +5 volts. The output may be taken from point B which will be low when the bit is written as a 1 and will be high when the bit is written as a 0.

For a high impedance load, voltage swing will be nearly a full 5 volts for the high level and .3 volts for the low level. If a direct TTL level output is desired, it can be obtained from solder pad E if the 1K resistor in the base lead is removed and a jumper placed in its location and the transistor removed so as not to provide undesired load for a high level output.



IMS Associates Inc.
ASSEMBLY DIAGRAM
PIC 8 REV 3 2/76
2/27/76



PIC 8 REV. 3

BOARD: PIC-8

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
7400	36-0740001	1	Quad 2 Input NAND/SN7400N
74LS02	36-0740202	1	Quad 2 Input NOR (Low Power Schottky)/ SN7402N
74LS04	36-0740402	2	Hex Inverter (LPS)/SN74LS04N
7410	36-0741001	1	Triple 3 Input NAND/SN7410N
74LS30	36-0743002	1	8 Input NAND (LPS)/SN74LS30N
7474	36-0747401	1	Dual D Flip-Flop, Preset and Clear/ SN7474N
74LS75	36-0747501	1	Quad Bistable Latch (LPS)/SN74LS75N
7490	36-0749001	6	Decode Counter/SN7490AN
7805	36-0780501	1	5V Positive Voltage Regulator/MC7805CP
8214	36-0821401	1	Priority Interrupt Control Unit/ P8214/S1260
Transistor	35-2000002	1	NPN Transistor/2N3904
8T98	36-0089801	1	Hex Tri-State Buffer/8T98
Capacitor	32-2010010	15	.luF Disk Ceramic Capacitor
Heat Sink	16-0100002	1	Thermalloy/6106B-14
Header	23-4000001	4	16 Pin Integrated Circuit Header
PC Board	92-0000012	1	PIC-8, Rev. 1
Resistor	30-3220362	1	220 Ohm, ¼ Watt/red, red, brown
Resistor	30-4100362	14	lK Ohm, ¼ Watt/brown, black, red
Socket	23-0800001	4	16 Pin Solder Tail Socket
Socket	23-0800002	1	24 Pin Solder Tail Socket

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Capacitor	32-2233070	2	33-25 Tantalum Capacitor
Screw	20-3402001	1	6-32x3/8" Phillips Pan Head Machine
Nut	21-3120001	1	6-32 Hex Nut
Lockwasher	21-3350001	1	#6 Internal Star Lockwasher
Solder	15-0000001	5'	

ASSEMBLY INSTRUCTIONS

1) Unpack your board and check all parts against the parts lists enclosed in the package.

2) If gold contacts on the edge connector appear to be corroded, use pencil eraser to remove any oxidation. NOTE: Do not use Scotchbright or any abrasive material as it will remove the gold plating.

RESISTOR INSTALLATION

3) Insert and solder each of the fourteen 1K ohm 1/4 watt resistors (brown/black/red) R1, R3 through R15. See Assembly Diagram for locations.

4) Insert and solder the one 220 ohm 1/4 watt resistor (red/red/brown) R2. See Assembly Diagram for location.

IC INSTALLATION

NOTE: When looking at component side of the board with edge connector down, all IC pin 1's point to the right hand side of the board and are indicated by small dots on the board.

5) Insert and solder the one 7400 at location B4.

6) Insert and solder the one 74LS02 at location A2.

7) Insert and solder each of the two 74LS04's at location D4 and D5.

8) Insert and solder the one 7410 at location B5.

9) Insert and solder the one 74LS30 at location C5.

10) Insert and solder the one 7474 at location A4.

11) Insert and solder the one 74LS75 at location A5.

12) Insert and solder each of the six 7490's at locations A3, B2, B3, C2, C3, and D3.

13) Insert and solder the one 8T98 at location E6.

DISCRETE COMPONENT INSTALLATION

14) Insert and solder each of the four 16 pin IC sockets at locations C4, D2, E4, and E5.

15) Insert and solder the one 24 pin IC socket at location E2.

16) Plug in each of the four 16 pin IC jumper headers in the four

16 pin IC sockets.

17) Insert and solder each of the fifteen .1 uf disk capacitors C3 through C17. See Assembly Diagram for locations.

18) Insert and solder each of the two 33uf tantalum capacitors C1 and C2. See Assembly Diagram for locations. Observe polarity as indicated on board.

19) Insert and solder the one 2N3904 NPN transistor Q1, orient as indicated on Assembly Diagram.

HEAT SINK AND REGULATOR INSTALLATION

20) Bend the leads of the 7805 regulator at 90° angles approximately 1/4" from the bottom edge of the regulator to facilitate insertion on top of the heat sink.

21) Insert the #6 screw through the regulator and heat sink and attached washer and nut from back side of board. NOTE: Be sure to hold the heat sink in proper vertical position while tightening the screw in order to prevent shorting to adjacent traces. Solder regulator leads.

22) Finally, plug the one 8214 24 pin IC into the 24 pin IC socket located at E2.

NOTE:

1) The 16 pin IC socket and header located at C4 is to be used for interrupt interval select.

2) The 16 pin IC socket and header located at D2 is to be used for interval priority select.

3) The two 16 pin IC sockets and header located at E4 and E5 are to be used for output port address select.

USER GUIDE

Request for an interrupt appears at the PIC-8 board in the form of one of the eight priority interrupt request lines being pulled to a logic 0 level. The 8214 chip will recognize that one or more interrupts are being requested and it will determine which multiple request has the highest priority.

The eight priority levels are numbered 0 through 7, with 7 being the highest priority. The priority level of the highest current interrupt request is then compared against the value stored in the current priority status register in bits 0, 1 and 2. If the currently-requested priority level is equal to or lower than the value stored in the current priority status register, no interrupt will be generated.

If the priority interrupt being requested is 0 and the current priority status register contains a 0, no interrupt will be generated. Thus, if a 5 were stored in the current priority status register, then only interrupt levels 6 and 7 would generate an interrupt. Interrupt levels 5 and lower would not be acted upon at this time.

If the priority interrupt being requested is 0, and the current priority status register contains a 0, no interrupt would be generated as the priority level is not greater than that stored in the current priority status register. If the current priority status register data bit 3 is written as a 1, the compare to the current priority status register is overridden, and the request for an interrupt priority 0 is acted upon and an interrupt to restart position 0 is generated.

If other priority level interrupts are requested during the time that data bit 3 has been written as a 1 in the current priority status level, then the highest priority interrupt requested will be acted upon.

At any time, if there is more than one priority level of interrupt being requested, only the highest priority level is acted upon, and any interrupt requests not serviced must be held present until the system can return to them.

After each interrupt has been generated, and the processor has responded to it, it is necessary that the current priority status register be restored to either the same or a different value; otherwise, no further interrupts will be generated.

When interrupts are initially enabled in a system, the current priority status register should also be initialized to insure that the interrupt generating system will respond to an interrupt.

It should be noted that the current priority status register inputs data bits 0, 1, and 2, are input in the complement form. Further information on the operation of the 8214 priority interrupt and coding can be located in the Intel Data Book.

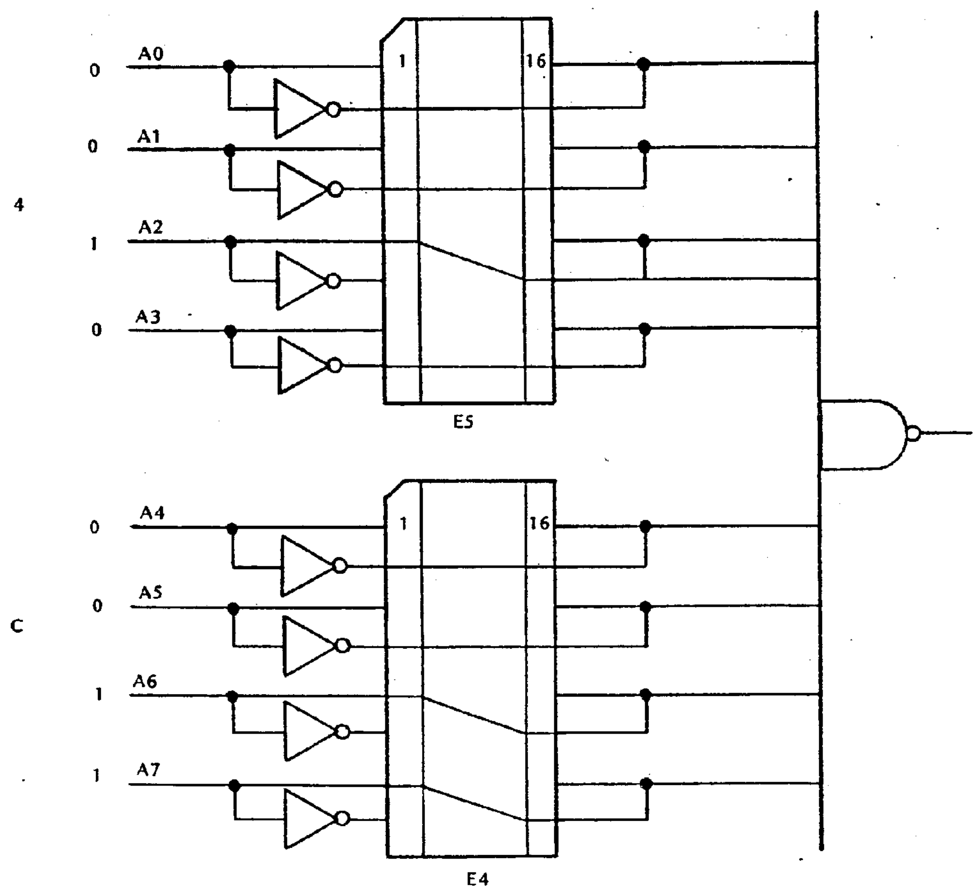
The program controlled clock's functions are selected by both user jumpers and software. After jumpers have been installed in the interval selection and priority select sockets, writing to the PIC-8's output port address can enable the clock circuitry. Data bits 4, 5, and 6 control the user-selected intervals.

In normal use, only one interval will be selected at a time; thus, only one of the three bits, 4, 5, and 6 in the output port will be 1 at a given time. If two or more of these bits are written 1 at the same time, then the different rates will interact and interrupts will not occur continuously at the highest rate, but will occur at the highest rate for only portions of the time and not at all during other portions of the time as determined by the specific rates selected. For example, if both the rates 1 millisecond and 1 second are selected at the same time, one millisecond interrupts will be received for 1/2 of one second and then no interrupts will be received for the second half of that second and this pattern will repeat every second.

Should an interval interrupt not be acted upon in the time remaining between it's occurrence and the occurrence of the following interval interrupt request, the interrupt request will be taken away at the following pulse, and the request will again be asserted on the second interval following the first. This pattern of requesting an interrupt every other interval will continue until the system is able to respond to the interrupts within the time period required.

Whenever a byte is output to select or change the selection of the interrupt interval, it must be remembered that the lower 4 bits of the same output byte affect the interrupt generating circuitry, and will set it so that it is ready to respond to the next interrupt. The desired value for the current priority status register, must be present in the output bytes lower 4 bits every time a bit is output for any purpose, whether it is to select or change the selection of the interrupt interval desired, or whether it is to change the current priority status register, or to output a bit 7 to the special purpose circuitry supplied by the user. Similarly, any time the output byte is used to set or change the current priority status level, bits 4, 5, and 6 must be also output according to the desired interrupt interval selected. Any bit which is written without changing does not cause any momentary glitches or other effects.

BOARD ADDRESSING



Positions E4 and E5 contain the user-jumpered 16-pin address selection sockets. These jumpers allow the PIC-8 board to respond to any 1 of the 256 possible I/O port addresses.

As shown on the schematic, to enable the CRI board it is necessary to have all eight inputs to the 74LS30 (C5) high. The user should select the desired address, and then jumper the address selection sockets so that when that address appears on address lines A0 through A7, all the NAND inputs are high, and the board is then enabled.

Each socket contains values of 4 lines and their complements. Socket E5 controls lines A0 through A3. Socket E4 controls lines A4 through A7. If the user-selected address presents a 1 on an address line, that line should be directly connected to the NAND input via a short wire jumper on the socket header. Conversely, if the user selected address presents a 0 on an address line, the inverted address line value should be connected to the NAND.

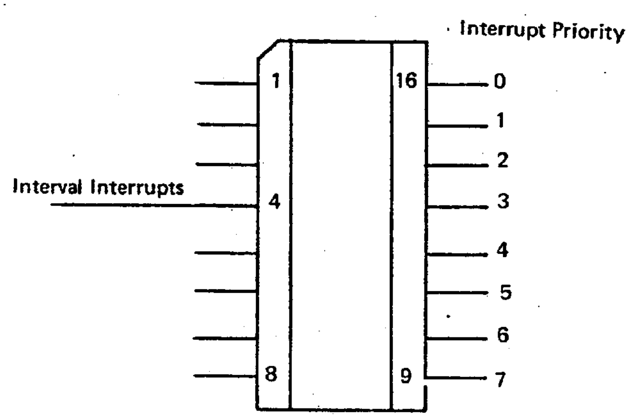
It is suggested that for lines jumpered to enable on a 1 value that the jumpers be placed diagonally across the socket (i.e., Pin 1 to Pin 15) and for lines jumpered for a 0 value, the jumper be placed straight across the header (i.e., Pin 2 to Pin 15). This convention allows easy visual determination of the selected address, for 1's appear as diagonals and 0's as horizontals. An example of a correctly jumpered socket pair for the address C4 hex or 304 octal is shown above.

If desired, very frequent address changes may be easily implemented through the exchange of an 8 pole DIP switch for each socket.

All 8 of the NAND inputs should be jumpered to respond to either a 1 or a 0. While any input left unconnected will appear to act as a 1, open inputs are very susceptible to noise pulses.

PRIORITY SELECT FOR THE INTERVAL GENERATING CIRCUIT

In position D2, the jumper socket permits the selection of the priority level at which the interrupts generated by the interval clock circuit will occur. The interrupt request level from the interval clock circuit appears on pin 4 of the jumper socket, and the eight available priority levels inputs appear on pins 9 through 16 of the jumper socket. A jumper should be placed between pin 4 and the pin corresponding to the priority level desired for the interval clock's interrupts.



CLOCK INTERVAL SELECTION

While 3 interrupt intervals may be program selected on the PIC-8 board, jumper selection from among the nine available interrupt intervals must be made in the jumper socket in position C4 to choose with three interrupt intervals the program is capable of selecting among. As indicated in the diagram, Pins 12, 13, and 14 on the jumper socket are the three inputs to the interrupt generating circuitry from among which port bits 4, 5, and 6 are used to select one or more of the levels to be active. A high level on

A transistor is provided to permit bit 7 to drive small loads such as a speaker or relay, and space is provided on the end of the board for such a device to be mounted. Connections to the driving transistor are made through solder paths available at the top of the board. If a small flat speaker is attached to the board here, and connected to the driving transistor, it will permit program control of audio pulses or tones which, in connection with the interrupt intervals available, allows for precise control of such things as musical notes rather than the empirical methods such as setting a small receiver beside the system and trying various different loops just to see what one gets. Since the program would have positive control of such a speaker, the output may be precisely calculated, and any modifications made with the assurance of the desired results.

The maximum current drawn by this output transistor should be held to 200 milliamperes or below. Provision is made on the board for a resistor in series with the open collector of this output transistor to limit the current to the desired value. A small speaker may be driven audibly by inserting a 220 ohm resistor in series with the collector and connecting the speaker between this resistor and the +8 volts on the board, using solder pads B and C.

UCRI BOARD

FUNCTIONAL DESCRIPTION

The September, 1975 "Computer Bits" column in Popular Electronics proposed a standard for the recording on magnetic tape of hobbyist's digital information. The column included a description of the proposed standard (known as the Hobbyist Interchange Tape System), circuitry to interface the tape recorder and digital system, and a listing of sample programs to drive the interface. The IMSAI UCRI Board provides the hobbyist this circuitry and all necessary logic to control the interface from the 8080 system bus. The only additional hardware required to read or write information is an audio-quality tape recorder and 2 patch cords with RCA phono jacks.

Tri-state buffer drivers and low power Schottky TTL circuits are used for completely 8080-bus-compatible plug-in interface. The UCRI Board requires one Mother board slot, and occupies one I/O port address. Jumper options permit the user to select any one of the 256 possible addresses.

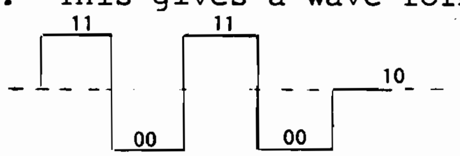
Physically, the board measures 3" x 10" and connects to the computer with a standard 100 pin edge connector. An on-card voltage regulator with current limiting and high-quality tantalum by-pass capacitors are included.

For further details of the HIT Digital Recording Standard and system, please refer to the September, 1975 Popular Electronics, pages 57-61. You will note that our circuit and software are slightly different than PES because we found that theirs had a few problems.

THEORY OF OPERATION

The UCRI board is a direct implementation of the circuit published in the "Computer Bits" column of the September 1975 issue of Popular Electronics, with the additional logic needed to run this circuit from the IMSAI 8080 system.

The interface circuit published uses the LM3900 Quad-Op amp with Norton current sensitive inputs. (See the enclosed notes). The output circuit uses one section of the LM3900 as a buffer amplifier with a gain of approximately 2/5ths and an integrating capacitor to reduce the response to higher frequencies, thus rounding the corners of the square waves produced by the program through the output bits 0 and 1. The signal is then fed to the output jack to the tape recorder through a DC blocking capacitor and stepped down to a lower level to permit connection to a microphone input with a resistive divider. Square waves are put out by alternately outputting bits 0 and 1 as both 1's. Then both 0's. When the signal is stopped, bits 0 and 1 should be written as one "1" and one "0", to give an intermediate output level. This gives a wave form as illustrated.



The input section of the interface uses the remaining three sections of the LM3900. Two of the sections are used as inverting gain-of-10 and non-inverting gain-of-10 amplifiers to buffer the signal. With the proper input voltage levels, one member of this amplifier pair will saturate and chip on the positive going peak of the signal, and give 0 volts output during the negative-going peak. The other member of this pair does the same but with the opposite peaks; it also inverts so that both outputs are positive. A diode is connected from each of these sections to the input of the third section, which is connected with positive feedback as a Schmitt trigger, to produce approximately a square wave at the output.

This square wave will be high if either of the 2 previous amplifiers' outputs are high; thus, if a tone signal is present, the Schmitt trigger output will be high except for the very brief moments that the signal crosses through zero. C14 acts as a filter and holds the Schmitt input on for these short periods, insuring a constant high output from the Schmitt trigger. Of course, if the tone input ceases, C14 discharges and the Schmitt output falls; thus, the input circuit produces a 1 (high) output during the time it receives a tone input, and a 0 (low) output when it receives a blank spot of tape.

To interface this circuit to the 8080 system, the UCRI board's logic must recognize the proper address and enable the board. The board is enabled when properly addressed (74LS30 output low) and either of the signals SINP (STATUS INPUT) or SOUT (STATUS OUTPUT) is high. The inputs of the 74LS30 (B8) are determined by the selection of the address jumper wiring in positions B7 and B9.

When these conditions are met, the PROCESSOR READY line is enabled (via B4, Pin 13) and held high to indicate no WAIT cycles are necessary. If the processor strobes the DBIN (DATA BUS IN) line, the input bit from the Schmitt trigger is put on the DATA IN bus, while if the processor strobes the WR line, the 7475 latch stores the data on the DATA OUT bus and presents it to the UCRI board's input circuitry.

A 7805 voltage regulator powers the UCRI board circuitry and by-pass capacitors are used to eliminate any high-frequency noise from the power bus.

Notes on Norton Current Op-Amps

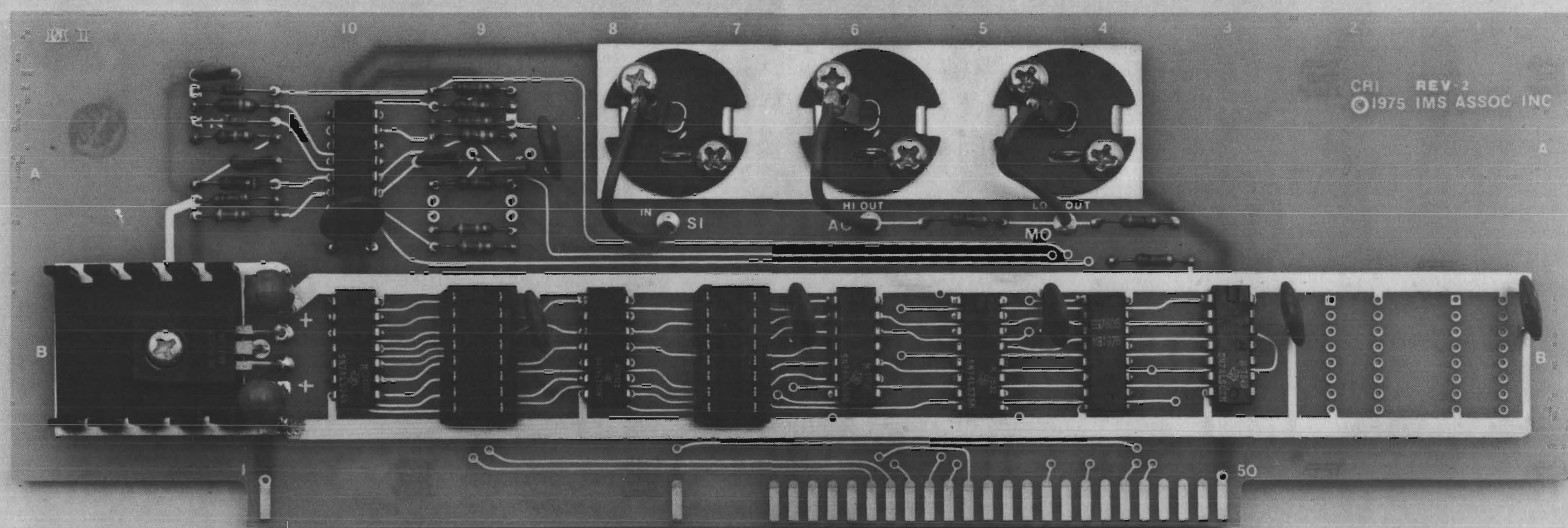
For those unfamiliar with Norton Current Amplifiers, it should be noted that the current-sensitive inputs are different from the inputs in most integrated circuit operational amplifiers in that they are sensitive to current rather than voltage. Thus while a normal integrated circuit Op-amp can have a common mode voltage, and input currents are always extremely low, the LM-3900 op-amp always has an input voltage of approximately 0 volts, and it can have a common mode current.

One of the clearest circuits to indicate the difference between the Norton input operational amplifier and a conventional input operational amplifier is the non-inverting amplifier used to clip the signal from the audio tape recorder. This is the section using pins 11 and 12 as inputs and 10 as the output.

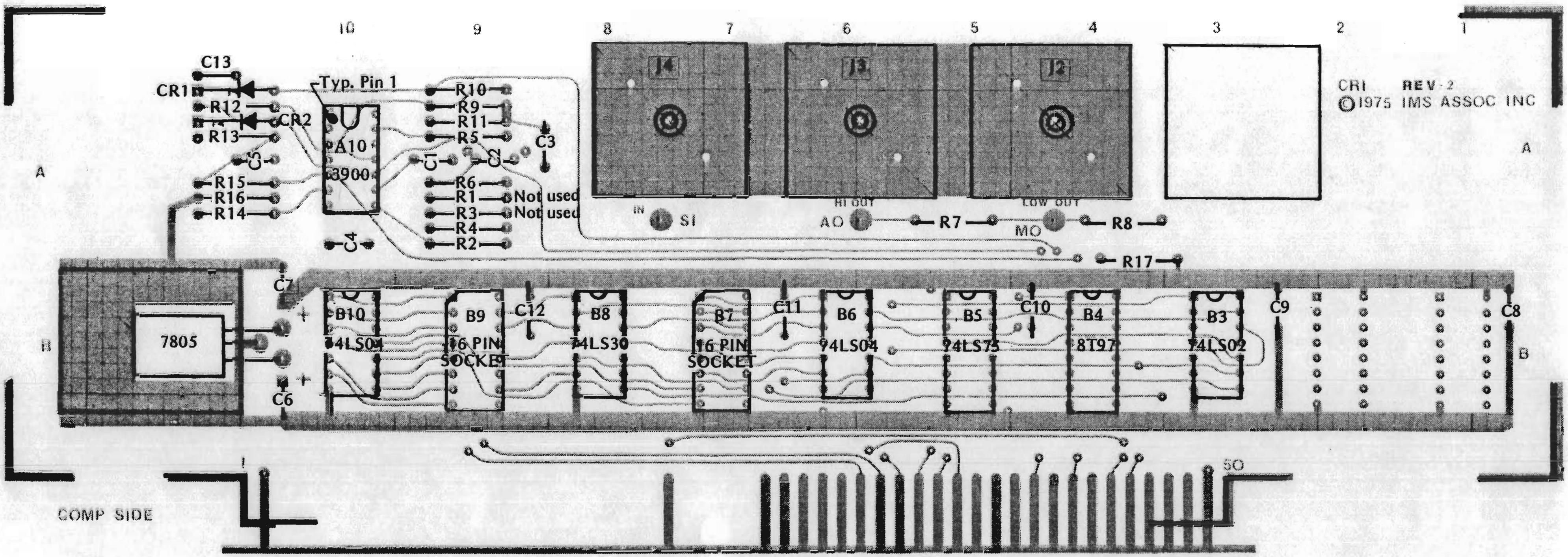
If this operational amplifier were the conventional type, the input currents would be extremely low and, therefore, the voltage drop across both the 22K input resistor to Pin 12 and the 220K feedback resistor to Pin 11 would be approximately 0 volts, because of the very low current.

The negative feedback loop would cause the operational amplifier to adjust its output voltage to equal the input voltage and the circuit would perform as a one-to-one voltage follower. However, with a Norton input operational amplifier, both inputs are always held at approximately ground potential, thus the current through the 22K ohm resistor is 10 times as great as that through the 220K ohm resistor for the same voltage. In other words, if one has 1/10th of a volt at the input producing a current into Pin 12 of the operational amplifier through the 22K ohm resistor, one needs one full volt at the output to produce the same current through the 220K ohm resistor into the negative input of the amplifier in order to balance the inputs according to current levels.

Thus, with a Norton Input Amplifier with input sensitive to current levels, this circuit produces an amplification factor of 10.



UCRI REV. 2



IMS Associates Inc.

ASSEMBLY DIAGRAM

UCRI REV. 2 3/76

ASSEMBLY INSTRUCTIONS

Begin assembly by inserting the 15 resistors into the proper locations on the board. Verify proper values and positions, and solder. (Note: Resistors R1 and R3 are omitted, leave these spaces empty). Next, insert the two 1N914 Diodes (CRI, CR2). Check to be sure the cathode (banded end) is oriented as shown (toward the square pad), and solder. Insert the seven integrated circuits; check for proper location, bent pins, and correct orientation (Pin 1 towards top of board, away from edge connector). Solder the seven IC's.

Install the six .1 uf by-pass capacitors and the other five ceramic disc capacitors. Check for proper values and locations, and solder them in place. Fasten the 7805 and heat sink (with #6 screw, lockwasher, and nut), and solder the 7805 in place. Install the two .33 uf tantalum by-pass capacitors; observe polarity (+ to +), and solder.

The RCA phone jacks should now be inserted into the appropriate holes and fastened with #6 screws, lockwashers and nuts.

Short wires should now be run from the pads on the board (SI, AO, MO) to the center connectors of each of the RCA phono jacks The 16-pin sockets should be inserted and soldered in locations B7 and B9 and the board is now ready for use.

USER GUIDE

The UCRI board user should refer to the Popular Electronics Article about the HIT system. The circuit published in that article is reproduced exactly on the UCRIboard with the addition of the logic necessary to interface the circuit to the 8080 system. The user must select a board address (see Jumper Socket instructions) and provide the software to run the UCRI board.

DATA OUT BITS 0 and 1 (DO 0, DO1) are used to create the recording signal. The software routine should write out a square wave (equal time of both 1's and both 0's) at approximately 2000 H_z on these bits. While it was suggested by Popular Electronics to use another data bit as an "envelope" to control the 2000 Hz square wave, we have found it works better to control the square wave in software. In order to leave the output signal at 0 volts when the square wave is stopped, bits 0 and 1 should be written as one "1" and one "0".

To write to tape, connect the tape recorder's Auxillary or Mic inputs to the appropriate UCRIboard jacks. The presence of the 2000 Hz square wave will write a tone into the tape, while the absence of the square wave will write a blank onto the tape.

To read from tape, connect the recorder's output jack to the SIGNAL INPUT jack of the UCRI Board. The presence of the tone on tape will produce a 1 signal (high) on INPUT BIT 0, while the presence of a blank spot of tape will produce a 0 (low) on INPUT BIT 0. If this bit is always read as a 1, then the hook-up should be checked for excessive noise or hum which may be causing the receive circuit to always indicate the presence of a signal. The UCRI Board provides 4 output bits and 4 input bits so that should for any special purpose the user desire to output or input more bits of information, such as to turn a recorder on or off, the capability is already built into the board. All that is needed is the interface between the TTL levels provided and the function desired to be controlled.

The traces presently on the board* are arranged so that all 4 bits input to the same section of the LM3900. Using binary weighted values of resistors for R1, R2, R3 and R4 will result in a 4-bit digital to analog converter; permitting arbitrary 16-level wave shapes to be created by software and output to an audio system. We suggest the values:

		Controlled By
R3	500K or 470K	Bit 3
R1	1.0 M	Bit 2
R4	2.0 M	Bit 1
R2	4.0 M or 3.9 M	Bit 0

The circuit can also be used for the more recent Byte Magazine-sponsored audio cassette data interchange standards. A software routine can be written which will output either 8 cycles of 2400 Hz or 4 cycles of 1200 Hz as required for mark or space bits. The filter circuit will properly condition these signals for recording with no changes at all. However, the receive circuit must now detect whether the signal from the recorder is above or below 0, rather than detect just the presence or absence of a signal. With this recording standard, there is always a signal and what is detected is the frequency of this signal. If the software routine could sense at any given instant whether the signal was above or below 0, then software timing loops could then determine whether it was changing at a 1200 or 2400 Hz rate.

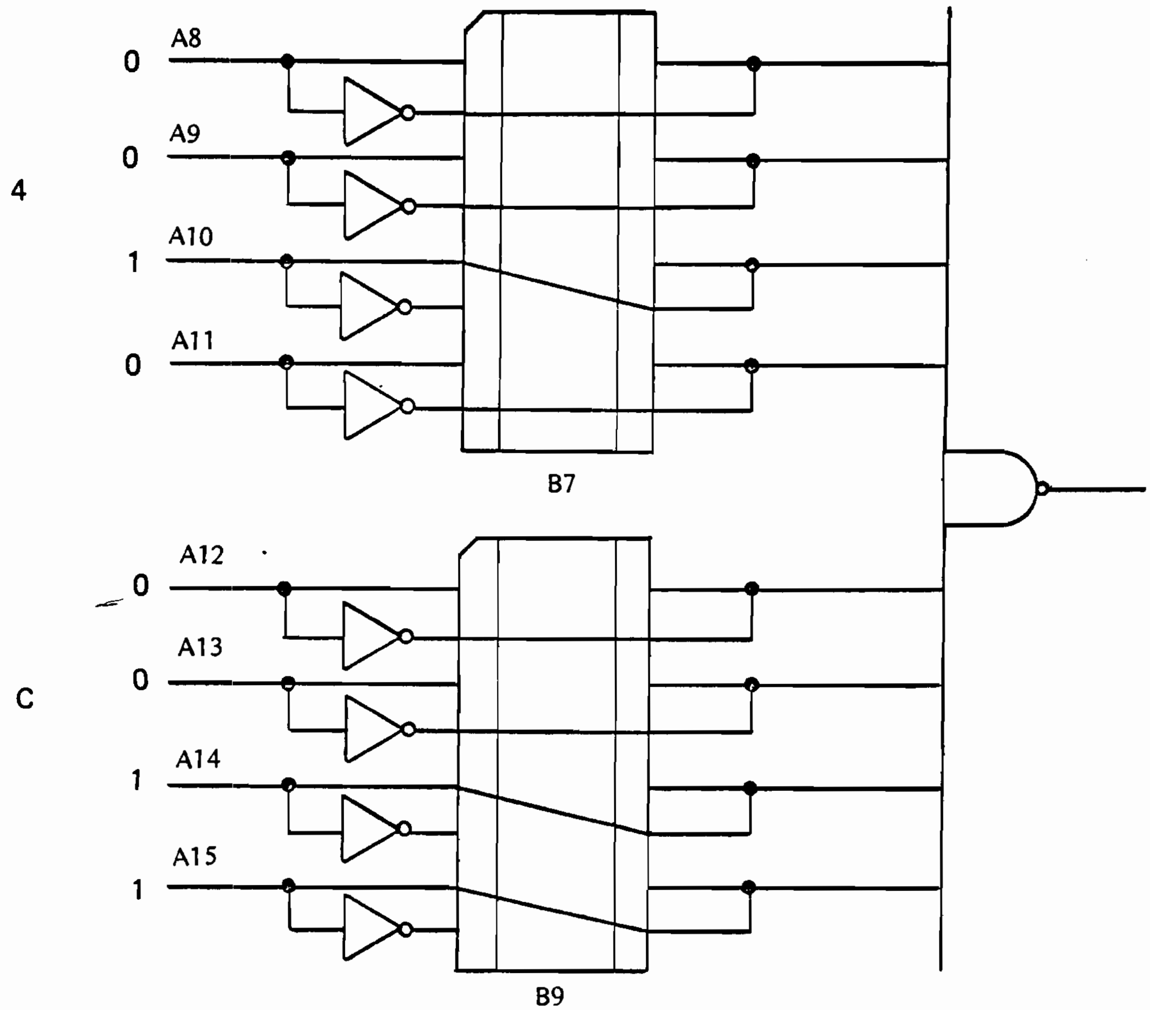
To achieve this function, the input circuit may be used in either of two ways. For the first modification, remove diode CR2 and capacitor C13. The signal on INPUT BIT 0 will now be a square wave of the same frequency as the recorded tone. Software could determine this frequency, and subsequently decide if the data is a 1 or a 0. This is the preferred modification for using the Byte standard.

*Rev. 2; earlier revisions can be easily modified to do this by cutting (R3 & R4) free of (A10 pin 13 & R5); then connecting (R3 & R4) to (R1 & R2 and A10 pin 8 and R6 & C1).

Alternatively, the second method may be used. Pin 4 will be high during positive swings of the input, while pin 10 will be high during negative swings. These points are connected to data in bits 1 and 2 on the board, producing complementary "0's" and "1's" readable by the software. These inputs, however, have not gone through a Schmitt Trigger so that the signals will not be as well defined as the bit 0 signal. By detecting simultaneous zero crossing (the two signals are complements), software is again able to determine the frequency, hence value, of the recorded data. This modification leaves the UCRI board to function normally and, also, has fair noise immunity due to the simultaneous zero-crossing detection scheme.

Listings of basic routines to run both types of recording follow.

BOARD ADDRESSING



Positions B7 and B9 contain the user-jumpered 16-pin address selection sockets. These jumpers allow the UCRI board to respond to any 1 of the 256 possible I/O port addresses.

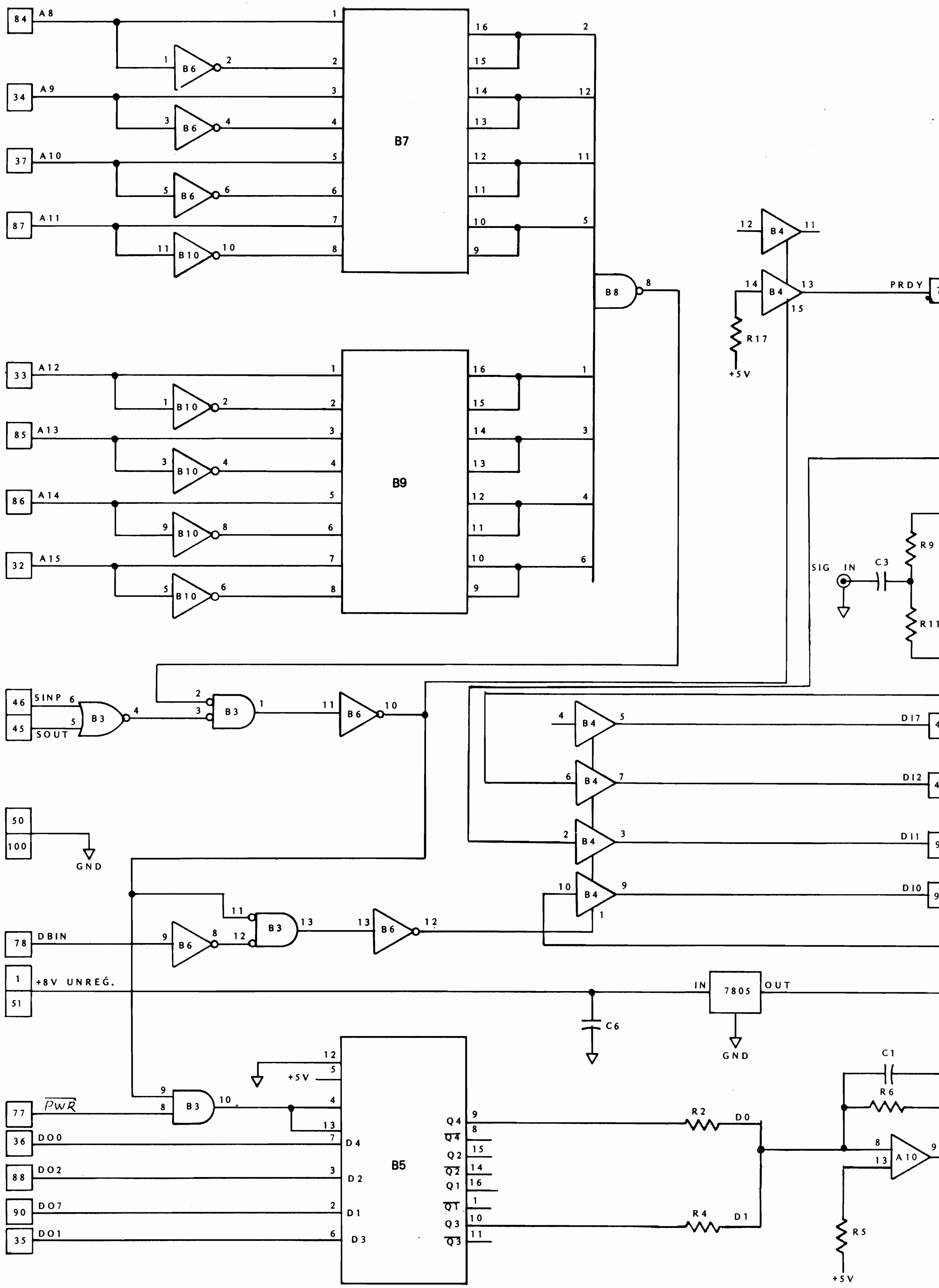
As shown on the schematic, to enable the UCRI board it is necessary to have all eight inputs to the 74LS30 (B8) high. The user should select the desired address and then jumper the address selection sockets so that when that address appears on address lines A8 through A15, all the NAND inputs are high, and the board is then enabled.

Socket B7 controls lines A8 through A11 while socket B9 controls lines A12 through A15. Each socket contains values of 4 lines and 4 line complements. If the user-selected address presents a 1 on an address line, that line should be directly connected to the NAND input via a short wire jumper on the socket header. Conversely, if the user selected address presents a 0 on an address line value should be connected to the NAND.

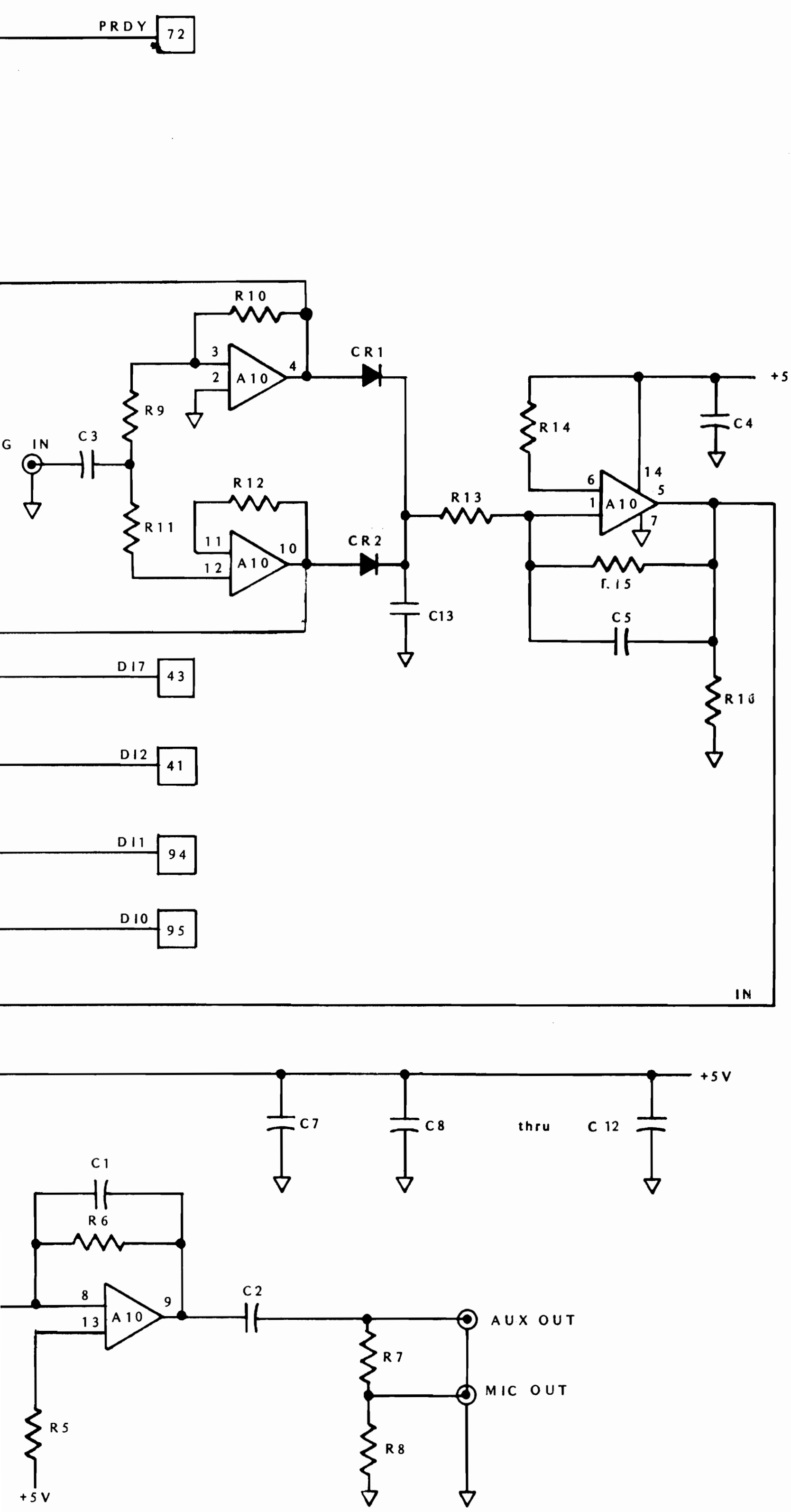
It is suggested that for lines jumpered to enable on a 1 value that the jumpers be placed diagonally across the socket (i.e., Pin 1 to Pin 15) and for lines jumpered for a 0 value, the jumper be placed straight across the header (i.e., Pin 2 to Pin 15). This convention allows easy visual determination of the selected address, for 1's appear as diagonals and 0's as horizontals. An example of a correctly jumpered socket pair for the address C4 hex or 304 octal is shown above.

If desired, very frequent address changes may be easily implemented through the exchange of an 8 pole dip switch for each socket.

All 8 of the NAND inputs should be jumpered to respond to either a 1 or a 0. While any input left unconnected will appear to act as a 1, open inputs are very susceptible to noise pulses.



A8
A9
A10
A11
A12
A13
A14
A15
B6
B7
B8
B9
B10
B3
B4
B5
SINP
SOUT
GND
DBIN
+8V UNREG.
7805
IN
OUT
C6
PWR
DO0
DO2
DO7
DO1
D4
D2
D1
D3
Q4
Q2
Q1
Q3
R2
R4
D0
D1
A10
C1
R6
R5
+5V
PRDY
R17
SIG IN
C3
R9
DI7
DI2
DI1
DI0



A10	3900
B3	74LS02
B4	8T97
B5	75LS75
B6 B10]	74LS04
B7 B9]	16 PIN DIP SOCKET
B8	74LS30
C1	470pF
C2	.47mF
C3	.005mF
C4 C8 thru C12]	.1mF
C5	47pF
C6 C7]	33mF
C13	.001mF
CR1 CR2]	IN914
J2 thru J4	PHONO PLUGS
R2 R4	470K ¼W
R3	Not used
R14	470K¼W
R5	330K¼W
R6 R13]	100K¼W
R7 R16 R17]	IK¼W
R8	47¼W
R9 R11]	22K¼W
R10 R12]	220K¼W
R15	IM¼W

IMS ASSOCIATES INC.
SCHEMATIC DIAGRAM
UCRI REV 2 3/76

3 2/76

UCRI SOFTWARE

FUNCTIONAL DESCRIPTION

The UCRI Driver is a 512-byte program containing all code necessary to generate the signal to be written to tape and to interpret the signal when read from the tape.

The Driver has routines which write/read HIT standard records in blocked or unblocked format, dump/load 4K blocks of memory and test the write and read operation of the recorder. The write/read record routine, CRIOD, is substantially the same as in Revision 0, except that a checksum has been added at the end of each blocked record. Tapes written with the Revision 0 Driver should be readable with the Revision 1 Driver.

THEORY OF OPERATION

The UCRI Driver uses one subroutine, WRITB, to perform all output to the UCRI Board. WRITB encodes one byte of data into a signal to be written on tape. Each byte is output as a string of eight data bits, least significant bit first, followed by one stop bit, which is always a zero. Each bit is represented by a frame of 6 milliseconds duration. A 2000 Hz signal is generated by the Driver at the start of each frame by alternately outputting a "11" and a "00". The last part of each frame is a null signal, which the Driver generates by outputting a "10". A zero is represented by a short signal, lasting for less than half the frame, and a one by a long signal, lasting for more than half the frame.

All input is performed by the subroutine READB, which reads a byte from tape. When the tape is playing, the UCRI Board sets or resets status bit 0 to indicate the presence or absence, respectively, of the 2000 Hz signal. READB samples this bit and determines the value of each data bit by comparing the duration of each signal with the time between signals. After assembling a complete byte, READB waits for a stop bit. An alternative entry at READA is used to read the first sync character of each record and will not wait for a stop bit until a sync has been found.

In unblocked mode, the specified number of bytes is copied from memory directly to the tape without formatting.

In blocked mode, the data is preceded by a header and followed by a trailer. The detailed format of each block is as follows:

HEADER				TRAILER	
SYNC	STX	LENGTH	DATA	ETX	CHECK
32 BYTES	1 BYTE	1 BYTE	255 BYTES MAX	1 BYTE	2 BYTES

The fields are:

SYNC	32 SYN characters (16 hex). Used to make sure the program is synchronized with the data.
STX	A Start-of-Text character (02 hex).
LENGTH	The length of the data field.
DATA	The data copied from memory.
ETX	An End-of-Text character (03 hex).
CHECK	The sum of all data bytes plus 255.

It is recommended that blocked format be used, if only to insure synchronization.

USER GUIDE

The UCRI Driver has been assembled to use input/output port FC, so the address selection jumpers on the UCRI Board should be wired for FC (see the UCRI Board User Guide). The Driver has been assembled to originate at FA00.

The Driver is non-standard and has the following five entry points:

ADDRESS	ROUTINE
FA00	Write/Read Record
FA03	Dump Memory
FA06	Load Memory
FA09	Test Write
FA0C	Test Read

The first routine must be CALLed since is concludes with a RET. The other routines end in infinite loops and may be run from the front panel.

The first-time user should run the Test Write routine for several minutes on a blank tape. The Test Write routine writes the same character (the initialization character mentioned in the listing) over and over. It should be possible to hear the signal while it is being recorded.

After running the Test Write routine for several minutes, rewind the tape and run the Test Read routine. The Test Read routine reads bytes off the tape and displays them in the PROGRAMMED OUTPUT lights. The volume adjustment is critical. Since only one character was written, the lights should not flicker, but should produce a steady pattern. The Test Read routine has no way of finding a byte boundary so the character displayed may be shifted by one or more bits. And, since the stop bit written after each data byte is always a zero (see Theory of Operation), an extra zero will be inserted into the displayed byte if it is shifted. The volume that results in a steady pattern in the lights should always be used when reading tapes.

The Write/Read Record routine, CRIOD, requires parameters in the following registers:

B reg — Status Byte: Bit 0 = 0 to write, 1 to read; Bit 7 = 0 for unblocked format, 1 for blocked.

C reg — Length of data buffer.

HL reg — Address of first byte of buffer.

A typical calling sequence to write a blocked record of 100 bytes starting at location 1200 hex might be:

```
CRIOD EQU     0FA00H    ;ADDRESS OF ROUTINE
      MVI     B,80H     ;GET CODE FOR BLOCKED WRITE
      MVI     C,100     ;GET COUNT
      LXI     H,1200H   ;GET ADDRESS OF DATA
      CALL    CRIOD     ;WRITE RECORD
                        ;RETURN HERE
```

On return, carry is cleared unless an error occurred. Errors are only detected when reading. If carry is set, the A reg will hold one of the following error codes:

Code	Error
1	No initial sync characters found
2	More than 32 sync characters found
3	No STX character found
4	No ETX character found
5	Record longer than buffer
6	Checksum error

The Dump/Load Memory routines input the starting block address and the number of blocks to be dumped/loaded in the left and right nybbles, respectively, of the PROGRAMMED INPUT byte. Both routines require RAM in the block following the area of memory to be dumped/loaded. For example, if the user wished to dump 8K of memory starting at B000 he would set the switches for B2. Memory at B and C would be dumped and RAM would be needed at D.

It has been found that recorders with automatic volume control take 5 to 10 seconds to adjust the volume when recording. To compensate, the Dump routine begins by writing a number of initialization characters which are ignored by the Load routine.

The Dump/Load routines make repeated calls to CRIOD to effect the transfer of blocked records.

The procedure for dumping memory is as follows:

1. Stop the computer.

2. Set the ADDRESS switches for FA03 and press RESET and EXAMINE.

3. Set the PROGRAMMED INPUT switches for the block of memory to be dumped.

4. Start the tape and wait about 10 seconds.

5. Start the program.

While data is being written, the high order byte of the address of the byte being written will be displayed in the PROGRAMMED OUTPUT lights. When the program is complete, the lights will be turned off.

The procedure for loading memory is as follows:

1. Stop the computer.
2. Set the ADDRESS switches for FA0C, press RESET, EXAMINE and RUN.
3. Start the tape and let it run until a stable pattern emerges in the PROGRAMMED OUTPUT lights.
4. Pause the tape.
5. Stop the computer.
6. Set the ADDRESS switches for FA06, press RESET and EXAMINE.
7. Set the PROGRAMMED INPUT switches for the block of memory to be loaded.
8. Start the program.
9. Restart the tape.

While the data is being read, the high order byte of the address of the byte being read will be displayed in the PROGRAMMED OUTPUT lights. If an error occurs, the error code will be flashed. If the program completes successfully, the lights will be turned off.

```
FA00                    0000 ; ******** UCRI CASSETTE INTERFACE DRIVER ********
FA00                    0010 ;
FA00                    0020 ; REVISION 1        27 JUL 76        BRH
FA00                    0030 ;
FA00                    0050 ;
FA00                    0060 ; I/O PORTS
FA00                    0070 ;
FA00                    0100 ;
FA00                    0110 ; TAPE INITIALIZATION PARAMS (WHEN WRITING DUMMY
FA00                    0120 ;   CHARS TO STABILIZE AUTOMATIC VOLUME CONTROL)
FA00                    0130 ;
FA00                    0160 ;
FA00                    0170 ; MISCELLANEOUS PARAMETERS
FA00                    0180 ;
FA00                    0250 ;
FA00                    0260 ; SPECIAL CHARACTERS
FA00                    0270 ;
FA00                    0310 ;
FA00                    0320 ; ERROR CODES
FA00                    0330 ;
FA00                    0400 ;
FA00                    0410 ; ENTRY POINTS (THIS DRIVER IS NON-STANDARD).
FA00                    0420 ;
FA00 C3 6E FA           0430            JMP     CRIOD    ;ORIGINAL DRIVER ENTRY POINT
FA03 C3 0F FA           0440            JMP     DUMP     ;DUMP MEMORY TO TAPE
FA06 C3 14 FA           0450            JMP     LOAD     ;LOAD MEMORY FROM TAPE
FA09 C3 E4 FB           0460            JMP     TESTW    ;TEST WRITE ROUTINE
FA0C C3 EA FB           0470            JMP     TESTR    ;TEST READ ROUTINE
FA0F                    0480 ;
FA0F                    0490 ; PICK UP PROPER STATUS BYTE FOR DUMP/LOAD.
FA0F                    0500 ;
FA0F 06 80              0510 DUMP:      MVI     B,80H    ;BLOCKED WRITE STATUS BYTE
FA11 C3 16 FA           0520            JMP     START    ;START COMMON CODE
FA14 06 81              0530 LOAD:      MVI     B,81H    ;BLOCKED READ STATUS BYTE
FA16                    0540 ;
FA16                    0550 ; INITIALIZE FOR WRITE/READ RECORD LOOP.
FA16                    0560 ;
FA16 0E 80              0570 START:     MVI     C,128    ;LENGTH OF EACH RECORD
FA18 DB FF              0580            IN      FRONP    ;READ SWITCHES
FA1A 57                 0590            MOV     D,A      ;SAVE TEMPORARILY
FA1B E6 F0              0600            ANI     0F0H     ;ZERO LOW NIBBLE
FA1D 5F                 0610            MOV     E,A      ;SAVE HIGH BYTE OF 1ST ADDR
FA1E 7A                 0620            MOV     A,D      ;GET SWITCHES BACK
FA1F 93                 0630            SUB     E        ;ZERO HIGH NIBBLE
FA20 CA 4A FA           0640            JZ      EXIT     ;EXIT IF COUNT IS ZERO
FA23 57                 0650            MOV     D,A      ;SAVE COUNT OF 4K INC'S
FA24 07                 0660            RLC              ;SWAP NIBBLES...
FA25 07                 0670            RLC
FA26 07                 0680            RLC
FA27 07                 0690            RLC
FA28 83                 0700            ADD     E        ;COMPUTE AND SAVE HIGH BYTE
FA29 67                 0710            MOV     H,A      ;  OF RAM ADDRESS
FA2A 2E 50              0720            MVI     L,80     ;SET SP HIGH ENOUGH TO LEAVE
FA2C F9                 0730            SPHL             ;  ROOM FOR DESCENDING STACK
FA2D 63                 0740            MOV     H,E      ;POINT HL AT 1ST MEMORY ADDR
FA2E 2E 00              0750            MVI     L,0      ;  TO BE DUMPED/LOADED
FA30 78                 0760            MOV     A,B      ;PUT STATUS BYTE IN A
FA31 1F                 0770            RAR              ;PUT WRITE/READ BIT IN CARRY
```

```
FA32 D4 5E FA            CNC     INIT    ;IF WRITING, INIT AUTO VOL CTRL
FA35                ;
FA35                ; LOOPS TO:
FA35                ;    1) DUMP/LOAD 4K INC'S UNTIL COUNT EXHAUSTED, AND
FA35                ;    2) WRITE/READ 32 RECORDS.
FA35                ;
FA35 1E 20          LOOP1:   MVI     E,32    ;# OF 128-BYTE BLOCKS IN 4K
FA37 7C             LOOP2:   MOV     A,H     ;PUT HIGH ADDRESS BYTE IN A
FA38 2F                      CMA             ;COMPLEMENT FOR OUTPUT
FA39 D3 FF                   OUT     FRONP   ;SET LIGHTS
FA3B CD 6E FA                CALL    CRIOD   ;WRITE/READ A 128-BYTE RECORD
FA3E DA 4A FA                JC      EXIT    ;EXIT IF ERROR
FA41 1D                      DCR     E       ;COUNT RECORDS DOWN
FA42 C2 37 FA                JNZ     LOOP2   ;LOOP UNTIL ALL RECORDS DONE
FA45 15                      DCR     D       ;COUNT 4K INC'S DOWN
FA46 C2 35 FA                JNZ     LOOP1   ;LOOP UNTIL DONE
FA49 AF                      XRA     A       ;DONE, SET LIGHTS OFF
FA4A                ;
FA4A                ; EXIT ROUTINE. FLASH COMPLETION CODE IN A.
FA4A                ;
FA4A 2F             EXIT:    CMA             ;COMPLEMENT FOR OUTPUT
FA4B 57                      MOV     D,A     ;SAVE OUTPUT CODE IN D
FA4C 26 FF                   MVI     H,0FFH  ;PUT 00 OUT CODE IN H
FA4E 7C             OUT:     MOV     A,H     ;GET CODE TO OUTPUT
FA4F D3 FF                   OUT     FRONP   ;SET LIGHTS
FA51 EB                      XCHG            ;EXCHANGE CODES IN D & H
FA52 01 00 40                LXI     B,4000H ;GET DELAY COUNTER
FA55 0B             DELAY:   DCX     B       ;START OF DELAY LOOP
FA56 78                      MOV     A,B     ;IS COUNT ZERO?...
FA57 B1                      ORA     C
FA58 C2 55 FA                JNZ     DELAY   ;LOOP UNTIL IT IS
FA5B C3 4E FA                JMP     OUT     ;OUTPUT OTHER CODE
FA5E                ;
FA5E                ; IN ORDER TO LET THE AUTOMATIC VOLUME CONTROL
FA5E                ;    STABILIZE, WRITE A NUMBER OF DUMMY INIT CHARS.
FA5E                ;
FA5E F5             INIT:    PUSH    PSW     ;SAVE REGISTERS TO BE USED...
FA5F C5                      PUSH    B
FA60 0E FF                   MVI     C,INCNT ;GET INIT CHAR COUNT
FA62 3E 33                   MVI     A,INCHR ;GET INIT CHAR
FA64 CD 21 FB       INIT0:   CALL    WRITB   ;WRITE IT
FA67 0D                      DCR     C       ;COUNT DOWN
FA68 C2 64 FA                JNZ     INIT0   ;LOOP UNTIL COUNT EXHAUSTED
FA6B C1                      POP     B       ;RESTORE SAVED REGISTERS...
FA6C F1                      POP     PSW
FA6D C9                      RET
FA6E                ;
FA6E                ; HIT STANDARD CASSETTE RECORDER I/O DRIVER.
FA6E                ;
FA6E F3             CRIOD:   DI              ;INSURE ACCURATE TIMING
FA6F C5                      PUSH    B       ;SAVE REGISTERS TO BE USED...
FA70 D5                      PUSH    D
FA71 78                      MOV     A,B     ;PUT STATUS BYTE IN A
FA72 0F                      RRC             ;PUT WRITE/READ BIT IN CARRY
FA73 DA 96 FA                JC      CREAD   ;JUMP IF READING
FA76 78                      MOV     A,B     ;PUT STATUS BYTE IN A
FA77 07                      RLC             ;PUT BLOCKED BIT IN CARRY
FA78 DC C6 FA                CC      WRITH   ;WRITE HEADER IF BLOCKING
FA7B 11 FF 00                LXI     D,0FFH  ;INITIALIZE CHECKSUM
FA7E 0C                      INR     C       ;PRIME COUNT FOR LOOP
```

```
FA7F 0D              WLOOP:   DCR    C        ;COUNT DOWN
FA80 CA 8E FA                 JZ     WTAIL    ;EXIT LOOP IF NO MORE DATA
FA83 7E                       MOV    A,M      ;PUT NEXT BYTE IN A
FA84 23                       INX    H        ;BUMP POINTER
FA85 CD BD FA                 CALL   CKSUM    ;ADD BYTE TO CHECKSUM
FA88 CD 21 FB                 CALL   WRITB    ;WRITE BYTE
FA8B C3 7F FA                 JMP    WLOOP    ;LOOP UNTIL ALL DATA OUT
FA8E 78              WTAIL:   MOV    A,B      ;PUT STATUS BYTE IN A
FA8F 07                       RLC             ;PUT BLOCKED BIT IN CARRY
FA90 DC AA FB                 CC     WRITT    ;WRITE TAIL IF BLOCKING
FA93 C3 B9 FA                 JMP    CDONE    ;ALL DONE, RETURN
FA96 78              CREAD:   MOV    A,B      ;PUT STATUS BYTE IN A
FA97 07                       RLC             ;PUT BLOCKED BIT IN CARRY
FA98 DC DF FA                 CC     READH    ;READ HEADER IF BLOCKED DATA
FA9B DA BA FA                 JC     ERROR    ;EXIT IF ERROR
FA9E 11 FF 00                 LXI    D,0FFH   ;INITIALIZE CHECKSUM
FAA1 0C                       INR    C        ;PRIME COUNT FOR LOOP
FAA2 0D              RLOOP:   DCR    C        ;COUNT DOWN
FAA3 CA B1 FA                 JZ     RTAIL    ;EXIT LOOP IF NO MORE DATA
FAA6 CD 69 FB                 CALL   READB    ;READ A BYTE
FAA9 77                       MOV    M,A      ;STORE IT IN BUFFER
FAAA 23                       INX    H        ;BUMP POINTER
FAAB CD BD FA                 CALL   CKSUM    ;ADD BYTE TO CHECKSUM
FAAE C3 A2 FA                 JMP    RLOOP
FAB1 78              RTAIL:   MOV    A,B      ;PUT STATUS BYTE IN A
FAB2 07                       RLC             ;PUT BLOCKED BIT IN CARRY
FAB3 DC B8 FB                 CC     READT    ;READ TAIL IF BLOCKED DATA
FAB6 DA BA FA                 JC     ERROR    ;NOTE ERROR
FAB9 AF              CDONE:   XRA    A        ;NORMAL RETURN, RESET FLAGS
FABA D1              ERROR:   POP    D        ;RESTORE SAVED REGISTERS...
FABB C1                       POP    B
FABC C9                       RET
FABD                 ;
FABD                 ; ADD BYTE IN A TO CHECKSUM IN DE.
FABD                 ;
FABD F5              CKSUM:   PUSH   PSW      ;SAVE A
FABE 83                       ADD    E        ;ADD TO LOW BYTE
FABF 5F                       MOV    E,A      ;STORE NEW LOW BYTE
FAC0 3E 00                    MVI    A,0      ;RESET A
FAC2 8A                       ADC    D        ;ADD CARRY TO HIGH BYTE
FAC3 57                       MOV    D,A      ;STORE NEW HIGH BYTE
FAC4 F1                       POP    PSW      ;RESTORE A
FAC5 C9                       RET
FAC6                 ;
FAC6                 ; WRITE HIT STANDARD BLOCK HEADER.
FAC6                 ;
FAC6 F5              WRITH:   PUSH   PSW      ;SAVE REGISTERS TO BE USED...
FAC7 C5                       PUSH   B
FAC8 3E 16                    MVI    A,SYNCH  ;GET SYNC CHAR
FACA 06 20                    MVI    B,NSYNC  ;GET # OF SYNC CHARS TO OUT
FACC CD 21 FB        WRTH0:   CALL   WRITB    ;WRITE SYNC CHAR
FACF 05                       DCR    B        ;COUNT DOWN
FAD0 C2 CC FA                 JNZ    WRTH0    ;LOOP UNTIL ALL OUT
FAD3 3E 02                    MVI    A,STXCH  ;GET START-OF-TEXT CHAR
FAD5 CD 21 FB                 CALL   WRITB    ;WRITE IT
FAD8 79                       MOV    A,C      ;PUT CHAR COUNT IN A
FAD9 CD 21 FB                 CALL   WRITB    ;WRITE IT
FADC C1                       POP    B        ;RESTORE SAVED REGISTERS...
FADD F1                       POP    PSW
FADE C9                       RET
```

```
FADF                    ;
FADF                    ; READ HIT STANDARD BLOCK HEADER.
FADF                    ;
FADF C5                 READH:    PUSH   B        ;SAVE REGISTER TO BE USED
FAE0 0E 00                        MVI    C,0      ;C WILL COUNT NON-SYNC'S
FAE2 CD 61 FB           REAH0:    CALL   READA    ;TRY TO READ A SYNC CHAR
FAE5 FE 16                        CPI    SYNCH    ;IS IT A SYNC?
FAE7 CA F3 FA                     JZ     REAH1    ;YES, PROCEED
FAEA 0D                           DCR    C        ;COUNT DOWN
FAEB C2 E2 FA                     JNZ    REAH0    ;LOOP IF MORE NON-SYNC'S OK
FAEE 3E 01                        MVI    A,ERR1   ;NO SYNC'S FOUND
FAF0 37                           STC             ;NOTE ERROR
FAF1 C1                           POP    B        ;PAMPER STACK
FAF2 C9                           RET             ;ABORT
FAF3 0E 20              REAH1:    MVI    C,NSYNC  ;GOT A SYNC, GET COUNT ALLOWED
FAF5 CD 69 FB           REAH2:    CALL   READB    ;READ A BYTE
FAF8 FE 16                        CPI    SYNCH    ;IS IT A SYNC?
FAFA C2 06 FB                     JNZ    REAH3    ;NO, CHECK IT OUT
FAFD 0D                           DCR    C        ;COUNT DOWN
FAFE C2 F5 FA                     JNZ    REAH2    ;LOOP IF MORE SYNC'S OK
FB01 3E 02                        MVI    A,ERR2   ;TOO MANY SYNC'S
FB03 37                           STC             ;NOTE ERROR
FB04 C1                           POP    B        ;TAKE CARE OF STACK
FB05 C9                           RET             ;ABORT
FB06 C1                 REAH3:    POP    B        ;RESTORE BUFFER LENGTH
FB07 FE 02                        CPI    STXCH    ;DOES STX FOLLOW SYNC'S?
FB09 CA 10 FB                     JZ     REAH4    ;YES, PROCEED
FB0C 3E 03                        MVI    A,ERR3   ;NO, ERROR
FB0E 37                           STC             ;NOTE ERROR
FB0F C9                           RET             ;ABORT
FB10 CD 69 FB           REAH4:    CALL   READB    ;READ LENGTH OF DATA FIELD
FB13 B9                           CMP    C        ;WILL DATA FIT IN BUFFER?
FB14 DA 1E FB                     JC     REAH5    ;YES, EASILY
FB17 CA 1F FB                     JZ     REAH6    ;YES, EXACTLY
FB1A 3E 05                        MVI    A,ERR5   ;NO, RECORD LONGER THAN BUFFER
FB1C 37                           STC             ;NOTE ERROR
FB1D C9                           RET             ;ABORT
FB1E 4F                 REAH5:    MOV    C,A      ;PUT NEW COUNT IN C
FB1F A7                 REAH6:    ANA    A        ;CLEAR CARRY
FB20 C9                           RET
FB21                    ;
FB21                    ; WRITE BYTE IN A TO TAPE.
FB21                    ;
FB21 F5                 WRITB:    PUSH   PSW      ;SAVE REGISTERS TO BE USED...
FB22 C5                           PUSH   B
FB23 D5                           PUSH   D
FB24 A7                           ANA    A        ;CLEAR CARRY (RESET STOP BIT)
FB25 06 09                        MVI    B,BPC    ;GET BIT COUNT
FB27 0E 05              BITS0:    MVI    C,ZPL    ;SET TIMER TO ZERO PULSE LENGT
FB29 1F                           RAR             ;CARRY HOLDS NEXT BIT TO OUTPUT
FB2A D2 2F FB                     JNC    ZEROB    ;SKIP IF BIT IS A ZERO
FB2D 0E 0F                        MVI    C,OPL    ;SET TIMER TO ONE PULSE LENGTH
FB2F A7                 ZEROB:    ANA    A        ;RESET CARRY TO INIT FOR HCYC0
FB30 16 18                        MVI    D,HCPBF  ;GET # HALF CYCLES PER BIT FRA
FB32                    ;
FB32                    ; THE FOLLOWING HALF CYCLE LOOP IS CRITICALLY TIMED.
FB32                    ;   SINCE THE DESIRED TRANSMISSION FREQUENCY IS 2000HZ,
FB32                    ;   A FULL CYCLE SHOULD TAKE 500US, SO EACH PASS THROUGH
FB32                    ;   THE LOOP SHOULD TAKE 250US. WITH A CLOCK CYCLE TIME
FB32                    ;   OF 500NS, THE LOOP SHOULD CONSUME 500 MACHINE CYCLES.
```

```
FB32                 ;   THE NUMBER OF MACHINE CYCLES CONSUMED BY EACH
FB32                 ;   INSTRUCTION BELOW IS NOTED.
FB32                 ;
FB32 5F              HCYC0:   MOV    E,A      ; 5. SAVE A TEMPORARILY
FB33 0D                       DCR    C        ; 4. COUNT DOWN TONE PULSES
FB34 FA 45 FB                 JM     PTSKP    ;10. SKIP PULSE TRANSMISSION IF DONE
FB37 3F                       CMC             ; 4. TOGGLE TRANSMISSION BIT
FB38 DA 40 FB                 JC     SBITS    ;10. SET BITS FOR OUTPUT IF CARRY
FB3B 3E 00                    MVI    A,0      ; 7. RESET BITS FOR OUTPUT
FB3D C3 4B FB                 JMP    SBOUT    ;10. GO TO OUTPUT BITS
FB40 3E 03           SBITS:   MVI    A,3      ; 7. SET BITS FOR OUTPUT
FB42 C3 4B FB                 JMP    SBOUT    ;10. GO TO OUTPUT BITS
FB45 0A              PTSKP:   LDAX   B        ; 7. WASTE 14 CYCLES...
FB46 02                       STAX   B        ; 7.
FB47 3E 02                    MVI    A,2      ; 7. SET ONE BIT ON AND ONE OFF
FB49 7F                       MOV    A,A      ; 5. WASTE 10 CYCLES...
FB4A 7F                       MOV    A,A      ; 5.
FB4B D3 FC           SBOUT:   OUT    UCRIP    ;10. OUTPUT BITS
FB4D 3E 1B                    MVI    A,27     ;10. GET COUNT FOR LOOP
FB4F 3D              WAIT0:   DCR    A        ; 5. COUNT DOWN
FB50 C2 4F FB                 JNZ    WAIT0    ;10. LOOP 27 TIMES
FB53 7B                       MOV    A,E      ; 5. RESTORE ORIGINAL A
FB54 7F                       MOV    A,A      ; 5. WASTE 5 MORE CYCLES
FB55 15                       DCR    D        ; 5. COUNT DOWN HALF CYCLES
FB56 C2 32 FB                 JNZ    HCYC0    ;10. KEEP LOOPING TILL FRAME DONE
FB59                 ;
FB59 05                       DCR    B        ;COUNT DOWN BITS
FB5A C2 27 FB                 JNZ    BITS0    ;OUTPUT NEXT BIT, UNLESS DONE
FB5D D1                       POP    D        ;RESTORE SAVED REGISTERS AND
FB5E C1                       POP    B        ;  RETURN...
FB5F F1                       POP    PSW
FB60 C9                       RET
FB61                 ;
FB61                 ; READ BYTE FROM TAPE INTO A.
FB61                 ;
FB61 D5              READA:   PUSH   D        ;SAVE REGISTER
FB62 16 00                    MVI    D,0      ;SCAN UP TO 256 BITS FOR A SYNC
FB64 1E 00                    MVI    E,0      ;LOOKING FOR A SYNC FLAG
FB66 C3 6E FB                 JMP    RESET    ;START COMMON CODE
FB69 D5              READB:   PUSH   D        ;SAVE REGISTER
FB6A 16 08                    MVI    D,DBPC   ;GET # OF DATA BITS PER CHAR
FB6C 1E 01                    MVI    E,1      ;NOT LOOKING FOR A SYNC FLAG
FB6E C5              RESET:   PUSH   B        ;SAVE REGISTER
FB6F AF                       XRA    A        ;RESET A & CARRY
FB70 47                       MOV    B,A      ;INITIALIZE BYTE HOLDER
FB71 DB FC           TRAPL:   IN     UCRIP    ;WAIT FOR TRAILING PULSE...
FB73 1F                       RAR
FB74 DA 71 FB                 JC     TRAPL
FB77 0E 00           GBIT0:   MVI    C,0      ;INITIALIZE PULSE LENGTH COUNT
FB79 DB FC           LEDGE:   IN     UCRIP    ;WAIT FOR LEADING EDGE OF SIGNAL
FB7B 1F                       RAR
FB7C D2 79 FB                 JNC    LEDGE
FB7F 0C              TIMPL:   INR    C        ;BUMP PULSE LENGTH COUNT
FB80 C3 83 FB                 JMP    $        ;(SYMMETRICAL WITH NEXT LOOP)
FB83 DB FC                    IN     UCRIP    ;SIGNAL STILL PRESENT?...
FB85 1F                       RAR
FB86 DA 7F FB                 JC     TIMPL    ;LOOP UNTIL SIGNAL OFF
FB89 0D              TIMPN:   DCR    C        ;COUNT BACK DOWN
FB8A CA 93 FB                 JZ     BITSD    ;BIT STATE DETERMINED IF 0
FB8D DB FC                    IN     UCRIP    ;SIGNAL STILL OFF?...
```

```
FB8F 1F                  RAR
FB90 D2 89 FB            JNC     TIMPN   ;LOOP UNTIL NEXT SIGNAL
FB93 78         BITSD:   MOV     A,B     ;BIT STATE DETERMINED, GET BYTE
FB94 1F                  RAR             ;INSERT NEW BIT
FB95 47                  MOV     B,A     ;SAVE BYTE AGAIN
FB96 D6 16               SUI     SYNCH   ;IS IT A SYNC?
FB98 B3                  ORA     E       ;LOOKING FOR A SYNC?
FB99 CA A0 FB            JZ      STOPL   ;WAIT FOR STOP BIT IF YES
FB9C 15                  DCR     D       ;COUNT DOWN BITS
FB9D C2 77 FB            JNZ     GBIT0   ;LOOP UNTIL ALL BITS READ
FBA0 DB FC      STOPL:   IN      UCRIP   ;WAIT FOR STOP BIT PULSE...
FBA2 1F                  RAR
FBA3 D2 A0 FB            JNC     STOPL
FBA6 78                  MOV     A,B     ;PUT BYTE IN A
FBA7 C1                  POP     B       ;RESTORE SAVED REGISTERS...
FBA8 D1                  POP     D
FBA9 C9                  RET
FBAA            ;
FBAA            ; WRITE HIT STANDARD BLOCK TRAILER.
FBAA            ;
FBAA 3E 03      WRITT:   MVI     A,ETXCH ;GET END-OF-TEXT CHAR
FBAC CD 21 FB            CALL    WRITB   ;WRITE IT TO TAPE
FBAF 7A                  MOV     A,D     ;GET HIGH BYTE OF CHECKSUM
FBB0 CD 21 FB            CALL    WRITB   ;WRITE IT
FBB3 7B                  MOV     A,E     ;GET LOW BYTE OF CHECKSUM
FBB4 CD 21 FB            CALL    WRITB   ;WRITE IT
FBB7 C9                  RET
FBB8            ;
FBB8            ; READ HIT STANDARD BLOCK TRAILER.
FBB8            ;
FBB8 CD 69 FB   READT:   CALL    READB   ;READ A BYTE FROM TAPE
FBBB FE 03               CPI     ETXCH   ;IS IT AN ETX?
FBBD CA C4 FB            JZ      READX   ;CONTINUE IF IT IS
FBC0 3E 04               MVI     A,ERR4  ;NO ETX FOUND
FBC2 37                  STC             ;NOTE ERROR
FBC3 C9                  RET             ;ABORT
FBC4 CD 69 FB   READX:   CALL    READB   ;READ NEXT BYTE
FBC7 BA                  CMP     D       ;IS IT HIGH CHECKSUM BYTE?
FBC8 CA D3 FB            JZ      READY   ;CONTINUE IF IT IS
FBCB B7                  ORA     A       ;IS IT ZERO?
FBCC CA D3 FB            JZ      READY   ;CONTINUE IF IT IS
FBCF 3E 06               MVI     A,ERR6  ;CHECKSUM ERROR
FBD1 37                  STC             ;NOTE ERROR
FBD2 C9                  RET             ;ABORT
FBD3 CD 69 FB   READY:   CALL    READB   ;READ NEXT BYTE
FBD6 BB                  CMP     E       ;IS IT LOW CHECKSUM BYTE?
FBD7 CA E2 FB            JZ      READZ   ;CONTINUE IF IT IS
FBDA B7                  ORA     A       ;IS IT ZERO?
FBDB CA E2 FB            JZ      READZ   ;CONTINUE IF IT IS
FBDE 3E 06               MVI     A,ERR6  ;CHECKSUM ERROR
FBE0 37                  STC             ;NOTE ERROR
FBE1 C9                  RET             ;ABORT
FBE2 A7         READZ:   ANA     A       ;CLEAR CARRY
FBE3 C9                  RET             ;NORMAL RETURN
FBE4            ;
FBE4            ; TEST WRITE ROUTINE. WRITE INIT CHARS UNTIL STOPPED.
FBE4            ;
FBE4 CD 5E FA   TESTW:   CALL    INIT    ;WRITE A NUMBER OF INIT CHARS
FBE7 C3 E4 FB            JMP     TESTW   ;LOOP INDEFINITELY
FBEA            ;
```

```
FBEA                ; TEST READ ROUTINE. READ BYTES FROM TAPE AND DISPLAY
FBEA                ;    THEM UNTIL STOPPED. THE USER SHOULD ADJUST VOLUME
FBEA                ;    UNTIL THE LIGHTS DON'T FLICKER.
FBEA                ;
FBEA CD 69 FB       TESTR:      CALL    READB    ;READ A BYTE FROM TAPE
FBED 2F                         CMA              ;COMPLEMENT IT FOR OUTPUT
FBEE D3 FF                      OUT     FRONP    ;DISPLAY IT
FBF0 C3 EA FB                   JMP     TESTR    ;LOOP INDEFINITELY
```

```
; ******** UCRI CASSETTE INTERFACE DRIVER ********
;
; REVISION 1        27 JUL 76        BRH
;
                ORG     0FA00H
;
; I/O PORTS
;
UCRIP           EQU     0FCH       ;UCRI I/O PORT
FRONP           EQU     0FFH       ;FRONT PANEL I/O PORT
;
; TAPE INITIALIZATION PARAMS (WHEN WRITING DUMMY
;    CHARS TO STABILIZE AUTOMATIC VOLUME CONTROL)
;
INCHR           EQU     033H       ;DUMMY INITIALIZATION CHAR
INCNT           EQU     0FFH       ;# OF INCHR'S TO WRITE
;
; MISCELLANEOUS PARAMETERS
;
BPC             EQU     9          ;# OF BITS PER CHARACTER
DBPC            EQU     8          ;# OF DATA BITS PER CHARACTER
ZPL             EQU     5          ;# OF HALF CYCLES IN ZERO PULSE
OPL             EQU     15         ;# OF HALF CYCLES IN ONE PULSE
HCPBF           EQU     24         ;# OF HALF CYCLES PER BIT FRAME
NSYNC           EQU     32         ;# OF SYNC'S STARTING A RECORD
;
; SPECIAL CHARACTERS
;
SYNCH           EQU     16H
STXCH           EQU     02H
ETXCH           EQU     03H
;
; ERROR CODES
;
ERR1            EQU     1          ;NO INITIAL SYNC CHARS FOUND
ERR2            EQU     2          ;MORE THAN 32 SYNC CHARS FOUND
ERR3            EQU     3          ;NO STX CHAR FOUND
ERR4            EQU     4          ;NO ETX CHAR FOUND
ERR5            EQU     5          ;RECORD LONGER THAN BUFFER
ERR6            EQU     6          ;CHECKSUM ERROR
;
; ENTRY POINTS (THIS DRIVER IS NON-STANDARD).
;
                JMP     CRIOD      ;ORIGINAL DRIVER ENTRY POINT
                JMP     DUMP       ;DUMP MEMORY TO TAPE
                JMP     LOAD       ;LOAD MEMORY FROM TAPE
                JMP     TESTW      ;TEST WRITE ROUTINE
                JMP     TESTR      ;TEST READ ROUTINE
;
; PICK UP PROPER STATUS BYTE FOR DUMP/LOAD.
;
DUMP:           MVI     B 80H      ;BLOCKED WRITE STATUS BYTE
                JMP     START      ;START COMMON CODE
LOAD:           MVI     B,81H      ;BLOCKED READ STATUS BYTE
;
; INITIALIZE FOR WRITE/READ RECORD LOOP.
;
START:          MVI     C,128      ;LENGTH OF EACH RECORD
                IN      FRONP      ;READ SWITCHES
```

```
                MOV     D,A      ;SAVE TEMPORARILY
                ANI     0F0H     ;ZERO LOW NIBBLE
                MOV     E,A      ;SAVE HIGH BYTE OF 1ST ADDR
                MOV     A,D      ;GET SWITCHES BACK
                SUB     E        ;ZERO HIGH NIBBLE
                JZ      EXIT     ;EXIT IF COUNT IS ZERO
                MOV     D,A      ;SAVE COUNT OF 4K INC'S
                RLC              ;SWAP NIBBLES...
                RLC
                RLC
                RLC
                ADD     E        ;COMPUTE AND SAVE HIGH BYTE
                MOV     H,A      ;  OF RAM ADDRESS
                MVI     L,80     ;SET SP HIGH ENOUGH TO LEAVE
                SPHL             ;  ROOM FOR DESCENDING STACK
                MOV     H,E      ;POINT HL AT 1ST MEMORY ADDR
                MVI     L,0      ;  TO BE DUMPED/LOADED
                MOV     A,B      ;PUT STATUS BYTE IN A
                RAR              ;PUT WRITE/READ BIT IN CARRY
                CNC     INIT     ;IF WRITING, INIT AUTO VOL CTRL
;
; LOOPS TO:
;    1) DUMP/LOAD 4K INC'S UNTIL COUNT EXHAUSTED, AND
;    2) WRITE/READ 32 RECORDS.
;
LOOP1:          MVI     E,32     ;# OF 128-BYTE BLOCKS IN 4K
LOOP2:          MOV     A,H      ;PUT HIGH ADDRESS BYTE IN A
                CMA              ;COMPLEMENT FOR OUTPUT
                OUT     FRONP    ;SET LIGHTS
                CALL    CRIOD    ;WRITE/READ A 128-BYTE RECORD
                JC      EXIT     ;EXIT IF ERROR
                DCR     E        ;COUNT RECORDS DOWN
                JNZ     LOOP2    ;LOOP UNTIL ALL RECORDS DONE
                DCR     D        ;COUNT 4K INC'S DOWN
                JNZ     LOOP1    ;LOOP UNTIL DONE
                XRA     A        ;DONE, SET LIGHTS OFF
;
; EXIT ROUTINE. FLASH COMPLETION CODE IN A.
;
EXIT:           CMA              ;COMPLEMENT FOR OUTPUT
                MOV     D,A      ;SAVE OUTPUT CODE IN D
                MVI     H,0FFH   ;PUT 00 OUT CODE IN H
OUT:            MOV     A,H      ;GET CODE TO OUTPUT
                OUT     FRONP    ;SET LIGHTS
                XCHG             ;EXCHANGE CODES IN D & H
                LXI     B,4000H  ;GET DELAY COUNTER
DELAY:          DCX     B        ;START OF DELAY LOOP
                MOV     A,B      ;IS COUNT ZERO?...
                ORA     C
                JNZ     DELAY    ;LOOP UNTIL IT IS
                JMP     OUT      ;OUTPUT OTHER CODE
;
; IN ORDER TO LET THE AUTOMATIC VOLUME CONTROL
;    STABILIZE, WRITE A NUMBER OF DUMMY INIT CHARS.
;
INIT:           PUSH    PSW      ;SAVE REGISTERS TO BE USED...
                PUSH    B
                MVI     C,INCNT  ;GET INIT CHAR COUNT
                MVI     A,INCHR  ;GET INIT CHAR
INIT0:          CALL    WRITB    ;WRITE IT
```

```
1190            DCR     C           ;COUNT DOWN
1200            JNZ     INIT0       ;LOOP UNTIL COUNT EXHAUSTED
1210            POP     B           ;RESTORE SAVED REGISTERS...
1220            POP     PSW
1230            RET
1240 ;
1250 ; HIT STANDARD CASSETTE RECORDER I/O DRIVER.
1260 ;
1270 CRIOD:     DI                  ;INSURE ACCURATE TIMING
1280            PUSH    B           ;SAVE REGISTERS TO BE USED...
1290            PUSH    D
1300            MOV     A,B         ;PUT STATUS BYTE IN A
1310            RRC                 ;PUT WRITE/READ BIT IN CARRY
1320            JC      CREAD       ;JUMP IF READING
1330            MOV     A,B         ;PUT STATUS BYTE IN A
1340            RLC                 ;PUT BLOCKED BIT IN CARRY
1350            CC      WRITH       ;WRITE HEADER IF BLOCKING
1360            LXI     D,0FFH      ;INITIALIZE CHECKSUM
1370            INR     C           ;PRIME COUNT FOR LOOP
1380 WLOOP:     DCR     C           ;COUNT DOWN
1390            JZ      WTAIL       ;EXIT LOOP IF NO MORE DATA
1400            MOV     A,M         ;PUT NEXT BYTE IN A
1410            INX     H           ;BUMP POINTER
1420            CALL    CKSUM       ;ADD BYTE TO CHECKSUM
1430            CALL    WRITB       ;WRITE BYTE
1440            JMP     WLOOP       ;LOOP UNTIL ALL DATA OUT
1450 WTAIL:     MOV     A,B         ;PUT STATUS BYTE IN A
1460            RLC                 ;PUT BLOCKED BIT IN CARRY
1470            CC      WRITT       ;WRITE TAIL IF BLOCKING
1480            JMP     CDONE       ;ALL DONE, RETURN
1490 CREAD:     MOV     A,B         ;PUT STATUS BYTE IN A
1500            RLC                 ;PUT BLOCKED BIT IN CARRY
1510            CC      READH       ;READ HEADER IF BLOCKED DATA
1520            JC      ERROR       ;EXIT IF ERROR
1530            LXI     D,0FFH      ;INITIALIZE CHECKSUM
1540            INR     C           ;PRIME COUNT FOR LOOP
1550 RLOOP:     DCR     C           ;COUNT DOWN
1560            JZ      RTAIL       ;EXIT LOOP IF NO MORE DATA
1570            CALL    READB       ;READ A BYTE
1580            MOV     M,A         ;STORE IT IN BUFFER
1590            INX     H           ;BUMP POINTER
1600            CALL    CKSUM       ;ADD BYTE TO CHECKSUM
1610            JMP     RLOOP
1620 RTAIL:     MOV     A,B         ;PUT STATUS BYTE IN A
1630            RLC                 ;PUT BLOCKED BIT IN CARRY
1640            CC      READT       ;READ TAIL IF BLOCKED DATA
1650            JC      ERROR       ;NOTE ERROR
1660 CDONE:     XRA     A           ;NORMAL RETURN, RESET FLAGS
1670 ERROR:     POP     D           ;RESTORE SAVED REGISTERS...
1680            POP     B
1690            RET
1700 ;
1710 ; ADD BYTE IN A TO CHECKSUM IN DE.
1720 ;
1730 CKSUM:     PUSH    PSW         ;SAVE A
1740            ADD     E           ;ADD TO LOW BYTE
1750            MOV     E,A         ;STORE NEW LOW BYTE
1760            MVI     A,0         ;RESET A
1770            ADC     D           ;ADD CARRY TO HIGH BYTE
1780            MOV     D,A         ;STORE NEW HIGH BYTE
```

```
                POP     PSW     ;RESTORE A
                RET
;
; WRITE HIT STANDARD BLOCK HEADER.
;
WRITH:          PUSH    PSW     ;SAVE REGISTERS TO BE USED...
                PUSH    B
                MVI     A,SYNCH ;GET SYNC CHAR
                MVI     B,NSYNC ;GET # OF SYNC CHARS TO OUT
WRTH0:          CALL    WRITB   ;WRITE SYNC CHAR
                DCR     B       ;COUNT DOWN
                JNZ     WRTH0   ;LOOP UNTIL ALL OUT
                MVI     A,STXCH ;GET START-OF-TEXT CHAR
                CALL    WRITB   ;WRITE IT
                MOV     A,C     ;PUT CHAR COUNT IN A
                CALL    WRITB   ;WRITE IT
                POP     B       ;RESTORE SAVED REGISTERS...
                POP     PSW
                RET
;
; READ HIT STANDARD BLOCK HEADER.
;
READH:          PUSH    B       ;SAVE REGISTER TO BE USED
                MVI     C,0     ;C WILL COUNT NON-SYNC'S
REAH0:          CALL    READA   ;TRY TO READ A SYNC CHAR
                CPI     SYNCH   ;IS IT A SYNC?
                JZ      REAH1   ;YES, PROCEED
                DCR     C       ;COUNT DOWN
                JNZ     REAH0   ;LOOP IF MORE NON-SYNC'S OK
                MVI     A,ERR1  ;NO SYNC'S FOUND
                STC             ;NOTE ERROR
                POP     B       ;PAMPER STACK
                RET             ;ABORT
REAH1:          MVI     C,NSYNC ;GOT A SYNC, GET COUNT ALLOWED
REAH2:          CALL    READB   ;READ A BYTE
                CPI     SYNCH   ;IS IT A SYNC?
                JNZ     REAH3   ;NO, CHECK IT OUT
                DCR     C       ;COUNT DOWN
                JNZ     REAH2   ;LOOP IF MORE SYNC'S OK
                MVI     A,ERR2  ;TOO MANY SYNC'S
                STC             ;NOTE ERROR
                POP     B       ;TAKE CARE OF STACK
                RET             ;ABORT
REAH3:          POP     B       ;RESTORE BUFFER LENGTH
                CPI     STXCH   ;DOES STX FOLLOW SYNC'S?
                JZ      REAH4   ;YES, PROCEED
                MVI     A,ERR3  ;NO, ERROR
                STC             ;NOTE ERROR
                RET             ;ABORT
REAH4:          CALL    READB   ;READ LENGTH OF DATA FIELD
                CMP     C       ;WILL DATA FIT IN BUFFER?
                JC      REAH5   ;YES, EASILY
                JZ      REAH6   ;YES, EXACTLY
                MVI     A,ERR5  ;NO, RECORD LONGER THAN BUFFER
                STC             ;NOTE ERROR
                RET             ;ABORT
REAH5:          MOV     C,A     ;PUT NEW COUNT IN C
REAH6:          ANA     A       ;CLEAR CARRY
                RET
;
```

```
; WRITE BYTE IN A TO TAPE.
;
WRITB:      PUSH    PSW      ;SAVE REGISTERS TO BE USED...
            PUSH    B
            PUSH    D
            ANA     A        ;CLEAR CARRY (RESET STOP BIT)
            MVI     B,BPC    ;GET BIT COUNT
BITS0:      MVI     C,ZPL    ;SET TIMER TO ZERO PULSE LENGTH
            RAR              ;CARRY HOLDS NEXT BIT TO OUTPUT
            JNC     ZEROB    ;SKIP IF BIT IS A ZERO
            MVI     C,OPL    ;SET TIMER TO ONE PULSE LENGTH
ZEROB:      ANA     A        ;RESET CARRY TO INIT FOR HCYC0
            MVI     D,HCPBF  ;GET # HALF CYCLES PER BIT FRAME
;
; THE FOLLOWING HALF CYCLE LOOP IS CRITICALLY TIMED.
;   SINCE THE DESIRED TRANSMISSION FREQUENCY IS 2000HZ,
;   A FULL CYCLE SHOULD TAKE 500US, SO EACH PASS THROUGH
;   THE LOOP SHOULD TAKE 250US. WITH A CLOCK CYCLE TIME
;   OF 500NS, THE LOOP SHOULD CONSUME 500 MACHINE CYCLES.
;   THE NUMBER OF MACHINE CYCLES CONSUMED BY EACH
;   INSTRUCTION BELOW IS NOTED.
;
HCYC0:      MOV     E,A      ; 5. SAVE A TEMPORARILY
            DCR     C        ; 4. COUNT DOWN TONE PULSES
            JM      PTSKP    ;10. SKIP PULSE TRANSMISSION IF DONE
            CMC              ; 4. TOGGLE TRANSMISSION BIT
            JC      SBITS    ;10. SET BITS FOR OUTPUT IF CARRY
            MVI     A,0      ; 7. RESET BITS FOR OUTPUT
            JMP     SBOUT    ;10. GO TO OUTPUT BITS
SBITS:      MVI     A,3      ; 7. SET BITS FOR OUTPUT
            JMP     SBOUT    ;10. GO TO OUTPUT BITS
PTSKP:      LDAX    B        ; 7. WASTE 14 CYCLES...
            STAX    B        ; 7.
            MVI     A,2      ; 7. SET ONE BIT ON AND ONE OFF
            MOV     A,A      ; 5. WASTE 10 CYCLES...
            MOV     A,A      ; 5.
SBOUT:      OUT     UCRIP    ;10. OUTPUT BITS
            MVI     A,27     ;10. GET COUNT FOR LOOP
WAIT0:      DCR     A        ; 5. COUNT DOWN
            JNZ     WAIT0    ;10. LOOP 27 TIMES
            MOV     A,E      ; 5. RESTORE ORIGINAL A
            MOV     A,A      ; 5. WASTE 5 MORE CYCLES
            DCR     D        ; 5. COUNT DOWN HALF CYCLES
            JNZ     HCYC0    ;10. KEEP LOOPING TILL FRAME DONE
;
            DCR     B        ;COUNT DOWN BITS
            JNZ     BITS0    ;OUTPUT NEXT BIT, UNLESS DONE
            POP     D        ;RESTORE SAVED REGISTERS AND
            POP     B        ;  RETURN...
            POP     PSW
            RET
;
; READ BYTE FROM TAPE INTO A.
;
READA:      PUSH    D        ;SAVE REGISTER
            MVI     D,0      ;SCAN UP TO 256 BITS FOR A SYNC
            MVI     E,0      ;LOOKING FOR A SYNC FLAG
            JMP     RESET    ;START COMMON CODE
READB:      PUSH    D        ;SAVE REGISTER
            MVI     D,DBPC   ;GET # OF DATA BITS PER CHAR
```

```
               MVI     E,1      ;NOT LOOKING FOR A SYNC FLAG
     RESET:    PUSH    B        ;SAVE REGISTER
               XRA     A        ;RESET A & CARRY
               MOV     B,A      ;INITIALIZE BYTE HOLDER
     TRAPL:    IN      UCRIP    ;WAIT FOR TRAILING PULSE...
               RAR
               JC      TRAPL
     GBIT0:    MVI     C,0      ;INITIALIZE PULSE LENGTH COUNT
     LEDGE:    IN      UCRIP    ;WAIT FOR LEADING EDGE OF SIGNAL
               RAR
               JNC     LEDGE
     TIMPL:    INR     C        ;BUMP PULSE LENGTH COUNT
               JMP     $        ;(SYMMETRICAL WITH NEXT LOOP)
               IN      UCRIP    ;SIGNAL STILL PRESENT?...
               RAR
               JC      TIMPL    ;LOOP UNTIL SIGNAL OFF
     TIMPN:    DCR     C        ;COUNT BACK DOWN
               JZ      BITSD    ;BIT STATE DETERMINED IF 0
               IN      UCRIP    ;SIGNAL STILL OFF?...
               RAR
               JNC     TIMPN    ;LOOP UNTIL NEXT SIGNAL
     BITSD:    MOV     A,B      ;BIT STATE DETERMINED, GET BYTE
               RAR              ;INSERT NEW BIT
               MOV     B,A      ;SAVE BYTE AGAIN
               SUI     SYNCH    ;IS IT A SYNC?
               ORA     E        ;LOOKING FOR A SYNC?
               JZ      STOPL    ;WAIT FOR STOP BIT IF YES
               DCR     D        ;COUNT DOWN BITS
               JNZ     GBIT0    ;LOOP UNTIL ALL BITS READ
     STOPL:    IN      UCRIP    ;WAIT FOR STOP BIT PULSE...
               RAR
               JNC     STOPL
               MOV     A,B      ;PUT BYTE IN A
               POP     B        ;RESTORE SAVED REGISTERS...
               POP     D
               RET
     ;
     ; WRITE HIT STANDARD BLOCK TRAILER.
     ;
     WRITT:    MVI     A,ETXCH  ;GET END-OF-TEXT CHAR
               CALL    WRITB    ;WRITE IT TO TAPE
               MOV     A,D      ;GET HIGH BYTE OF CHECKSUM
               CALL    WRITB    ;WRITE IT
               MOV     A,E      ;GET LOW BYTE OF CHECKSUM
               CALL    WRITB    ;WRITE IT
               RET
     ;
     ; READ HIT STANDARD BLOCK TRAILER.
     ;
     READT:    CALL    READB    ;READ A BYTE FROM TAPE
               CPI     ETXCH    ;IS IT AN ETX?
               JZ      READX    ;CONTINUE IF IT IS
               MVI     A,ERR4   ;NO ETX FOUND
               STC              ;NOTE ERROR
               RET              ;ABORT
     READX:    CALL    READB    ;READ NEXT BYTE
               CMP     D        ;IS IT HIGH CHECKSUM BYTE?
               JZ      READY    ;CONTINUE IF IT IS
               ORA     A        ;IS IT ZERO?
               JZ      READY    ;CONTINUE IF IT IS
```

```
3590                MVI      A,ERR6   ;CHECKSUM ERROR
3600                STC               ;NOTE ERROR
3610                RET               ;ABORT
3620 READY:         CALL     READB    ;READ NEXT BYTE
3630                CMP      E        ;IS IT LOW CHECKSUM BYTE?
3640                JZ       READZ    ;CONTINUE IF IT IS
3650                ORA      A        ;IS IT ZERO?
3660                JZ       READZ    ;CONTINUE IF IT IS
3670                MVI      A,ERR6   ;CHECKSUM ERROR
3680                STC               ;NOTE ERROR
3690                RET               ;ABORT
3700 READZ:         ANA      A        ;CLEAR CARRY
3710                RET               ;NORMAL RETURN
3720 ;
3730 ; TEST WRITE ROUTINE. WRITE INIT CHARS UNTIL STOPPED.
3740 ;
3750 TESTW:         CALL     INIT     ;WRITE A NUMBER OF INIT CHARS
3760                JMP      TESTW    ;LOOP INDEFINITELY
3770 ;
3780 ; TEST READ ROUTINE. READ BYTES FROM TAPE AND DISPLAY
3790 ;    THEM UNTIL STOPPED. THE USER SHOULD ADJUST VOLUME
3800 ;    UNTIL THE LIGHTS DON'T FLICKER.
3810 ;
3820 TESTR:         CALL     READB    ;READ A BYTE FROM TAPE
3830                CMA               ;COMPLEMENT IT FOR OUTPUT
3840                OUT      FPONP    ;DISPLAY IT
3850                JMP      TESTR    ;LOOP INDEFINITELY
```

ERRATA SHEET
UCRI-1 SOFTWARE

The arrows on the object tape of the UCRI-1 driver point in the wrong direction, so the tape must be loaded backwards.

IMSAI

MIO

TABLE OF CONTENTS

MIO

FUNCTIONAL DESCRIPTION

INPUT/OUTPUT VERSATILITY

The MIO, Multiple Input Output Board, is designed to meet all Input/Output requirements of most 8080 System Users by providing the User with the following Input/Output interfaces:

1. one Data Storage interface to a standard audio cassette recorder;
2. two Parallel Input/Output (PIO) ports;
3. one Serial Input/Output port; and
4. one control port to be used for internal and external control functions.

As an example of its versatility, a single MIO Board could control a TV Typewriter, a Line Printer, a Teletype, and a cassette recorder.

SOFTWARE COMPATIBILITY

Board Addressing and Port Configuration capabilities allow the MIO Board to be Address Compatible with virtually all Software Packages.

The Board is jumper selectable to any one of the 64 groups of 4 Input/Output addresses available with the 8080. Jumper selection further allows each port to be configured in any order within the selected group of 4 addresses.

As an example, a TV Typewriter, which is a parallel I/O device, may be used with serial I/O software simply by configuring the MIO Board so that the parallel port for the TV Typewriter appears at the I/O address where the serial data normally appears.

EXTERNAL CONNECTIONS

External Interface Connections are made from the three 26-pin edge connectors at the top of the board. These contain the signals necessary for two identical parallel interfaces, and a serial I/O interface. The Current Loop or EIA options are normally configured to provide a standard EIA Data Transmission pinout at the connector.

INTERRUPT CAPABILITIES

Any of the Status Signals from each of the I/O Ports may be used to generate Interupts. Provision is made for jumpering these Status Signals to Vectored Interrupt Lines, if a PIC-8 Board is present. They may be directly jumpered to the CPU Interrupt Line for a single level Interrupt System.

SERIAL INPUT/OUTPUT PORT

The MIO Board provides for one complete Serial I/O port which is designed to require no initialization on power-up.

BOARD OPTIONS

A number of options are available and are easily selected by the User.

1. The Baud Rate is jumper selectable and can range from 45.5 to 9600 Baud.

2. Character Length, Parity Enable, and Even/Odd Parity selection are jumper selectable.

3. The Data output of the UART may be jumpered to an EIA Driver, a Current Loop Driver, or a TTL Driver.

   Similarly, the Data input of the UART may be jumpered to an EIA Receiver, a Current Loop Receiver, or a TTL Receiver.

4. Provision is made to monitor any of the UART Status Signals using the Control Input Port, or the interrupt inputs,

STATUS SIGNALS
The SIO Status Signals provided are as follows: TRANSMIT READY, the negation of TRANSMIT READY, RECEIVE READY, the negation of RECEIVE READY, PARITY ERROR, OVERRUN ERROR, and FRAMING ERROR.

An additional Status Signal, SIOS, is provided to assist in error checking routines. This signal simply indicates that one of three error conditions has occured,(PE,FE,or OE). It may be decoded via the Control Port to determine which of the three signals is active. This feature is provided to allow efficient use of the Control Port in a case where the complete board configuration is being used.

EXTERNAL INTERFACE CONNECTIONS
The SIO Port has available at a 26 pin edge connector, all signals necessary for Standard EIA, Current Loop and TTL Serial Interfaces.

PARALLEL INPUT/OUTPUT PORTS

The MIO Board provides for 2 identical 8 bit parallel input/output ports.

BOARD OPTIONS
Board options allow the User to:

1. Use one of four types of Input Strobes: 1.positive edge, 2. negative

edge, 3. positive level, and 4. negative level. It is also possible to continuously gate data into the latch.

2. Use PIO Status Signals to generate Interrupts or to be simply monitored by the Program via the Control Port.

STATUS SIGNALS

The PIO Status Signals which are provided are as follows:

ODR- one Output Data Ready line for each Parallel Output Port;

IDA- one Input Data Accepted line for each Parallel Input Port.

As with the SIO Port, an additional signal,PIOS, is provided to enhance the efficiency of the Control Port Input Bits.

EXTERNAL INTERFACE CONNECTIONS

The External Interface Connections for the PIO Output Ports provide for 8 Output Data Lines and 3 Control/Handshake Lines.

Each Input Port provides for 8 Input Data Lines and 2 Control/Handshake Lines.

All signals are available at two identical 26 pin edge connectors for easy interfacing to external parallel I/O devices.

CASSETTE INPUT/OUTPUT PORT

The MIO Board provides for one complete Cassette Recorder Interface.

BOARD OPTIONS

Board Options allow the User to:

1. Vary the recording rate from 500 to 62,500 bits per second.

2. Set the phase of the recorded signal to provide compatibility with most all audio cassette recorders.

The CRI Port writes Biphase Encoded Data to the tape. This can be used to generate Byte/Lancaster or Tarbell data formats.

The Biphase encoding generates Byte/Lancaster data formats by sending alternating 1's and 0's when a zero bit is to be recorded. It sends all 1's when a one bit is to be recorded. In this standard, the maximum data rate is 30 bytes per second.

The CRI can also operate in the Tarbell Standard, using one bit of phase encoding per data bit. This standard allows the User to record data at the standard rate of 187 bytes per second or faster if the recorder used is of sufficient quality.

The recorder section can have two cassette recorders connected to it at one time, thus providing the User with the basic capability for a cassette operating system.

MIO SPECIFICATIONS

Basic Configuration

1. The MIO board uses four I/O ports and is available with the following I/O interfaces:

   Two parallel (PIO) ports
   One control (CTL) port
   One cassette recorder (CRI) port
   One serial (SIO) port

2. There are three 26 pin edge connectors on the top of the board, two for the PIO ports and one for the SIO port. The SIO pin assignments are compatible with the standard EIA connectors. The PIO input pin numbers are the same as the PIO 4 port 0 input pin numbers, and the PIO output pin numbers are the same as the PIO 4 port 1 output pin numbers.

3. The board address (one of the 64 possible groups of four I/O ports) and the order of the addressing of the four ports on the board are jumper-selectable.

4. Interrupt requests are jumper-selectable to PIC-8 and CPU lines.

5. The operation of the individual ports is as follows:

   A) SIO
      1. Baud rate is jumper-selectable for rates of 45.5 to 9600 baud.
      2. Character length, parity enable, and even/odd parity select are jumper-selectable.
      3. Transmitted serial data is available in CTL output jumper area.
      4. Received serial data is available in the CTL input jumper area.
      5. Transmit ready (TRDY), receive ready (RRDY), parity error (PE), overrun error (OE), framing error (FE), the complements of TRDY and RRDY, and (SIOS), which can represent one of (OE), (PE), (FE) or the logical OR of the three, are all available in the CTL input jumper area.

   B) PIO
      1. Output data is latched and available at the PIO connector.
         Output Data Ready (ODR) is available at the PIO connector.
         Output Data Accepted (ODA) is available at CTL IJA.

Data Ready (DR) is available at CRL IJA.
Clear Data Ready (CDR) is available at the PIO connector.
Data Strobe (STB) is available at the PIO connector.

2. Input data is accepted from the PIO connector.
Input Data Strobe (IDS) is jumper-selectable for positive or negative edge triggering, gating or disable.
Input Data Accepted (IDA) is available at the PIO connector.
Input Data Ready is available at CTL IJA.

C) CTL

1. Bits 0-3 are latched and available in OJA.
Bit 4 = write enable for CRI
Bit 5 = read enable for CRI
Bits 6 and 7 are used to control the generation of SIOS, PIOS, CRIS and PIO port selection.

2. Output Jumper Area (OJA) has CTL bits 0-3 and SIO transmit as inputs and has the following possible outputs:

   Four EIA drivers
   One current loop driver (20 or 60 milliamp)
   Two TTL drivers
   Three open collector 20 milliamp, 40 volt drivers

3. Input Jumper Area (IJA) has output jumper positions to eight data input lines, eight interrupt request lines and the serial data input, and has as input the SIO, PIO and CRI status signals, as well as, four EIA and one current loop receiver.

D) CRI

The CRI is capable of writing or reading biphase encoded data at rates of 500 to 62,500 bits per second. It can operate in either the "Byte/Lancaster" or "Tarbell" recording standards. (Note: the standard rates for "Byte/Lancaster" and "Tarbell" operation are 2400 bps and 1500 bps, respectively.) The CRI can interrupt on a bit byte basis. It has two input and two output connections for cassette interface, although only one input may be operating at a given time.

THEORY OF OPERATION

The MIO, Multiple Input/Output Board, contains all the logic required to implement two latched parallel input/output (PIO) ports, a serial I/O (SIO) port, a cassette recorder interface (CRI) port and a port for the control of the other ports or external devices. The Theory of operation will be discussed by first describing the internal data bussing of the board and then discussing each of the individual types of I/O ports. The reader should be completely familiar with the MIO User Guide prior to reading the Theory of Operation.

Internal Data Bussing

The MIO board has an internal bi-directional, 8-bit data bus. The output information from the 8080 back panel is gated onto the internal bus whenever SOUT is asserted. When the MIO is selected and PDBIN is asserted, data is gated from the internal bus to the 8080 back panel bus. The gating is done with 74367's to increase the current sink capability to 32 milliamps per line. Each of the individual ports on the internal bus has its own 3-state driver. All of the ports except the control port have this driver as an integral part of the latches holding the information for these ports. The control input port uses a separate 74367 to gate the data onto the internal bus.

Interrupt Generation

Interrupt generation within the board is done by gating selected signals from the input jumper area onto the vectored interrupt and/or the CPU's interrupt lines using 74LS05s.

Address Selection and Decoding

Address selection for the MIO is performed with the use of six 74LS86 gates which receive the address bits as one input and receive as the other input a high if the selected address jumper is not present, or a ground if the jumper is present. This will cause the output of the 74LS86s to be asserted if the corresponding address bit is one and the jumper is present; or if it is a zero and the jumper is not present. The six address bits are then ANDed in the 74LS30 together with the fact that either an input or output instruction is being executed (SINP or SOUT) to indicate board selection. The select pulse is used to enable a 74LS155 decoder. The address inputs to this decoder are the two least significant address bits which are jumper selected to provide the desired addresses. The outputs of the 74LS155 consist of four REGISTER LOAD pulses and four READ ENABLEs, one for each of the ports. In the case of the two parallel I/O ports, the REGISTER LOAD and READ ENABLEs are both fed directly to the 8212s.

The DS2 input (to complete selection on the 8212) is controlled by bit 7 of the control register, thus providing the required multiplexing.

Serial I/O Port

The serial I/O port is implemented using a universal asynchronous receiver/transmitter chip (UART). The UART is designed to add the start and stop bits required for transmitting data and to recognize these start and stop bits when receiving data. Note that the jumper configuration for the UART consists of putting +V (Vcc through a 1K resistor) on the control load pin and either ground or +V on the other select pins. The setting of the options pins is discussed in the MIO User Guide.

Parallel I/O Ports

PIO Output Ports

The two parallel input and output ports use the 8212 chips for holding and receiving data. Note again that the most significant bit of the control register is used to determine whether port 1 or port 2 is selected via the DS2 select input pin. When the REGISTER LOAD is executed, the data is loaded during /PWR. The 8212 is deselected on the trailing edge of /PWR which causes the interrupt line (Pin 23) to go high on the 8212. This signal is used as a DATA READY output signal for the port. When the output system has accepted the data, it responds by sending a positive pulse (CLEAR OUTPUT DATA READY) on the strobe input. This causes the interrupt line in the 8212 to be cleared thus indicating that the external interface is ready for more parallel data.

PIO Input Ports

The strobe input from the external device first goes through an EXCLUSIVE OR gate. A jumper to this gate is used to sense a positive strobe, while the absence of a jumper is used for a negative strobe. The LOAD one shots are triggered on the high-to-low transition on the output of the 74LS86's. The second jumper area selects the input strobe, or the LOAD one shot, to gate the data into the 8212 and to set the interrupt line (Pin 23) low thus indicating that input data is ready. If no jumper is used, the input data is continuously available to the 8080. When the 8212 register is read by the computer, the 8212 being selected will cause the interrupt line to reset, indicating to the external system that the data has been accepted, and removing the ready pulse internally.

Control Port

The control register output consists of two 74LS175s; one of which is used to hold the four least significant bits of the data for use in the output jumper area and the other of which is used to hold the four most significant bits for controlling the internal operation.

The internal operations use bit 4 asserted to indicate that a write operation is being performed on the CRI and bit 5 to indicate that a read operation is being performed on the CRI. Bits 6 and 7 are used in two different modes: 1) to select the status input lines for SIOS and PIOS by providing the A and B inputs to the 74LS153 dual 4 to 1 selector; and 2) to multiplex the PIO select lines and the status signal, CRIS. Bit 7 is used to multiplex the PIO ports by having /CR7 as the DS2 input to the 8212s for PIO-Port 1; and CR7 as the DS2 input to the 8212s for PIO Port 2. Bit 6 is used to select the Byte Ready (/CR6) or Bit Ready (CR6) signal for input to the CRIS signal.

The input to the control port is accomplished by selecting the appropriate jumpers in the input area as described in the User Guide. These jumpered inputs are input to the 74LS367s for gating onto the internal data bus.

Cassette Recorder Interface

The Cassette Recorder interface uses the ANSI standard biphase encoding technique to record data on the tape by using a square wave clock to shift the data and EXCLUSIVE ORing the clock with the output data.

Timing

Figure 1 shows a timing illustration of how the CRI interface works with respect to shifting, recording, and recovering the data. The top line shows the serial data which is to be written on the tape. Below this is the clock pulse. The third line shows the serialized data as clocked out of U36. The fourth line shows EXCLUSIVE OR of the clock and the data. Notice that there are two flux reversals or one complete cycle per bit when a constant data stream is being written and only one flux reversal or one cycle per two bits when alternating ones and zeros are being written. The fifth line shows the EXCLUSIVE OR of the data and the inverted clock. The sixth line shows the resultant sinusoidal wave form which is written on the tape. This can also be considered to be the data read directly back from the tape. The seventh line shows the output of the 8T20 which is a digital form of the received data. Line 8 shows the output of the zero crossing one-shot detector as if it were never disabled. Line 9 shows the disable gate for this zero crossing detector. This is the output of the 74LS74 flip-flop. Line 10 shows bit 4 coming high in the counter. The leading edge of bit 4 is used to strobe the data on the return. Line 11 shows the reconstructed data stream.

The reader should become familiar with the diagram before proceeding on with the discussion. Notice that the polarity of the written data and/or the digital recovered data output of the 8T20 can be inverted when it goes through the EXCLUSIVE OR gates. Switches 7 and 8 in the External Address Jumper are used to invert the output and input data, respectively. This option is provided so that the proper data will be fed into your recorder and returned from it independently of the phase on which the recorder operates.

Pin 1 of the 8T20 fed back through R44 provides the hysteresis for the 8T20. The given value of R44 works with most popular recorders, If adjustment should be necessary, its value should be lowered to increase the hysteresis and raised to decrease it.

The shift register used in this section is a 74LS395. This provides both the tri-state outputs for gating onto the internal data bus and the cascadeable output for forming an 8-bit shift register. The timing generator consists of the two 74LS163s and the 74LS293. The 74LS163s should be jumper-selected so that they reload at sixteen times the required data frequency. The 74LS293 divides down the output of the 74LS163s to generate timing for the read and write circuitry.

Cassette Read Operation

In read operation, the first transition received from the recorder starts the CRI clock. After four clock cycles, the eight-bit shift register is clocked, loading the current level of the input data into the register. After twelve clock cycles, the 74LS293 is put into reset, and the 74LS163s are put into LOAD mode, thus presetting and holding them. The entire circuit then idles until the next input transition, which again allows the counters to run.

Referring again to Figure 1, line 8 (labelled zero crossing), represents the output of the 8T20 one shot as if it were never disabled. That is, it generates a short pulse for every zero-crossing transition input from the recorder. The one-shot disable flip-flop (U36), however, prevents the one-shot from detecting a transition from the time the first transition starts the counters until the twelfth clock cycle, when the clocks are disabled.

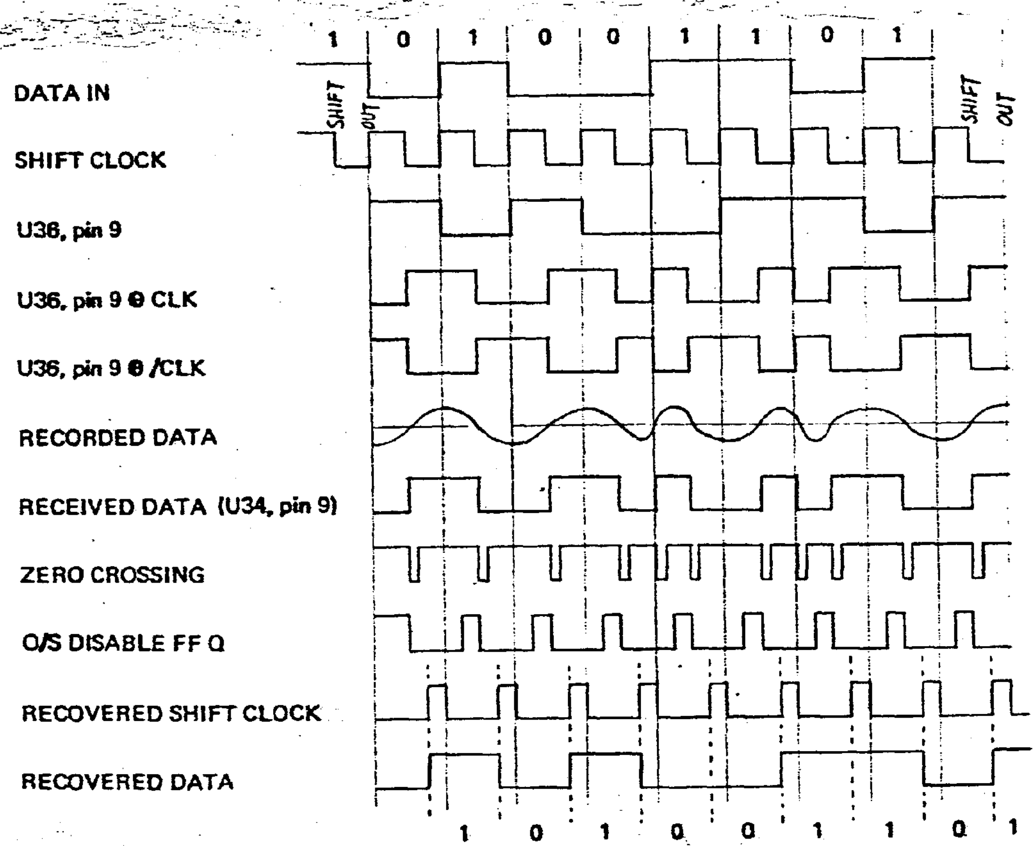
In the input data stream from the recorder, when the present data bit is the same as the previous data bit, a second transition occurs at approximately the eigth clock cycle. Because the one-shot is disabled, this transition will not be detected. However, the next transition will occur after the twelfth clock cycle, enabling the counters, and causing the data level to be read four clock cycles later, as described above.

Because two transitions have occurred since the last time the level was read, the new level and the previous level will be the same, which they should be to represent data bits which are the same.

Read Clock

The pulses from the one-shot will occur once per bit time because of the disabling described above, and are used to generate the clock for the shift register. This clock represents a reconstruction of the original write clock, one that is dependent only on transitions read from the tape, so that the tape format is inherently self-clocking, and immune to even large variations in tape speed.

Figure 1 Timing Diagram



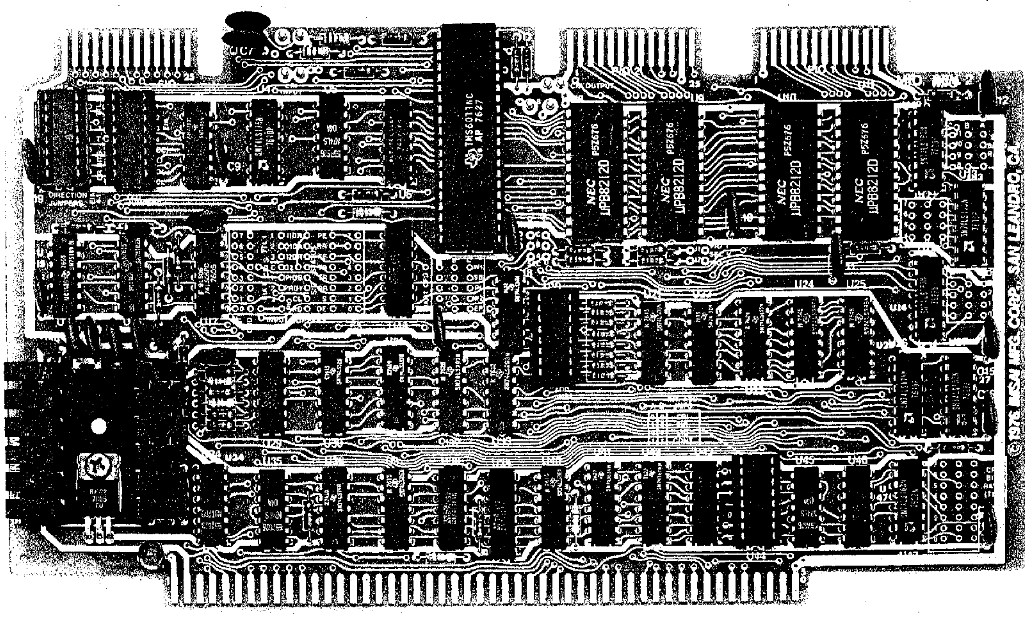
MIO Silk Screen Errata

J5 (Direct Interrupt Jumper) is not shown. It jumpers J1 Pin 4 and J1 Pin 73, as shown on the AP-44 Jumper example in the User Guide.

The following signal names in the output Jumper Area have been interchanged.

O1DR should read I1DR
O2DR should read I2DR
O1DA should read I1DA
O2DA should read I2DA

NOTE: AUGAT PINS NOT SHOWN



MIO REV. 2

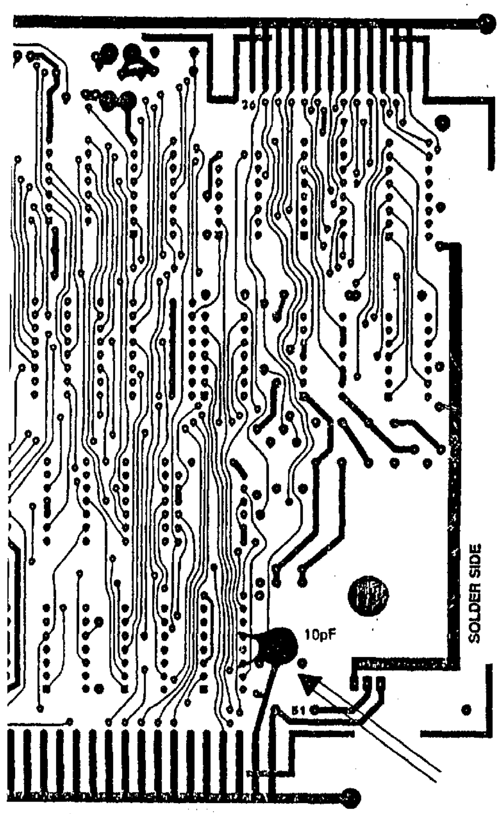
BOARD: MIO Rev. 2

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
PC Board	92-0000042	1	MIO Rev. 2
74LS00	36-0740002	2	Quad 2 Input NAND (Low Power Schottky)/ SN74LS00N
74LS04	36-0740402	3	Hex Inverter (LPS)/SN74LS04N
74LS05	36-0740502	1	Hex Inverter, Open Collector (LPS)/ 74LS05PC
7406	36-0740601	1	Hex Inverter Driver, Open Collector/ SN7406N
74LS30	36-0743002	1	8 Input NAND (LPS)/SN74LS30N
7432	36-0743201	1	Quad 2 Input OR/SN7432N
74LS32	36-0743202	1	Quad 2 Input OR (LPS)/SN74LS32N
74LS51	36-0745102	1	AND-OR Inverter (LPS)/DM74LS51N
74LS74	36-0747402	1	Dual D Flip-Flop (Preset and Clear) (LPS)/SN74LS74N
74LS86	36-0748602	3	Quad 2 Input EXCLUSIVE OR (LPS)/ SN74LS86N
74LS123	36-7412302	2	Dual One Shot/SN74LS123N (Alternate 74123/DM74123N)
74LS153	36-7415302	1	Dual 1 of 4 Data Selector (LPS)/ SN74LS153
74LS155	36-7415502	1	Dual 2 of 4 Line Decoders (LPS)/ SN74LS155N
74LS161	36-7416102	5	4 Bit Counter, Binary Asynchronous Clear (LPS)/SN74LS161N (Alternate 74LS163/SN74LS163N)
74LS175	36-7417502	2	Quad D Type Flip-Flop (LPS)/ 9LS/74LS175
74LS293	36-7429302	2	4 Bit Binary Counter (LPS)/74LS293PC
74367	36-7436701	4	Hex Tri-State Buffer/DM74367N

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
74LS395	36-7439502	2	4 Bit Shift Register with 3 State Outputs/SN74LS395N
75188	36-7518801	1	RS232 Driver/SN75188N
75189	36-7518901	1	RS232 Receiver/SN75189N
7805	36-0780501	1	5V Positive Regulator/7805CU
78L12	36-0781202	1	Regulator/MC78L12CP
8212	36-0821201	4	Input-Output Port/PB8212D
8T20	36-0082001	1	Bi-Directional One Shot/N8T20B
TR1602	36-0601101	1	Universal Asynchrous Receiver/Transmitter 51883/TMS 6011
1N751A	35-1000005	1	Zener Diode/1N751A
1N914	35-1000006	1	Silicon Diode/1N914
1N4742	35-1000009	1	Zener Diode/1N4742
Transistor	35-2000003	1	2N3906 Transistor/2N3906
Capacitor	32-2010010	17	.1uF Disk Ceramic
Capacitor	32-2233070	2	33uF, 25V Tantalum
Capacitor	32-2002010	2	.02uF Disk Ceramic
Heat Sink	16-0100004	1	Thermalloy 6 Prong/6072B
Header	23-0400001	4	16 Pin IC Header
Socket Carrier	23-0900008	9	Lead Socket Carrier Assembly/ AUGAT 716-AG2D
Socket	23-0800001	4	16 Pin Solder Tail IC Socket
Socket	23-0800004	1	40 Pin Solder Tail IC Socket
Resistor	30-2560462	1	56 Ohm, ½ Watt/green, blue, black
Resistor	30-3100362	5	100 Ohm, ¼ Watt/brown, black, brown (3 are supplied for optional 60MA current loop.)

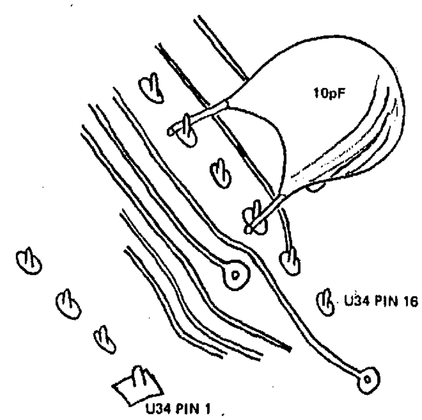
Parts List

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION/IDENTIFYING MARKS
Resistor	30-3330362	1	330 Ohm, ¼ Watt/orange, orange, brown
Resistor	30-3220362	1	220 Ohm, ¼ Watt/red, red, brown
Resistor	30-3470362	3	470 Ohm, ¼ Watt/yellow, violet, brown
Resistor	30-3470462	1	470 Ohm, ½ Watt/yellow, violet, brown
Resistor	30-4100362	20	1K Ohm, ¼ Watt/brown, black, red
Resistor	30-4120362	2	1.2K Ohm, ¼ Watt/brown, red, red
Resistor	30-5100362	6	10K Ohm, ¼ Watt/brown, black, orange
Resistor	30-5360362	5	36K Ohm, ¼ Watt/orange, blue, orange
Solder	15-0000001	10'	
Screw	20-3302001	1	6-32x5/16" Phillips Pan Head Machine
Nut	21-3120001	1	6-32 CAC Hex Nut
Lockwasher	21-3350001	1	#6 Internal Tooth CAC Lockwasher
Cassette	88-0000019	1	Test Cassette
Capacitor	32-0210010	1	10pF Disk Ceramic



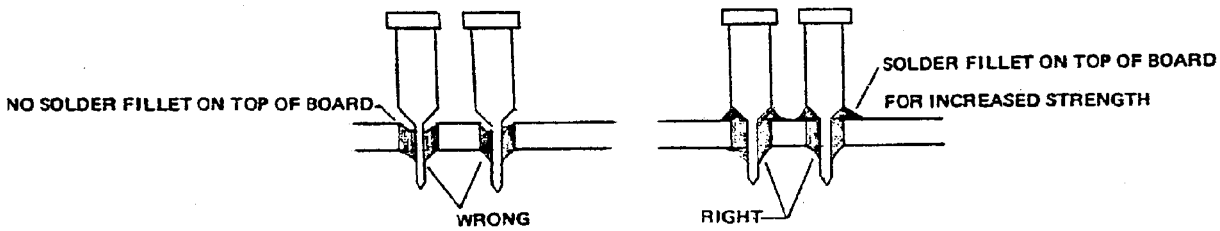
ERRATA MIO REV.2 1/22/77

Board Assembly Modification to lengthen the pulse generated by the 8T20 for more reliable resetting of U36: After the board is assembled, install a 10 picofarad disk capacitor between U34 pins 12 and 14. Cut leads to 1/8" and solder to U34 pins 12 and 14 on the solder side of the board as shown. Take care not to damage the solder mask or create a solder bridge to the neighboring traces.



Assembly Note:

1. When installing socket pins for jumpers, heat should be applied long enough (e.g. 3 seconds rather than 1 second) to allow solder to wick through the board and form a fillet on the component side. (Alternately solder can be applied from the top side.) This provides greater support to the socket pins so they won't bend during jumper installation. Number 26 or 27 wire, solid, is ideal to use for jumpers. Larger wire, up to 24 solid or cut leads from ¼ Watt resistors, may be used; however, the larger wire may spring the internal contacts, requiring that you always use the larger wire. Often, leads cut from signal diodes (1N914 and 1N4148) are the ideal smaller diameter.



2. It has come to our attention that the jumpers between the Augat pins can short to other pins. This can be solved by using either some spaghetti tubing on the No. 26 bare wire or using No. 26 solid insulated wire for these jumpers. DO NOT SOLDER JUMPERS INTO THE AUGAT PINS.

ASSEMBLY DIAGRAM AND SILK SCREEN ERRATA

IC's U41 through U43 are shown on both the Assembly Diagram and Silk Screen as 74LS367. Provided in your kit are 74367's. Please use the IC's provided in your kit.

ECN 77-0004

ASSEMBLY INSTRUCTIONS

( ) 1. Unpack your board and check all parts against the parts list enclosed in the package.

( ) 2. If gold contacts on the edge connector appear to be corroded, use pencil eraser to remove any oxidation. NOTE: Do not use Scotchbright or any abrasive material as it will remove the gold plating.

RESISTOR INSTALLATION

( ) 3. Insert and solder the five 36K Ohm, ¼ watt (orange, blue, orange) resistors at locations R1, R4, R24, R43 and R44 as shown on the Assembly Diagram.

( ) 4. Insert and solder the twenty 1K Ohm, ¼ watt (brown, black, red) resistors at locations R2, R3, R5, R6, R17, R18, R20, R22, R23, and R31 through R41 as shown on the Assembly Diagram.

( ) 5. Insert and solder the two 1.2K Ohm, ¼ watt (brown, red, red) resistors at locations R9 and R15 as shown on the Assembly Diagram.

( ) 6. Insert and solder the three 470 Ohm, ¼ watt (yellow, violet, brown) resistors at locations R10, R12 and R29 as shown on the Assembly Diagram.

( ) 7. Insert and solder the six 10K Ohm, ¼ watt (brown, black, orange) resistors at locations R11, R19, R21, R25, R26, and R42 as shown on the Assembly Diagram.

( ) 8. Insert and solder the two 100 Ohm, ¼ watt (brown, black, brown) resistors at locations R27 and R28 as shown on the Assembly Diagram.

( ) 9. Insert and solder the one 220 Ohm, ¼ watt (red, red, brown) resistor at location R30 as shown on the Assembly Diagram.

( ) 10. Insert and solder the one 330 Ohm, ¼ watt (orange, orange, brown) resistor at location R14 as shown on the Assembly Diagram.

( ) 11. Insert and solder the one 56 Ohm, ½ watt (green, blue, black) resistor at location R7 as shown on the Assembly Diagram.

( ) 12. Insert and solder the one 470 Ohm, ½ watt (yellow, violet, brown) resistor at location R8 as shown on the Assembly Diagram.

IC INSTALLATION

All Pin 1's are toward the lower right hand edge of the PC board and the 100 pin connector. The pads for Pin 1 are square.

( ) 13. Insert and solder the one 7406 at location U3 as shown on the Assembly Diagram.

( ) 14. Insert and solder the two 74LS175s at locations U4 and U26 as shown on the Assembly Diagram.

( ) 15. Insert and solder the three 74LS04s at locations U5, U35 and U45 as shown on the Assembly Diagram.

( ) 16. Insert and solder the four 8212s at locations U8 through U11 as shown on the Assembly Diagram.

( ) 17. Insert and solder the five 74LS161s (or 74LS163s) at locations U12 through U14, U27 and U47 as shown on the Assembly Diagram.

( ) 18. Insert and solder the one 75189 at location U15 as shown on the Assembly Diagram.

( ) 19. Insert and solder the one 75188 at location U16 as shown on the Assembly Diagram.

( ) 20. Insert and solder the one 74LS05 at location U17 as shown on the Assembly Diagram.

( ) 21. Insert and solder the one 74LS30 at location U18 as shown on the Assembly Diagram.

( ) 22. Insert and solder the three 74LS86s in locations U21, U37 and U40 as shown on the Assembly Diagram.

( ) 23. Insert and solder the one 7432 at location U22 as shown on the Assembly Diagram.

( ) 24. Insert and solder the four 74367s at locations U6 and U41 through U43 as shown on the Assembly Diagram.

( ) 25. Insert and solder the one 74LS155 in location U23 as shown on the Assembly Diagram.

( ) 26. Insert and solder the two 74LS395s at locations U24 and U25 as shown on the Assembly Diagram.

( ) 27. Insert and solder the two 74LS00s at locations U29 and U46 as shown on the Assembly Diagram.

( ) 28. Insert and solder the one 74LS32 at location U30 as shown on the Assembly Diagram.

( ) 29. Insert and solder the one 74LS153 at location U31 as shown on the Assembly Diagram.

( ) 30. Insert and solder the two 74LS293s at locations U32 and U33 as shown on the Assembly Diagram.

( ) 31. Insert and solder the one 8T20 at location U34 as shown on the Assembly Diagram.

( ) 32. Insert and solder the one 74LS74 at location U36 as shown on the Assembly Diagram.

( ) 33. Insert and solder the one 74LS51 at location U38 as shown on the Assembly Diagram.

( ) 34. Insert and solder the two 74123s at locations U39 and U48 as shown on the Assembly Diagram.

( ) 35. Insert and solder the 40 pin solder tail socket at location U7 as shown on the Assembly Diagram.

DISCRETE COMPONENT INSTALLATION

( ) 36. Insert and solder the seventeen .1uF disk capacitors at locations C2 through C5 and C8 through C20 as shown on the Assembly Diagram.

( ) 37. Insert and solder the 33uF tantalum capacitor at location C1 as shown on the Assembly Diagram. NOTE: Observe polarity (+ to +) as shown on the board.

( ) 38. Insert and solder the two .02UF capacitors at locations C6 and C7 as shown on the Assembly Diagram.

( ) 39. Insert and solder the 1N914 diode at location CR3 as shown on the Assembly Diagram.

( ) 40. Insert and solder the 1N4742 zener diode at location CR1 as shown on the Assembly Diagram.

( ) 41 Insert and solder the 1N751 zener diode at location CR2 as shown on the Assembly Diagram.

( ) 42. Insert and solder the 2N3906 transistor at location Q1 as shown on the Assembly Diagram.

( ) 43. Insert and solder the four 16 pin sockets at locations U1, U2, U19 and U44 as shown on the Assembly Diagram.

REGULATOR AND HEAT SINK INSTALLATION

( ) 44. Before installing the heat sink and regulator, bend the 7805 regulator leads at 90 degree angles to facilitate mounting on the heat sink.

( ) 45. Insert a #6 screw through the 7805 regulator and heat sink on the component side of the board and attach through the lockwasher and nut on the circuit side of the board. Tighten the screw carefully to insure proper alignment of the heat sink to prevent shorting to adjacent traces. Solder in the 7805 regulator leads.

( ) 46. Insert and solder the 78L12 regulator at location U49 as shown on the Assembly Diagram, above and to the left of the heat sink.

( ) 47. Insert and solder the lead sockets in the jumper pads in the various jumper areas. In most of these areas, the jumper pads are in lines, spaced on 0.1 inch centers, the same spacing as the lead sockets on their carriers. This allows you to insert the sockets in groups and hold them with the carrier while you solder them.

( ) 48. Finally, the UART chip, TR1602 or alternate, should be inserted in its socket at U7 with Pin 1 down toward the 100 pin edge connector at the bottom of the board. Addressing and baud rate jumpers should be installed and other option jumpers installed as required (see the User Guide). The board is ready for use.

<u>MIO USER GUIDE</u>

I .....GENERAL

The MIO board gives the User the following capabilities:

- one serial I/O port
- two parallel I/O ports
- one cassette I/O port
- one control port

NOTES ON THE USER GUIDE

The information which is needed to set-up and use the MIO board is divided into two classes: 1) information which is common to all types of I/O ports used on the MIO board; and 2) information which is required to set up a particular type of port. The USER GUIDE is structured to parallel this division.

Two sections of the USER GUIDE (II. Address Selection and III. Control Port Operation) contain information pertaining to the operation of all I/O ports (CRI, SIO, PIO-1, and PIO-2) used on the MIO board. The User is advised to read these sections before going on to the individual procedures for the SIO, PIO, or CRI ports.

The sections devoted to the individual ports (SIO Procedures, PIO Procedures, and CRI Procedures) include all information for setting up, using, and testing that particular type of port. This includes information on interface connections, jumper options, software access, and test program procedures and listings.

All test programs (Appendix B & C) assume the jumper settings shown in figure 9. If a jumper or jumper area is not mentioned, no particular configuration is required.

ORDER OF INSTALLATION

To avoid having to continually enter the test programs from the front panel, it is adviseable to complete the CRI interface before going on to the SIO or PIO ports. The test cassette can then be used to load Test Programs for checking the other ports (see Appendix A for a description of the test cassette).

II.....ADDRESS SELECTION

Address selection for the MIO board consists of: 1) SELECTING A GROUP of 4 I/O port addresses; and 2) CONFIGURING the 4 I/O ports within the selected group of 4 I/O port addresses.

Address selection is achieved through the use of THE EXTERNAL ADDRESS JUMPER AREA and THE INTERNAL ADDRESS JUMPER AREA.

EXTERNAL ADDRESS JUMPER AREA

The External Address Jumper Area selects one group of four I/O port addresses out of the 64 possible groups of addresses that the MIO board may occupy. This is accomplished by selecting Address bits 2-7 at jumper position U19.

Table 1 shows the relationship between a jumper position and an address bit. NOTE that jumper numbers 7 and 8 are USED FOR THE CRI CHANNEL.

For any given address bit, a ONE is selected if a jumper is installed in the corresponding jumper position. A ZERO is selected if no jumper is installed.

Table 1. Group Address Selection

Jumper	IC Pins	Address Bits
1	8, 9	7
2	7, 10	6
3	6, 11	5
4	5, 12	4
5	4, 13	3
6	3, 14	2
7	2, 15	CRI
8	1, 16	CRI

INTERNAL ADDRESS JUMPER AREA

Table 2 shows the possible combinations. All legal jumper combinations are shown. The comment column indicates the hardware (and software) compatability of the port combinations assuming the appropriate status inputs for the given application have been selected in the Input Jumper Area (see Section III.1).

TABLE 2 - Internal Address Selection

Jumpers Inserted	(Pin Number)	Port Numbers	Port Referenced	Comments
		0	CRI	
1	(8, 9)	1	PIO	IMSAI SIO
		2	SIO	
6	(3, 14)	3	CONT	
		0	PIO	
1	(8, 9)	1	CRI	Processor Tech
		2	CONT	3P+S
8	(1, 16)	3	SIO	
		0	SIO	
3	(6, 11)	1	CONT	
		2	CRI	
6	(3, 14)	3	PIO	
		0	CONT	
3	(6, 11)	1	SIO	Altair SIO
		2	PIO	
8	(1, 16)	3	CRI	
		0	CRI	Use Parallel port
2	(7, 10)	1	SIO	to be compatible
		2	PIO	with IMSAI SIO
5	(4, 13)	3	CONT	Software
		0	PIO	
4	(5, 12)	1	CONT	
		2	CRI	
5	(4, 13)	3	SIO	
		0	SIO	
2	(7, 10)	1	CRI	
		2	CONT	
7	(2, 15)	3	PIO	
		0	CONT	Use Parallel port
4	(5, 12)	1	PIO	to be compatible
		2	SIO	with Altair SIO
7	(2, 15)	3	CRI	Software

III.....THE CONTROL PORT

The CONTROL PORT is a complete 8 bit Input/ Output Port used for internal and external control functions. The operation of the Control Port is easily understood if we separate its functions into two categories: 1)Input Functions; and 2) Output Functions.

CONTROL PORT INPUT FUNCTIONS

As an input port, the eight bits of the Control Port serve to monitor 1) the status of the CRI, SIO, and PIO ports; and 2) external I/O control lines. All input functions of the Control Port are determined by the configuration of the INPUT JUMPER AREA (IJA).

CONTROL PORT OUTPUT FUNCTIONS

As an output port, the eight bits of the Control Port are used for external and internal control functions according to the following division.

1. BITS 0-3

   Bits 0-3 are latched and used to control external devices when needed. The function of these bits are determined by the configuration of the OUTPUT JUMPER AREA (OJA).

2. BITS 4-7

   Bits 4-7 are latched and serve three functions: 1) to control CRI functions;
   2) to select PIO ports 1 or 2; and
   3) to decode the status signals PIOS and SIOS.

   TABLE 3 shows the decoding of bits 4-7.

If either of the two status signals PIOS or SIOS are used as an input to the Control Port, bits 6 and 7 of the Control Port output word are used to determine which of the error lines are active. The decoding for this function is shown in Table 4.

Table 3 - Control of CRI and PIO

CONT BIT	VALUE	USE
4	1	Enable CRI Write Circuitry
	0	Disable CRI Write Circuitry
5	1	Enable CRI Read Circuitry
	0	Disable CRI Read Circuitry
6	1	Enable CRI Ready on each bit
	0	Enable CRI Ready on each byte
7	1	Select PIO Port 2
	0	Select PIO Port 1

Table 4 - Control Selection of PIO and SIO Status

Value of Control Signal		Available from Input Selector	
Bit 7	Bit 6	PIOS	SIOS
0	0	Port 1 Output Data Ready	Error = PE or FE or OE
0	1	Port 1 Input Data accepted	Overrun Error (OE)
1	0	Port 2 Output Data Ready	Parity Error (PE)
1	1	Port 2 Input Data accepted	Framing Error (FE)

III.1 .....INPUT JUMPER AREA

The INPUT JUMPER AREA is organized as shown in figure 2. Row A contains the 8 input bits of the Control Port. Rows B, C, and D contain three types of signals: 1) the status input sources; 2) the serial data input to the UART; and 3) the output of EIA, TTL, and Current Loop receivers used for receiving serial data and I/O control lines. Row E contains 8 Ground connection points and Row F contains the 8 vectored interrupt lines. All signals are defined in TABLE 5 by location and signal name.

FUNCTIONS

The INPUT JUMPER AREA serves four functions.

1) It allows the User to jumper any of the status input sources located in Rows B, C, and D to any of the 8 input bits of the Control Port (Row A).

2) It allows the User to jumper EIA and TTL receivers connected to I/O control lines (Rows C and D) to any of the 8 input bits of the Control Port (Row A).

3) It allows the User to jumper the Serial Data Line to Current Loop Receivers, EIA Receivers, or TTL Receivers for the SIO Channel of the MIO Board.

4) It allows the User to jumper any of the Vectored Interrupt Lines contained in Row F to any of the status input sources contained in Rows B. C and D.

Table 5: Input Jumper Area Signal Definition

Location	Signal Name	Description
A0 thru A7	SI0 thru SI7	Data Input for Bits 0-7
B0	OE	SIO UART Overrun Error
B1	SIOS	Determined by CNTL, see Table 4
B2	/RRDY	Logical Inversion of RRDY
B3	/TRDY	Logical Inversion of TRDY
B4	TRDY	SIO UART Ready for Transmit Data
B5	FE	SIO UART Framing Error
B6	RRDY	SIO UART has Received Data Ready
B7	PE	SIO UART Parity Error
C0	RDATA	SIO UART Receive Data Input Line
C1	CL1	Current Loop Input Data (+ on J4-8, - on J4-22)
C2	PRDY	O1DA or I1DR or O2DA or O2DR
C3	PIOS	Determined by CNTL, see Table 4
C4	I2DA	PIO Port 2 is Ready for More Output Data
C5	O2DR	PIO Port 2 has Input Data Ready
C6	I1DA	PIO Port 1 is Ready for More Output Data
C7	O1DR	PIO Port 1 has Input Data Ready
D0	REIA4	EIA Receiver Number 4
D1	REIA3	EIA Receiver Number 3
D2	REIA2	EIA Receiver Number 2
D3	REIA 1	EIA Receiver Number 1
D4	CRIS	Bit Ready or Byte Ready from CRI
D5	ITTL3	TTL Direct Input 3 (J9-12)
D6	ITTL2	TTL Direct Input 2 (J4-6)
D7	ITTL1	TTL Direct Input 1 (J4-16)
E0 thru E7	Ground	For Disabling Interrupts or Zeroing Data Bits
F0 thru F7	VI0 thru VI7	Interrupt Request Selects

SOURCE DEFINITIONS

The possible sources in ROWS B, C, and D are defined as follows.

B0-B6

B0-B6 are status signals used for the SIO channel. Note that B1,(SIOS) is a logical OR'ing of PE, FE, and OE. If this signal is used, the Control Port output word allows the User to decode this signal to determine which error (PE,FE,or OE) occurred. This is covered in CONTROL PORT OUTPUT FUNCTIONS.

C0-C6

C0 is the Serial Data which is to be input to the SIO Channel.

C1 (CL1) is the Current Loop Receiver for the SIO Channel.

C2-C7 are status signals used for the two PIO channels. Note that C2 and C3 are the logical OR'ing of the signals I1DA, O1DR, I2DA, and O2DR. If C3 (PIOS) is used, the Control Port output word allows the User to decode this signal to determine which of the four error lines is active. This is covered in the section, CONTROL PORT OUTPUT FUNCTIONS.

D0-D7

D0-D3 provide for 4 EIA Receivers to be used with the SIO Channel.

D4 is a status line used for the Cassette Channel to indicate when a bit or byte is ready.

D5-D7 provide for three TTL level inputs for the SIO Channel.

E0-E7

E0-E7 provide for 8 Ground points.

F0-F7

F0-F7 provide for the 8 Vectored Interrupt Lines.

NOTE

The configuration needed for each type of port will be covered in the SIO, CRI, and PIO Procedures.

III.2......OUTPUT JUMPER AREA

The OUTPUT JUMPER AREA is located at position U2 on the MIO Board, and is organized into three groups.

1) Pins 13-16 are the output bits 0-3 of the Control Port.

2) Pin 12 is the Serial Transmit Data from the SIO Channel.

3) Pins 1-8,10,11 are Drivers for EIA, TTL,CURRENT LOOP, and HIGH VOLTAGE (40v. 40Ma) levels.

4) 9 is Ground

The signals present at the Output Jumper Area are defined in Table 6.

FUNCTIONS

The OUTPUT JUMPER AREA serves two functions.

1. It allows the User to jumper the Serial Transmit Data from the SIO Channel to any one of the three types of output drivers (EIA, TTL, and Current Loop).

2. It allows the User to jumper Control output bits 0-3 to any of the output drivers to be used as I/O Control Lines.

The configuration needed for each type of port will be covered in the SIO, PIO, and CRI Procedures.

NOTE

If the Current Loop Driver is not used, it should be jumpered to the Ground signal at pin 9 of the OUTPUT JUMPER AREA.

Table 6 Output Jumper Area Signal Definitions

PIN#	SIGNAL NAME	DESCRIPTION
16	CR0	Control Register Bit 0
15	CR1	Control Register Bit 1
14	CR2	Control Register Bit 2
13	CR3	Control Register Bit 3
12	TDATA	SIO UART Serial Transmit Data
11	OTTL1	TTL Direct Output 1 (J4-10)
10	OTTL2	TTL Direct Output 1 (J4-2)
9	GND	Ground
8	DEIA1	EIA Transmitter Number 1
7	DEIA2	EIA Transmitter Number 2
6	DEIA3	EIA Transmitter Number 3
5	DEIA4	EIA Transmitter Number 4
4	OC1	High Voltage (40V) Power (40MA) Driver 1 (J4-23)
3	OC2	High Voltage (40V) Power (40MA) Driver 2 (J4-4)
2	OC3	High Voltage (40V) Power (40MA) Driver 3 (J4-19)
1	CLO	Current Loop Output (+on J4-20, -on J4-2

IV.....SIO PORT PROCEDURES

The SIO Port is a full 8 bit serial input/output port. It is used in conjunction with the Control Port, which in this case allows the User to 1) read selected status lines from the UART: and 2) read and write on external I/O control lines.

Setting up the SIO Port involves three steps:

1. configuring the hardware jumpers;
2. making the external interface connections; and
3. running test programs to check out the operation of the port.

IV.1...HARDWARE JUMPERS

INPUT JUMPER AREA

In the INPUT JUMPER AREA:

1. The serial data from the appropriate receiver (TTL, EIA, or Current Loop) must be jumpered to the RDATA terminal (C0) to be input to the UART.

2. The desired UART status signals and external control signals must be jumpered to the Control Port inputs. The UART status signals are available at Row B (B0-B7) and need to be jumpered to the desired input bit of the Control Port, available at Row A (A0-A7). Any external control signals will be taken from the appropriate type of receiver (TTL or EIA) and jumpered to the desired bit of the Control Port (Row A).

OUTPUT JUMPER AREA

In the OUTPUT JUMPER AREA:

1. The serial data from the UART (U2-12) must be jumpered to the appropriate transmitter (EIA, TTL, or Current Loop).

2. The output bits 0-3 of the Control Port must be jumpered to the appropriate type of transmitter EIA, TTL, or OC) to be used as external control signals.

SIO STATUS SIGNALS

PE - If a Parity Error occurs, PE goes high;
FE - If a Framing Error occurs, FE goes high;
OE - If an Overrun Error occurs, OE goes high;
SIOS - If any of the signals, PE, FE, or OE, are active, SIOS will go high. The type of error which occurred may be determined by using the Control Output Port bits 6 and 7 as shown in Table 4.
TRDY, $\overline{\text{TRDY}}$ - UART Transmitter Ready; and
RRDY, $\overline{\text{RRDY}}$ - UART Receiver Ready.

These signals are most typically used by jumpering them in the Output Jumper Area to bits of the Control Port (column A of the OJA).

SIO CONFIGURATION JUMPER AREA

The UART can be configured to transmit and receive a variety of character lengths and parity configurations. The SIO Configuration Jumper Area is used to hardwire the configuration desired. It provides +V (for a logic 1) on Row B and Ground (for a logic 0) on Row A for connection to the configuration inputs in Row C. Table 7 defines these inputs. Note that all inputs connected to +V provide the standard TTY configuration.

Table 7. UART Configuration Definition

SIGNAL	VALUE	UART OPERATION
	1	Do not transmit or check parity
PI	0	Transmit and check parity
	1	Transmit 1.5 stop bits for 5 bit characters, 2 for all others
SBS	0	Transmit 1 stop bit per character
WLS1 &	00	5 Bits/Character
WLS2	01	6 Bits/Character
	10	7 Bits/Character
	11	8 Bits/Character
	1	Generate and check Even Parity
EPE	0	Generate and check Odd Parity

SIO BAUD RATE SELECTION

The Baud rate for the UART is formed by dividing down Phase II. This permits the User to select virtually any rate between 45.5 and 9600 baud. The division is accomplished by presetting a 12 bit counter and incrementing it to a value of 4084, at which time it is reloaded. The formula for determining the preset value is: (in base 10)

P.V. = 4085 - (125,000/BAUD RATE)

In the SIO BAUD RATE JUMPER AREA, Row A provides Ground (used when the preset is a 0), Row B provides +V (used when the preset is a 1), and Row C is the counter input. Table 8 gives the preset value for standard BAUD rates.

Table 8 - Baud Rate Jumper Selections

BAUD RATE	PRESENT VALUE	HEX REP.	BINARY VALUE BY-BIT 11	10	9	8	7	6	5	4	3	(MSB=11) 2	1	0
9600	4072	FE8	1	1	1	1	1	1	1	0	1	0	0	0
4800	4059	FDB	1	1	1	1	1	1	0	1	1	0	1	1
2400	4033	FC1	1	1	1	1	1	1	0	0	0	0	0	1
1200	3981	F8D	1	1	1	1	1	0	0	0	1	1	0	1
600	3877	F25	1	1	1	1	0	0	1	0	0	1	0	1
300	3668	E54	1	1	1	0	0	1	0	1	0	1	0	0
150	3252	CB4	1	1	0	0	1	0	1	1	0	1	0	0
134.5	3156	C54	1	1	0	0	0	1	0	1	0	1	0	0
110	2949	B85	1	0	1	1	1	0	0	0	0	1	0	1
75	2418	972	1	0	0	1	0	1	1	1	0	0	1	0
45.5	1338	53A	0	1	0	1	0	0	1	1	1	0	1	0

IV.2 ....EXTERNAL INTERFACE CONNECTIONS

EIA CONNECTIONS

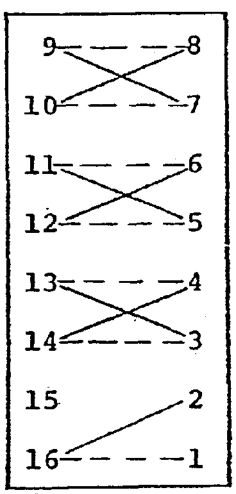
Table 9 gives the signal names for the SIO connections to the 26 pin edge connector and the corresponding EIA 25 pin connector number. Signals marked with an asterisk are standard RS232 definitions. The RS232 definition is given with respect to the terminal.

DIRECTION JUMPER AREA

The DIRECTION JUMPER AREA allows the SIO port to act as the computer or terminal end of an EIA RS232 line. Figure 1 shows the configuration for the two modes of operation.

Figure 1: Direction Configuration for EIA

INTERNAL SIGNAL	EIA DEFINITIONS
BA-Transmit Data	DEIA1
BB-Receive Data	REIA1
CA Request to Send	DEIA2
CB Clear to Send	REIA2
CD Data Terminal Ready	DEIA3
CC Data Set Ready	REIA3
Not Used	DEIA4
CF Carrier Detect	REIA4



______________ Connections Made to run complete EIA interface with a terminal
\- - - - - - - - - - - - - Connections made to run complete EIA interface with a modem

Table 9
SIO CONNECTOR (J4) SIGNAL DEFINITION

MIO Edge Connector	EIA Connector	Signal Name
1	1	Chassis Ground AA*
2	14	TTL Out 2
3	2	Transmit Data BA*
4	15	Open Collector Out 3
5	3	Receive Data BB*
6	16	TTL in 1
7	4	Request to Send CA*
8	17	Current Loop in +
9	5	Clear to Send CB*
10	18	TTL Out 1
11	6	Data Set Ready CC*
12	19	TTL in 2
13	7	Signal Ground AB*
14	20	Data Terminal Ready CD*
15	8	Carrier Detect CF*
16	21	TTL in 3
17	9	+5 Volts
18	22	------------------
19	10	Open Collector Out 2
20	23	Current Loop Out +
21	11	EIA Driver or Receiver
22	24	Current Loop In -
23	12	Open Collector Out 1
24	25	Current Loop Out -
25	13	------------------
26	--	------------------

* EIA Standard Signal Designation

CURRENT LOOP CONNECTIONS

The Current Loop Signals are:
1) IN+ (J4-8)
2) IN- (J4-22)
3) OUT+ (J4-20)
4) OUT- (J4-24).

Resistors R9 and R12 are defined to be 1.2K Ohms on the schematic. These resistors provide for 20 mA Current Loops on both Input and Output. If a 60 mA Current Loop is needed, the values of these two resistors should be changed to 100 Ohms.

IV.3.....SIO TEST PROGRAMS

There are four test programs which can be used with the SIO. To use the test programs the board should have the status bits configured as defined in Section 1.

SIO TEST 1

The board should be jumpered to interface with the peripheral to be used for this test. The starting address is 3100H. The value of the sense switches is continuously output as a character;and an input character (if any) is displayed in the Sense Lights. The Sense Lights will display the last character until a new character is received. If an error occurs, the Sense Lights will be set to all ones. The program will pause for fifteen seconds each time the value of the Sense Switches is changed.

SIO TEST 2

The board should be jumpered to interface with the peripheral to be used for this test. The starting address is 3103H. The SIO RRDY signal is continuously monitored and each time a character is received, it is transmitted to the SIO output. Errors on input cause the character to be ignored.

SIO TEST 3

The board should be jumpered to connect the SIO serial output to the SIO serial input for this test. The starting address is 3106 H.

The Sense Switches are used to define any bits which should not be transmitted as part of this test. Switches should be set to a zero for all bit positions to be transmitted (i.e., for seven bits, the MSB is set; for 6 bit, the two MSBs are set; etc.). If you leave your terminal connected to the board while running this test, do not attempt to type in as this will generate an error.

The test continously transmits all possible binary combinations within the pattern and compares the received results. If the test is running without error, the Sense Lights will all be out.

If a PE, OE, or FE occurs, the program will display ØFF Hex until one of the Sense Switches is changed. When a change is made, these errors are displayed in positions 4, 3, and 2 respectively. Changing the Sense Switches will cause the program to continue. If a failure occurs in the transmitted value versus received value, the program will display ØFE Hex. Changing of the Sense Switches will cause the value of the transmitted character to be displayed in the Sense Lights.

Changing the Sense Switches a second time will cause the value of the received character to be displayed in the front panel lights. Changing of the Sense Switches will also cause the program to continue with the next value.

V ...... PIO PORT PROCEDURES

The two parallel I/O ports available on the MIO are both addressed with the same I/O address from the 8080. The ports are multiplexed using bit 7 of the Control Output word as discussed in Section III. The two ports operate identically and have identical external interfaces on J2 and J3.

The two PIO Ports are used in conjunction with the Control Port, which in this case allows the User to read selected Status Signals from either or both ports.

The PIO Output Ports each contain eight output data lines and three control lines. The PIO Input Ports each contain 8 input data lines and two control lines.

Setting up the PIO Port involves:

1. confuguring the hardware jumpers;
2. making the external interface connections; and
3. running test programs to check out the operation of the port.

V.1 ....HARDWARE JUMPERS

INPUT JUMPER AREA

In the INPUT JUMPER AREA:

The PIO status signals (C3 - C7) must be jumpered to the desired input bits of the Control Port (Row A).

PIO STATUS SIGNALS

IDA - Input Data Accepted (as defined in PIO External Interface Connections)

ODR - Output Data Ready (as defined in PIO external Interface Connections)

PIO STATUS SIGNALS (cont.)

PIOS - If any of the signals IDA or ODR goes high, PIOS will go high. The signal which occurred may be determined by using the Control Output Port bits 6 and 7 as shown in Table 4.

INPUT STROBE JUMPER AREA

The INPUT STROBE JUMPER AREA allows the User to select one of five types of input strobe signals. Note that J1 and J3 for PIO port 1 correspond to J2 and J4 for PIO port 2.

If no jumper is placed at J3 (J4 for PIO port 2), the data lines will be monitored but not latched. A jumper is placed from A to C on J3 (J4) if an external pulse is used as the input strobe. A jumper is placed from B to C if edge triggering is to be used.

If J1 (J2 for PIO 2) is present, a positive strobe is selected. If J1 (J2) is omitted, a negative strobe is selected.

V.2 .... EXTERNAL INTERFACE CONNECTIONS

Table 10 lists the signals to be used in interfacing the two parallel I/O ports with external devices.

PIO CONTROL SIGNALS

The PIO Output Ports each have 3 Control Lines. These are defined as follows:

ODR - an Output Data Ready Line for each port, to indicate to the processor when the output device and thus the output port is ready to receive data. ODR may be monitored using Interrupts or the Control Input Port. This signal can be used as a positive data strobe for the external output device.

PIO CONTROL SIGNALS (cont.)

CODR - a Clear Output Data Ready Line for each port to set the ODR Lines active low. This signal is generated from the external device when it is ready to receive data.

OSTB - a negative Strobe Line is provided from each parallel output port. It may be used as an external strobe to an output device. It has the same timing as $\overline{\text{PWR}}$.

Normally when an external I/O device is ready to accept data, it asserts CODR, which in turn sets the ODR Line active low. When the processor finishes outputting data to the output port, ODR is reset high, providing a positive strobe to the external output device.

The PIO Input Ports each have two Control Lines. These are defined as follows:

IDA - one Input Data Accepted Line for each port, to indicate to the processor when data has been loaded from the external input device. This line is normally set low when the STB is received from the external device.

ISTB - one Input Strobe Line is provided for each port to strobe the data into the input latches and to set the IDA lines low. This signal originates at the external input device. Data is strobed on the leading edge of ISTB if there is no jumper at J1, and on the trailing edge if jumper J1 is present.

Normally the external input device sends a STB with the data. This latches the data and sets IDA low. When the processor senses IDA low (via Interrupts or the Control Port), it reads the data from the latch which in turn resets IDA high.

Table 10

PIO CONNECTOR (J2 and J3) SIGNAL DEFINITION

MIO EDGE CONNECTOR	EIA CONNECTOR	SIGNAL NAME
1	1	Ground
2	14	+16 Volts
3	2	Output Data Bit Ø
4	15	Input Data Bit Ø
5	3	Output Data Bit 1
6	16	Input Data Bit 1
7	4	Output Data Bit 2
8	17	Input Data Bit 2
9	5	Output Data Bit 3
10	18	Input Data Bit 3
11	6	Output Data Bit 4
12	19	Input Data Bit 4
13	7	Output Data Bit 5
14	20	Input Data Bit 5
15	8	Output Data Bit 6
16	21	Input Data Bit 6
17	9	Output Data Bit 7
18	22	Input Data Bit 7
19	10	Output Data Ready
20	23	Input Data Accepted
21	11	External Output Strobe
22	24	Input Strobe
23	12	+5 Volts
24	25	-16 Volts
25	13	Output Strobe
26	--	-----------------

V.3....PIO TEST PROGRAMS

There are three tests shown in Appendix B for the PIO. Test 1 (starting at address 3109 H) continuously reads the Sense Switches and outputs this value to both PIO Ports and to the Sense Lights. Test 2 continuously inputs from PIO Port 1 and outputs to PIO Port 1, PIO Port 2 and the Sense Lights. Test 3, starting at 31ØF Hex, has the same output as Test 2; the difference is that the input is from PIO Port 2.

To test the inputs using these tests, a jumper wire or test clip can be used to alternately apply ground and +5 volts to each input pin, while observing the effect on the Sense Lights. Ground applied to a pin will turn off the corresponding light, while +5 volts will turn it on. For protection, insert a 470 Ohm resistor in series with the test lead.

To test the outputs, monitor each output pin for the appropriate logic level, as set by the Sense Switches, using a voltmeter, logic probe, or oscilloscope.

VI......CRI PORT PROCEDURES

The CRI Port supports ANSI (Tarbell) and Byte/Lancaster Formats for storage and retrieval of information to and from cassette tape.

The CRI Port is used in conjunction with the Control Port, which in this case is used to read the status signal, CRIS, indicating whether a bit or byte is ready. The function of CRIS is determined by output bit 6 of the Control Port (Table 3).

ANSI (Tarbell) Formats

Data is recorded on the tape using Biphase Encoding to directly support the ANSI (Tarbell) data Format. The standard data rate is 1500 bits per second (187 bytes/second). This may be increased depending on the quality of the cassette recorder used.

Byte/Lancaster Formats

To support the Byte/Lancaster Format, the software tape handler in Appendix B must be used.

The conversion of Biphase Data Formats into Byte/Lancaster Formats is explained in the following discussion. Biphase Encoding results in two flux reversals per bit (one cycle) when recording a constant string of ones or zeros; and one flux reversal per bit (one-half cycle) when recording a string of alternating ones and zeros (see the Theory of Operation for more detail). Hence, recording a byte of all ones in the Biphase mode results in eight cycles being recorded. Recording a byte of alternating ones and zeros (e.g., 1010 1010 AA Hex) results in four cylces being recorded.

The Byte/Lancaster standard for recording data is then achieved by changing the recording speed to 2400 bits/second and recording a byte of FF Hex or 55 Hex to represent a one or zero bit respectively. For more detailed infromation, the User is referred to the article by Lancaster in the first issue of Byte Magazine.

DATA FORMATS

Writing a block of data to cassette consists of writing a Start Byte (for synchronizing the hardware data separation logic), a Sync Byte (for software recognition as a start-of-block indicator), the data bytes, and a check byte(s).

Reading the data back requires recognizing the Sync Byte, reading and storing the data bytes and then using the check bytes to insure the data was properly transferred.

Appendix B contains subroutines for writing the Start and Sync Bytes, writing a Data Byte, recognizing the Sync Byte and reading a Data Byte. Also included are handlers for writing or reading a block of 256 bytes using the standard CRC data check for insuring that the data is proper.

The routines listed in Appendices B and C are recorded in Tarbell format on the test cassette which is shipped with the board. The cassette is more fully described in Appendix A.

Setting up the CRI Port involves:

1. configuring the hardware jumpers;
2. making the external interface connections; and
3. running test programs to check out the operation of the port.

VI.1 ... HARDWARE JUMPERS

INPUT JUMPER AREA

In the INPUT JUMPER AREA:

The status signal CRIS (D4) must be jumpered to the desired input bit of the Control Port (Row A).

CRIS

This signal goes high to indicate to the processor when a bit or byte is ready. It may be jumpered to the Interrupt Lines or to the Input bits of the Control Port. Note that bit 6 of the Control Output Port is used to select a bit or byte ready (see Table 3).

CRI BIT RATE JUMPER AREA

The bit rate for recording data is formed by dividing down Ø2. This permits the User to select any data rate from 488 to 62,500 bits per second. The division is accomplished by presetting an 8 bit counter and counting it up to 255, at which time it is reloaded. The output of this counter is further divided by 16 to form the final recording speed. The formula for determining the preset value is: (in base 10)

P.V. = 256 - (125,000/BIT RATE)

In the CRI Bit Rate Jumper Area, Row A provides Ground (used when the preset is a zero). Row B provides +V (used when the preset value is a one), and Row C is the counter input. Table 11 gives the standard bit rates.

MIO
Errata
2/18/77

The test cassette supplied with your MIO Board has been recorded at 800 bits per second instead of 1500 as implied in the chapter.

Add the following line to Table 11: Standard Bit Rate (on page 2 - 31).

Bit Rate	Preset Value	Hex Repr.	Binary Value By Bit (MSB=7) 7	6	5	4	3	2	1	0
800	100	64	0	1	1	0	0	1	0	0

See the attached sheet Figure 5a for setting up the MIO Board to run 800 BPS.

ECN 77-0014

MIO
Errata
2/18/77

It has come to our attention that to read cassettes on an MIO Board that have been written on a Tarbell Board, the preferred Bit Rate is 1689 BPS.

Add the following line to Table 11: Standard Bit Rate (on page 2-31).

Bit Rate	Preset Value	Hex Repr.	Binary Value by Bit (MSB=7) 7	6	5	4	3	2	1	0
1689	182	B6	1	0	1	1	0	1	1	0

ECN 77-0014

Table 11: Standard Bit Rate

BIT RATE	PRESET VALUE	HEX REPR.	7	6	5	4	3	2	1	0
			BINARY VALUE BY BIT (MSB=7)							
4800	230	E6	1	1	1	0	0	1	1	0
2400	204	CC	1	1	0	0	1	1	0	0
1500	173	AD	1	0	1	0	1	1	0	1

RECORDING PHASE JUMPERS

Jumpers 7 and 8 of the External Address Jumper Area serve as the Recording Phase Jumpers. They serve to invert the polarity of the data written to the cassette. This option is determined by the phase of the recorder and the procedures for determining this are given in the CRI Initial Adjustments.

VI.2 ...EXTERNAL INTERFACE CONNECTIONS

Sockets are provided on the MIO Board for two each input and output lines for the CRI interface. This allows interface to two cassette recorders simultaneously, though only one may be read at a time

An optional cable set (IMSAI Cable M) is available to bring the cassette lines out to the back panel to the 8080 chassis. It terminates in a standard miniature phone jack at the back panel. Two cables, one for input and one for output, are included in each set, and one set is required for each recorder to be interfaced.

VI.3 ... Initial Adjustments

The adjustments required for operating consist of finding the proper volume settings for recording and reading back the data, and setting the interface so that it reads and writes in the proper phase (using jumper 7 and 8 respectively in the External Address Jumper Area). First find the input settings as follows:

1. Insert the test cassette to read on side 1.
2. Set the tone control on your recorder for best high frequency response.
3. Turn the volume to a middle position.
4. Load the Sync Recognition Program from Appendix C using the front panel switches.
5. Start the program at location 3ØØØ Hex.
6. Press the "play" button on your recorder.
7. Adjust the volume until the Sense Lights are all 1's. When the Sync Byte is recognized, the Sense Lights will all be 1, otherwise 0. If the lights are all 1's, go to step 9.
8. If the Sense Lights do not come on, insert Jumper 7 in the External Address Jumper Area. This will reverse the playback phase the interface uses. Repeat Step 7.
9. Adjust the volume in both directions until the Sense Lights go out. The middle setting of this range should be used for all future reading data.

Now, the appropriate output setting should be found as follows:

1. Insert a BLANK tape into your recorder.
2. Load the Sync Generation Program from Appendix B using the front panel switches.
3. Adjust your volume control to lowest position.
4. Start the program at location 3Ø1F Hex.
5. Start the recorder in record mode.
*6. Slowly and uniformly increase the volume until it reaches the maximum. This should be done so when the tape is read you can use the timing relationship to determine the best recording volume.

* On some recorders with AGC, the volume control has no effect on recording signals. Omit step 6 in these cases.

7. Rewind the tape and read the tape using the program from above and playback volume determined there. Use the Sense Lights to determine the best recording volume.

8. If the Sense Lights do not come on during step 7, insert Jumper 8 in the External Address Area (to reverse the recording phase) and repeat the above steps.

VI.4 ...CRI Recording and Reading Procedures

You are now ready to use your recorder to read in the programs from Appendix B. The object programs start about 3 minutes and 30 seconds from the start of the tape. The first 3 minutes is a sync stream consisting of recorded E6's; the next 30 seconds is ØØ's and after this are the MIO test programs. The steps to be used whenever recording or reading data are given below.

For recording a data block:

1. Turn the volume control all the way down.
2. Position the tape to the desired recording location.
3. Get to the point in the program where you can start recording with the push of a button.
4. Start the cassette recorder and slowly increase the volume to the proper setting.
5. Wait 5 seconds for writing leader, then start the program.
6. Stop the cassette when the program indicates the write operation is complete.

For Reading a data block:

1. Set the volume control to the playback position determined during the initial adjustment procedure.
2. Position the tape to the desired playback position (2 or 3 seconds into the leader).
3. Get to the point in the program where you can start reading with the push of a button.
4. Start the cassette in the playback mode and then start the program.
5. Stop the cassette when the program indicates the read operation is complete.

To read the object program from Appendix B, load the Bootstrap Program contained in Appendix A. Use the read procedure as defined above with the following additions.

1. Start tape position is 3 minutes and 30 seconds into side 1 of the tape.
2. Starting address for Bootstrap Program is 3800 Hex.
3. Programs will be completely loaded in 20 seconds.
4. The Programmed Output light will go out when finished

VI.5 ... CRI TEST PROGRAMS

To test the operation of your cassette with the CRI, two steps are required. First, use the block recording procedure to write a block of data onto a blank tape. The program to do this is contained in Appendix B and starts at location 3112 Hex. The Sense Lights are initially set to C0H, program completion is indicated by the lights going to 0. Then read the block using the read procedure and the program contained in Appendix A with a Starting Address of 3115 Hex. Successful completion of the read is indicated by the sense lights going to zero. Sense light read-out of FFH indicates a CRC error. Changing of the sense switches will cause the data compare to be done. Sense light read-out FEH indicates a data compare error. In this case, changing the switches once causes the display of the byte error (this is also the data). The second change causes the bad data to be displayed and the third time causes the compare to continue.

VII.... PERIPHERAL INTERFACING

This section will define the jumper configurations required to interface the MIO board with different types of peripherals. An example will be given for standard serial EIA interfaces, serial current loop interfaces (for teletypes) and a parallel interface. Prior to reading this section, the reader should review Section 1.2 as a refresher of the standard jumpers assumed. A set of illustrations showing jumper configurations for a number of common peripherals appears at the end of this section. Two worksheets for laying out your own jumpers are included.

VII.1.. RS-232-C EIA Interfaces

In addition to the jumpers specified in Section 1.2, the following two jumpers must be added.

IJA - REIA1 to RDATA
OJA - TDATA to DEIA1
GND to CLO (to put output current loop in
active state)

This provides all of the signals required for the interface to be standard interface. The cable shield or ground should be attached to the pad as indicated on the Assembly Diagram. It should be noted that the SIO Baud Rate and UART Configuration Jumpers must be installed to match the peripheral equipment. Insertion of one of the two possible Direction Configuration Jumpers will then complete the required jumpers. Table 12 shows the signals driven and received in the two configurations.

Table 12 RS-232-C EIA Signals

I/O BITS	TO RUN TERMINAL	TO RUN MODEM
SIO PORT (all)	DATA IN & OUT	DATA IN & OUT
Control IN - BIT 7	REQUEST TO SEND	CLEAR TO SEND
Control IN - BIT 6	DATA TERMINAL READY	DATA SET READY
Control IN - BIT 5	NOT USED	CARRIER DETECT
Control IN - BIT 1	RECEIVED DATA READY	RECEIVE DATA READY
Control IN - BIT 0	TRANSMIT READY	TRANSMIT READY
Control OUT - BIT 2	CARRIER DETECT	NOT USED
Control OUT - BIT 1	CLEAR TO SEND	REQUEST TO SEND

VII.2 .. Serial Current Loop Interface

The simplest current loop interface to a Teletype uses only the serial input and output data lines. Hence, only bits 1 and 0 of the Control Input are used to indicate transmitter and receiver status. Internal to the MIO, the following jumpers must be added.

IJA - CLI to RDATA
OJA - TDATA to CLO
SIO BAUD - Jumper for 110 Baud
SIO CONFIG - All Jumpers to +V (i.e., Row B to C)

There is a terminal strip located at the right rear of the teletype (ASR33 or KSR33). The terminal strip is behind a panel of square white plastic connectors and also connects to the TTY power cord. The terminals are numbered from 1 to 9. The connectios required between the MIO and these terminals are shown in Table 13. In addition to making these connections, it may be necessary to perform the following operations on your teletype.

1. Full Duplex Operation - Move YEL/BRN wire from Terminal 3 to Terminal 5 and move WHT/BLU wire from Terminal 4 to Terminal 5.
2. Change receiver current level from 60 ma to 20 ma; move VIO wire from Terminal 8 to Terminal 9.
3. Change current source resistor to 1450 Ohms. Locate the current source resistor in front of the power supply and move the BLU wire to the tap labelled 1450.

Table 13 Connections for ASR33 and KSR33

Signal Name	26 Pin Edge Connector	25 Pin EIA Connector	Terminal Strip
Current Loop Out +	20	23	7
Current Loop Out -	24	25	6
Current Loop In +	8	17	3
Current Loop In -	22	24	4

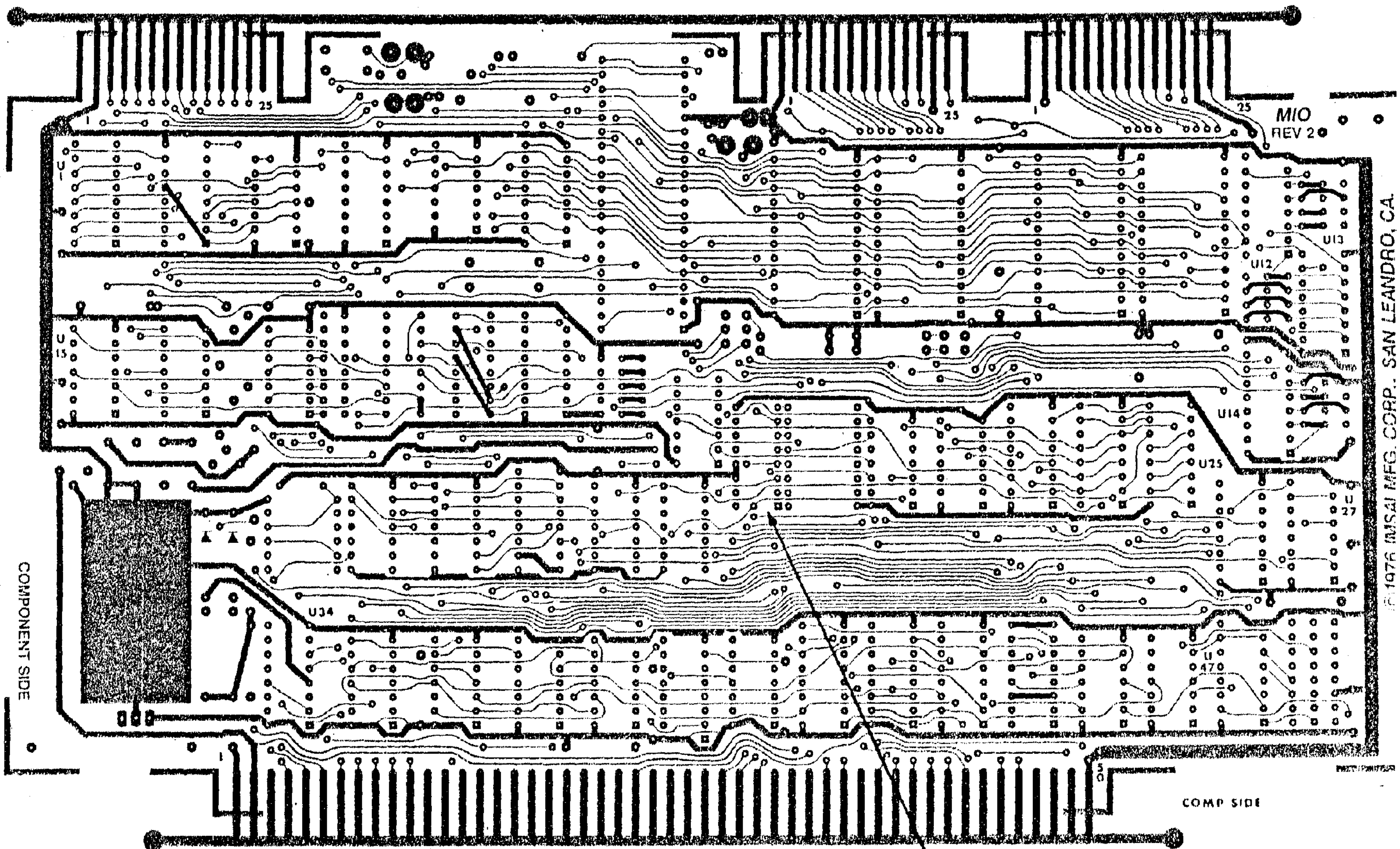
VII.3 .. Parallel Interface

The IMSAI Key-1 Keyboard provides an example of a parallel interface. The keyboard uses one PIO input port with its associated handshake signals. The example shown in the illustration at the end of this section uses the processor interrupt request line to signal that an input character is ready, and the interrupt acknowledge to signal acceptance of the character.

Interface Examples

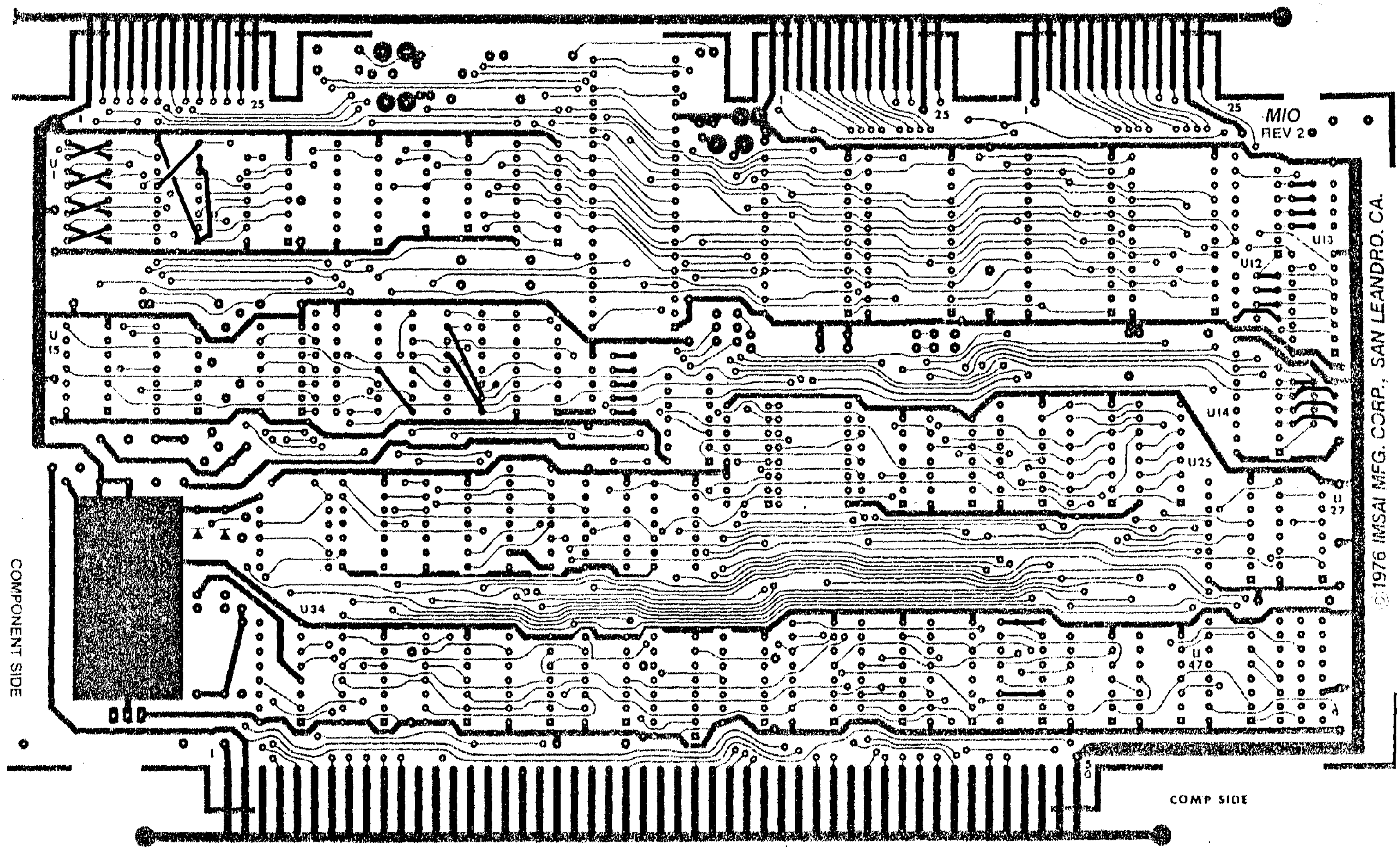
Figure 2 below shows the location of the jumper areas described in the User Guide. Specific examples of the use of these jumpers for interfacing common peripherals are shown on the following pages, followed by a worksheet that you can use to lay-out your own jumper connections.

VII.4 JUMPER EXAMPLE ILLUSTRATIONS



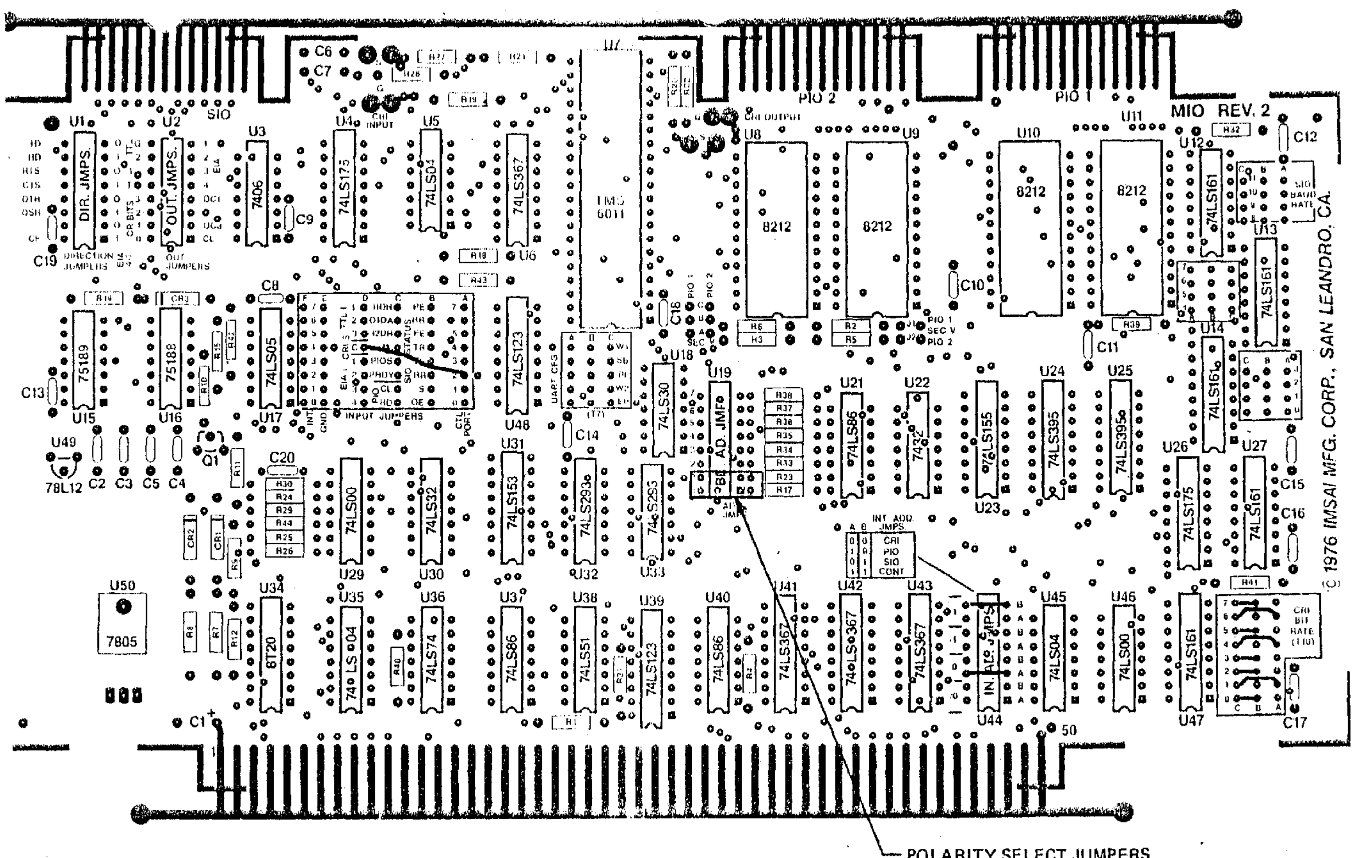
BOARD ADDRESS JUMPERS.
LEAVE OPEN TO PUT BOARD AT BLOCK 0.

TELETYPE

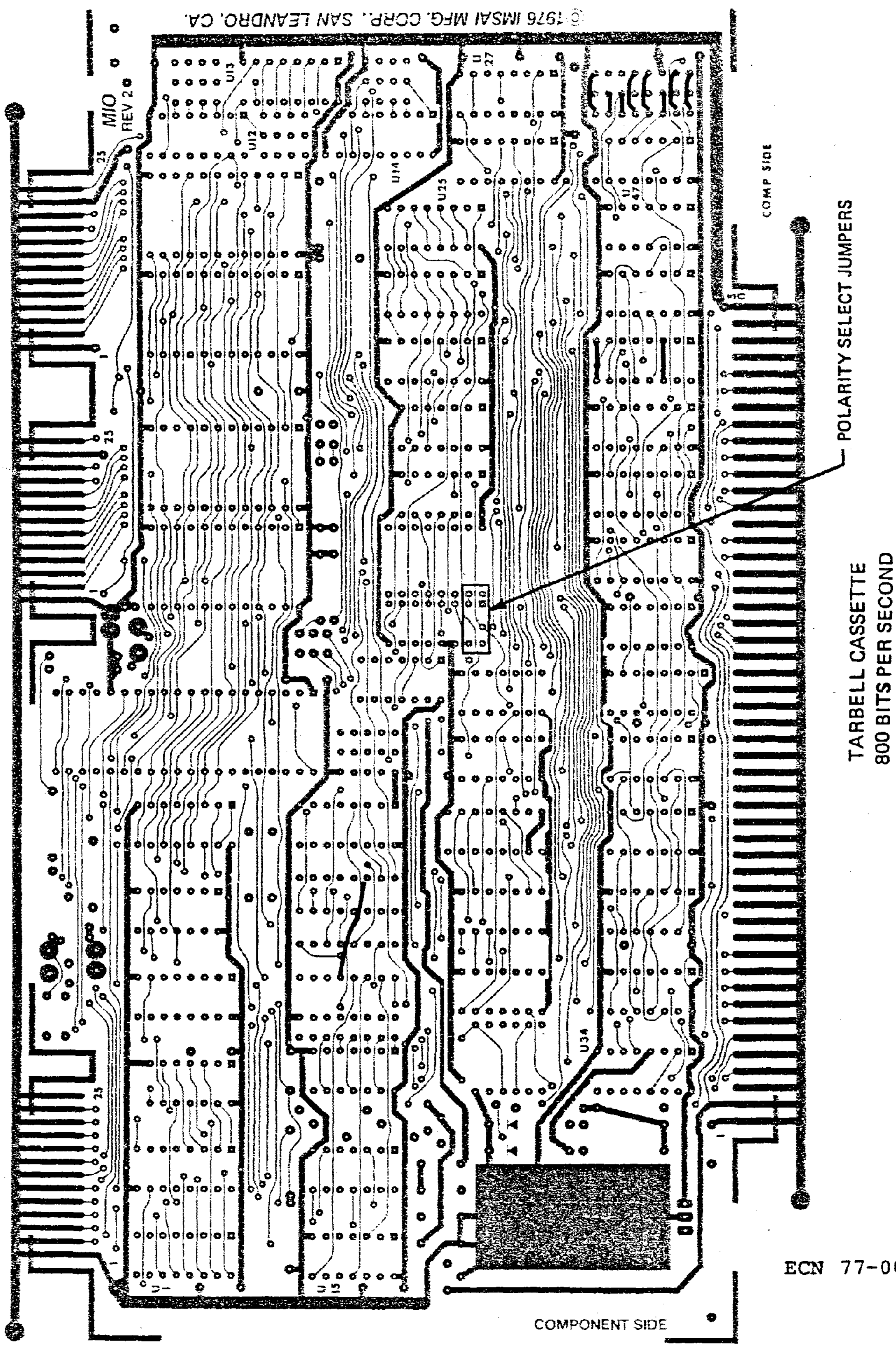


ADM–3
9600 BAUD
© 1976
IMSAI MFG. CORP.
SAN LEANDRO, CA.

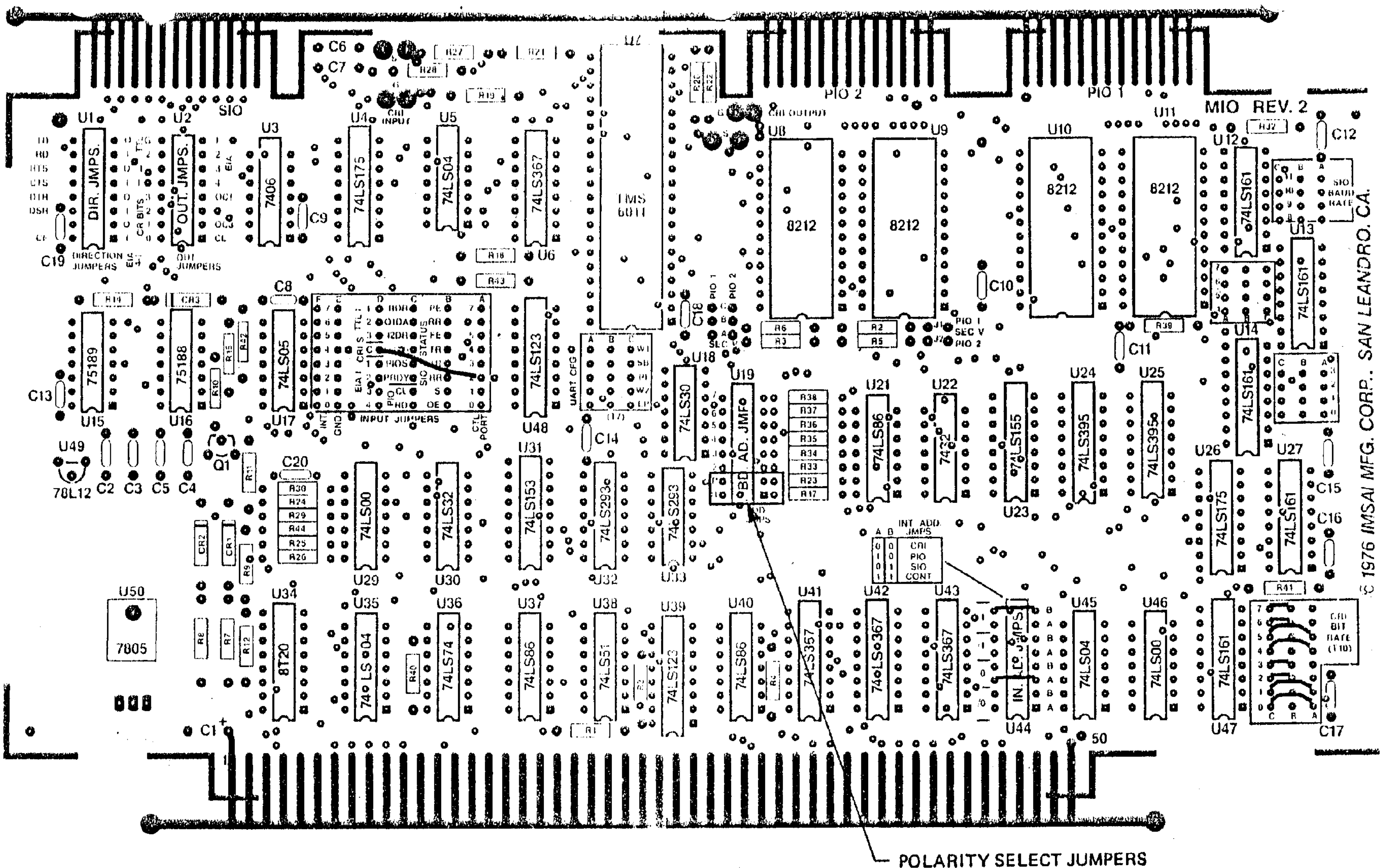
ECN 77–0011 2/77



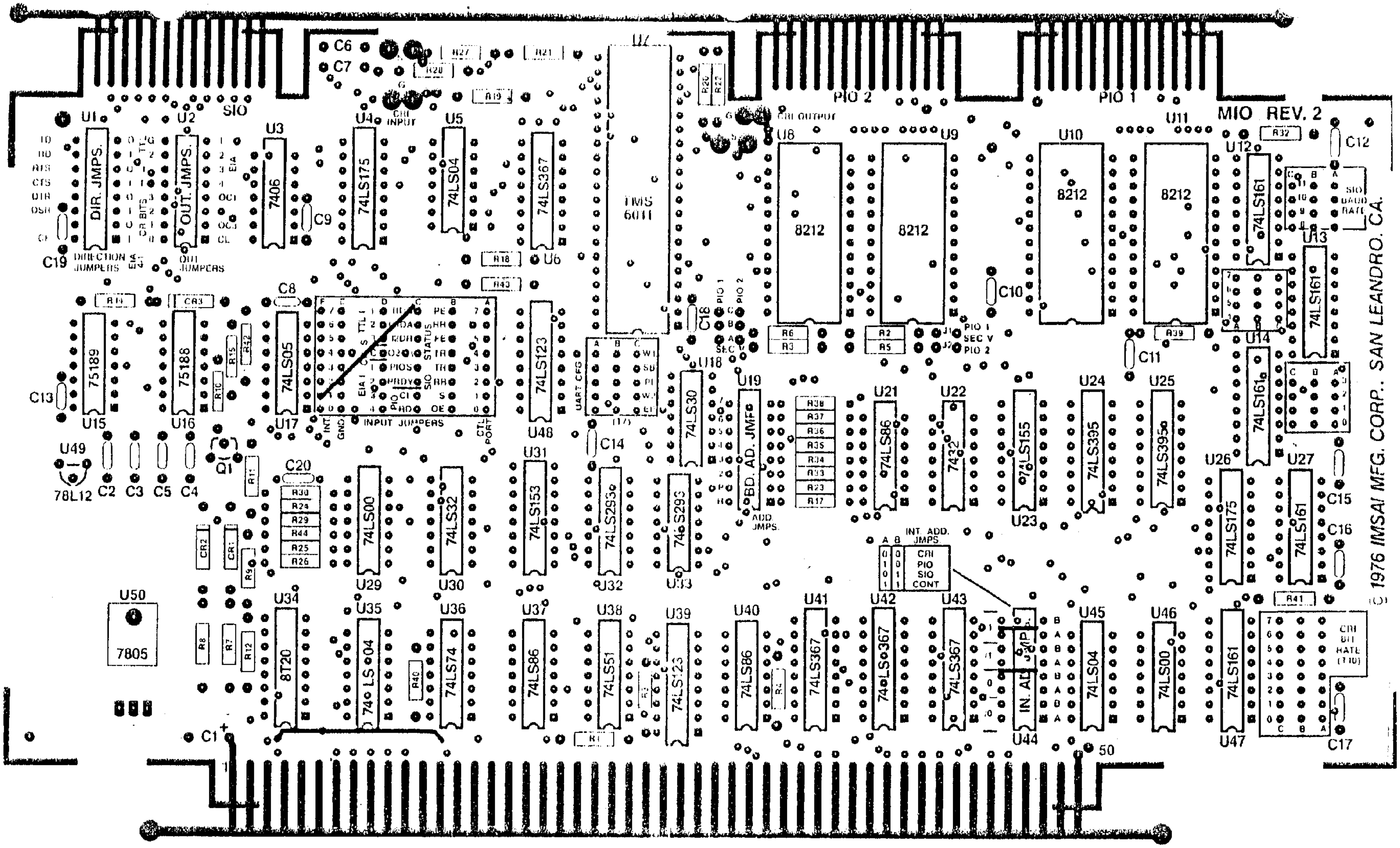
TARBELL CASSETTE
1500 BITS PER SECOND


©1976 IMSAI MFG. CORP., SAN LEANDRO, CA.
MIO
REV 2
COMP SIDE
POLARITY SELECT JUMPERS
TARBELL CASSETTE
800 BITS PER SECOND
COMPONENT SIDE

ECN 77-0014

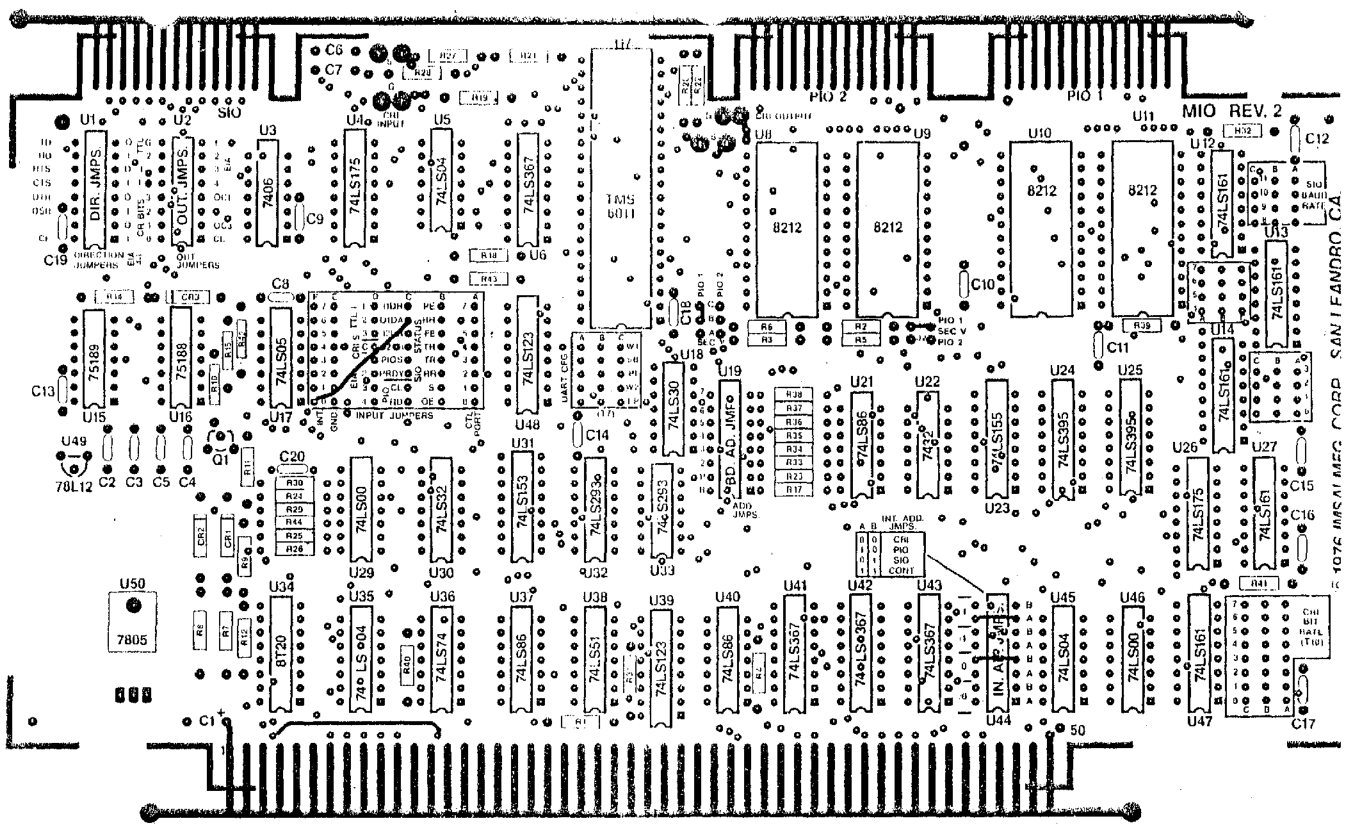


BYTE / LANCASTER
2400 BITS PER SECOND

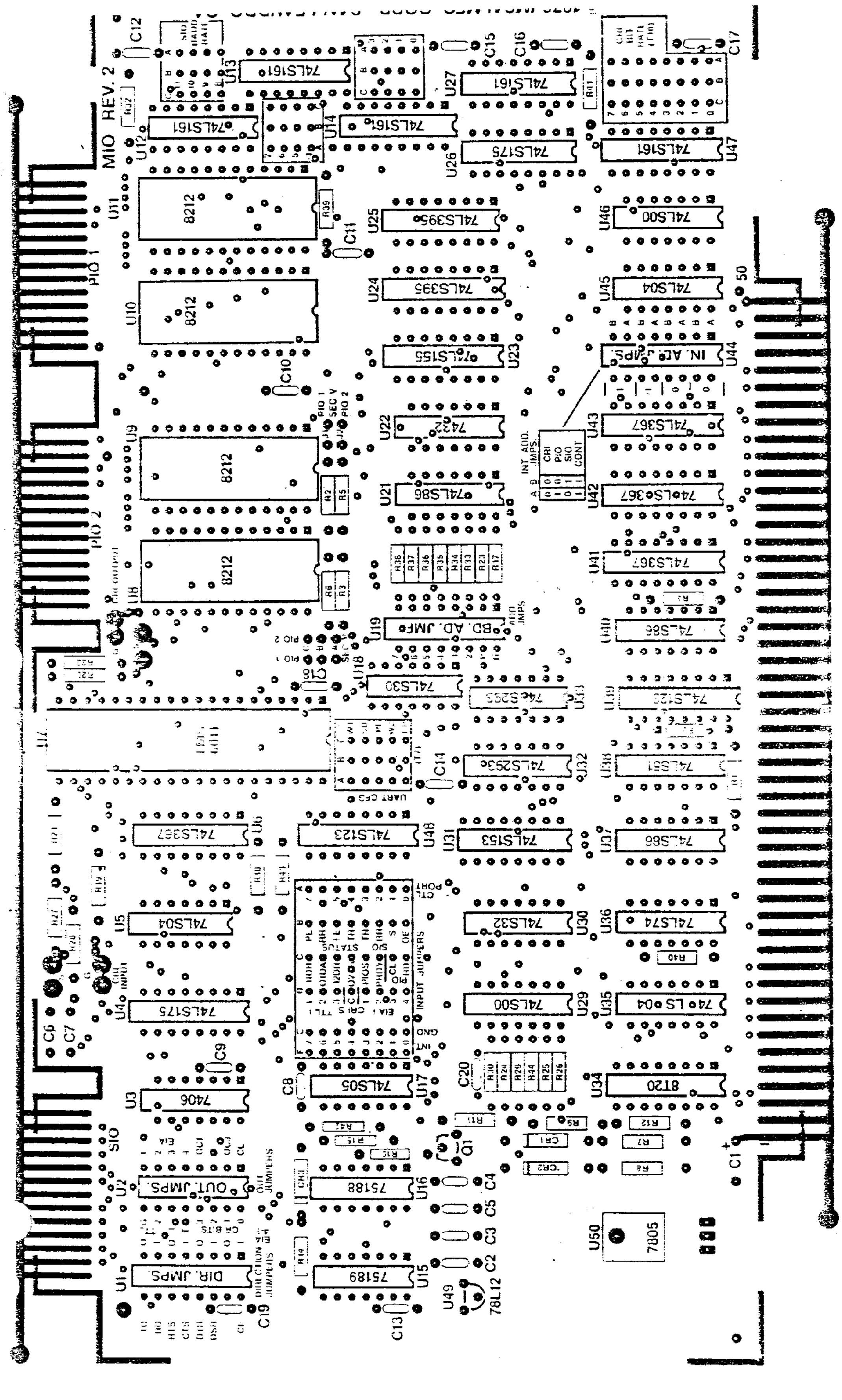


AP—44 PRINTER

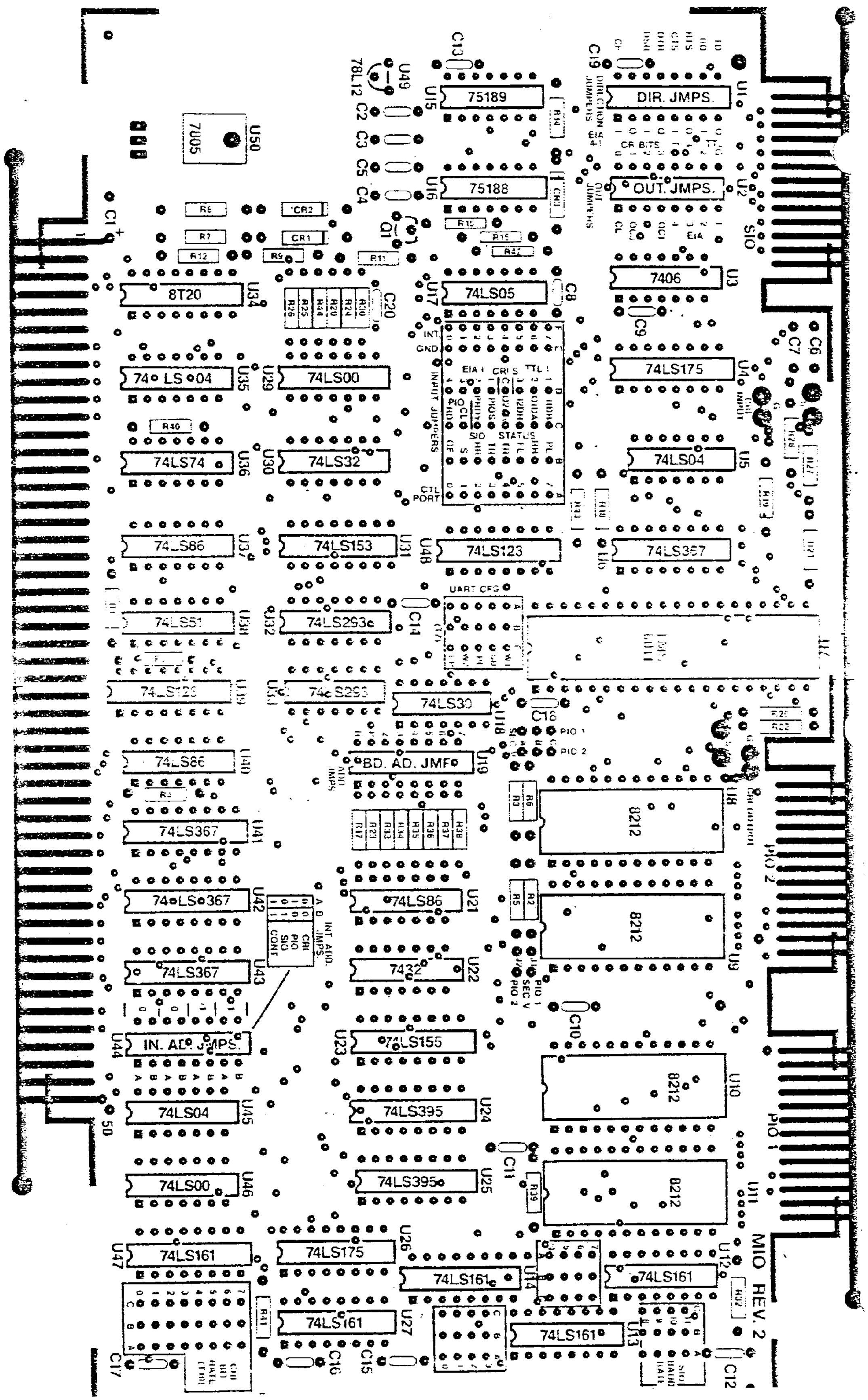
NON—VECTORED INTERRUPT MODE



KEY–1 KEYBOARD
NON–VECTORED INTERRUPT MODE



MIO REV. 2
PIO 1
PIO 2
SIO
8212
74LS161
74LS175
74LS395
74LS155
74LS367
74LS86
74LS00
74LS04
74LS153
74LS123
74LS32
74LS74
8T20
7406
75188
75189
7805
78L12



MIO REV. 2
7805
U50
78L12
U49
75189
U15
DIR. JMPS.
U1
75188
U16
OUT. JMPS.
U2
SIO
8T20
U34
74LS05
U17
7406
U3
74 LS 04
U35
74LS00
U29
74LS175
U4
74LS74
U36
74LS32
U30
74LS04
U5
74LS86
U37
74LS153
U31
74LS123
U48
74LS367
U6
74LS51
U38
74LS293
U32
74LS123
U39
74LS293
U33
74LS30
U18
74LS86
U40
BD. AD. JMP
U19
PIO 1
PIO 2
74LS367
U41
8212
U8
74LS367
U42
74LS86
U21
8212
U9
74LS367
U43
7432
U22
IN. AD. JMPS.
U44
74LS155
U23
8212
U10
74LS04
U45
74LS395
U24
74LS00
U46
74LS395
U25
8212
U11
74LS161
U47
74LS175
U26
74LS161
U14
74LS161
U12
74LS161
U27
74LS161
U13
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
C14
C15
C16
C17
C18
C19
C20
Q1

MIO

User Guide

Appendices

APPENDIX A

Test Cassette Description

The Test Cassette contains the programs MIOA and MIOB (the listings of which appear in Appendix B and C, respectively) recorded in standard Tarbell Format at 1500 bits per second plus a sync stream. These programs contain all the test routines described in the User Guide, as well as software handlers for sync generation, block formation, and CRC generation and checking.

The cassette programs are origined to run starting at location 3ØØØ Hex, and they initialize the stack pointer at 36ØØ Hex. Consequently, 1½ K (1536 bytes) of RAM, starting at 3ØØØ, are required to support it.

The test cassette was designed as an aid in debugging and testing the operation of the various ports. The operation of the various functions are described individually in the sections of the User Guide devoted to those ports. It is suggested that, in bringing up an MIO board for the first time, that the CRI interface be tested first. With an operating CRI, the other functions may be tested conveniently by loading the test routines into the computer from the test cassette.

```
                    ;********        MIO TEST CASSETTE LOADER        ********
                    ;
                    ; I/O PARAMETERS
                    ;
0040 =              CRI     EQU     40H     ;CASSETTE PORT
0043 =              CRL     EQU     43H     ;CONTROL PORT
0004 =              CRY     EQU     04H     ;CASSETTE READY BIT
                    ;
3800                        ORG     3800H
3800 310040                 LXI     SP,4000H
3803 3E60                   MVI     A,60H   ;SET TO READ BY BIT...
3805 D343                   OUT     CRL
3807 CD2A38         SYNC:   CALL    CASIN   ;READ 8 BITS
380A FEE6                   CPI     0E6H    ;IS IT SYNC YET?
380C C20738                 JNZ     SYNC    ;WAIT TILL IT IS
380F 3E20                   MVI     A,20H   ;SET TO READ BY BYTE...
3811 D343                   OUT     CRL
3813 11B203                 LXI     D,3B2H  ;INIT COUNT
3816 210030                 LXI     H,3000H ;GET START LOAD ADDRESS
3819 CD2A38         READ:   CALL    CASIN   ;READ A BYTE
381C 77                     MOV     M,A     ;STASH IT...
381D 23                     INX     H
381E 1B                     DCX     D       ;COUNT DOWN
381F 7A                     MOV     A,D     ;IS COUNT 0?...
3820 B3                     ORA     E
3821 C21938                 JNZ     READ    ;CHECK ALL BYTES
3824 2F                     CMA             ;CLEAR LIGHTS...
3825 D3FF                   OUT     0FFH
3827 C32738         HANG:   JMP     HANG    ;HANG HERE
                    ;
382A DB43           CASIN:  IN      CRL     ;WAIT TILL DATA AVAILABLE...
382C E604                   ANI     CRY
382E CA2A38                 JZ      CASIN
3831 DB40                   IN      CRI     ;READ 8 BITS
3833 C9                     RET
                    ;
3834                        END
```

APPENDIX B

MIOA LISTING

APPENDIX B

```
                  ;MIO BOARD CRI INITIALIZATION PROGRAMS
                  ;ADDRESS DEFINITIONS FOR MIO BOARD CONFIGURED
                  ;AS DEFINED IN MIO USER GUIDE - SECTION I.2
0042 =            SIO     EQU 42H
0041 =            PIO     EQU 41H
0043 =            CNT     EQU 43H
0040 =            CRI     EQU 40H
00FF =            SSPT    EQU 0FFH        ;SENSE LIGHTS AND SWITCHES
3100 =            BASA    EQU 3100H
3000 =            BASB    EQU 3000H
3600 =            BUFR    EQU 3600H
3600 =            STACK   EQU 3600H
3100                      ORG BASA
                  ;JUMP TABLE FOR ENTRY TO MIO TESTS
3100 C31831               JMP SIO1
3103 C33031               JMP SIO2
3106 C34531               JMP SIO3
3109 C31732               JMP PIO1
310C C33D32               JMP PIO2
310F C34232               JMP PIO3
3112 C34732               JMP CRIWT
3115 C36732               JMP CRIRT
                  ;SIO TEST 1     OUTPUT THE VALUE CONTAINED IN THE
                  ;               SENSE SWITCHES TO THE SIO PORT.  IF AN
                  ;               INPUT CHARACTER IS READY AND NO INPUT
                  ;               ERRORS OCCUR DISPLAY THE CHARACTER IN
                  ;               IN THE SENSE LIGHTS.  IF AN INPUT ERROR
                  ;               OCCURS, DISPLAY ALL ONES.  PAUSE 15
                  ;               SECONDS EACH TIME THE SWITCHES ARE CHANGED.
3118 310036       SIO1:   LXI SP,STACK
311B AF                   XRA A   ;SET UP CONTROL REG
311C D343                 OUT CNT
311E CDE231       SIO11:  CALL SSIN       ;GET SENSE SWITCHES
3121 CD9E31               CALL SOUT       ;OUTPUT CHAR
3124 CDAA31               CALL SINP       ;TEST INPUT
3127 CA1831               JZ SIO1         ;IF NO INPUT READY
312A 2F                   CMA
312B D3FF                 OUT SSPT        ;OUTPUT CHAR OR ERROR FLAG
312D C31E31               JMP SIO11
                  ;SIO TEST 2     READ INPUT CHARACTERS FROM SIO DEVICE
                  ;               IF CHARACTER IS READ WITHOUT ERROR,
                  ;               OUTPUT CHARACTER TO SIO DEVICE.  IF AN
                  ;               ERROR OCCURS,IGNORE CHARACTER
3130 310036       SIO2:   LXI SP,STACK
3133 AF                   XRA A   ;SET CONTROL REG
3134 D343                 OUT CNT
3136 CDAA31       SIO21:  CALL SINP       ;GET CHAR
3139 CA3631               JZ SIO21        ;NONE READY
313C FA3631               JM SIO21        ;ERROR ON INPUT
313F CD9E31               CALL SOUT       ;OUTPUT VALID CHAR
3142 C33631               JMP SIO21
                  ;SIO TEST 3     CONTINUOUSLY TRANSMIT ALL POSSIBLE BIT
                  ;               PATTERS MASKED WITH THE COMPLEMENT OF THE
                  ;               SENSE SWITCHES.  CHECK FOR RECEIVE ERRORS
                  ;               AND DISPLAY 0FFH IF ANY OCCUR FOLLOWED BY
                  ;               STATUS WITH PE,OE,FE,RRDY AND TRDY IN BITS
```

```
                  ;              4 TO 0 RESPECTIVELY.  COMPARE RECEIVED CHAR
                  ;              WITH TRANSMITTED CHAR.  DISPLAY 0FEH IF DIFFERENT
                  ;              FOLLOWED BY TRANSMITTED CHAR AND
                  ;              RECEIVED CHAR.  IN NORMAL OPERATION DISPLAY
                  ;              TRANSMITTED CHAR.
3145 310036       SIO3:   LXI SP,STACK
3148 AF                   XRA A           ;SET CONTROL
3149 D343                 OUT CNT
314B 0E00                 MVI C,0         ;ORIGINAL CHAR VALUE
314D DBFF         SIO31:  IN SSPT         ;GET ORIGINAL SENSE SWITCH
314F 32FA31               STA SSAV
3152 2F                   CMA             ;FORM CHAR
3153 A1                   ANA C
3154 0C                   INR C           ;SET NEXT VALUE
3155 57                   MOV D,A         ;SAVE IT FOR COMPARE
3156 2F                   CMA             ;FOR PROPER LIGHTS
3157 D3FF                 OUT SSPT        ;DISPLAY IT
3159 2F                   CMA             ;FOR PROPER VALUE
315A CD9E31               CALL SOUT       ;OUTPUT IT
315D CDAA31       SIO32:  CALL SINP       ;TEST INPUT
3160 CA5D31               JZ SIO32        ;IF NONE READY
3163 FA8131               JM SIO33        ;ON ERROR
3166 5F                   MOV E,A         ;MASK INPUT
3167 3AFA31               LDA SSAV
316A 2F                   CMA
316B A3                   ANA E
316C BA                   CMP D           ;COMPARE WITH OUTPUT
316D CA4D31               JZ SIO31        ;RELOOP IF OK
3170 5F                   MOV E,A
3171 3EFE                 MVI A,0FEH      ;ERROR FLAG
3173 CD8E31               CALL DISP       ;DISPLAY TILL SENSE SWITCHES CHANGE
3176 7A                   MOV A,D         ;TRANS CHAR
3177 CD8E31               CALL DISP
317A 7B                   MOV A,E         ;RECEIVED CHAR
317B CD8E31               CALL DISP
317E C34D31               JMP SIO31
3181 57           SIO33:  MOV D,A         ;SAVE ERRORS
3182 3EFE                 MVI A,0FEH      ;ERROR FLAG
3184 CD8E31               CALL DISP
3187 7A                   MOV A,D         ;STATUS RESULTS
3188 CD8E31               CALL DISP
318B C34D31               JMP SIO31
                  ;GENERAL UTILITY ROUTINES FOR SIO TEST.
                  ;THIS ROUTINE DISPLAYS THE VALUE IN A UNTIL
                  ;SENSE SWITCHES ARE CHANGED.
318E 2F           DISP:   CMA             ;FOR PROPER LIGHTS
318F D3FF                 OUT SSPT
3191 DBFF                 IN SSPT         ;INITIAL SENSE SWITCHES
3193 47                   MOV B,A
3194 CDFB31               CALL DLA5       ;WAIT A WHILE
3197 DBFF         DIS1:   IN SSPT         ;NEW VALUE?
3199 A8                   XRA B
319A CA9731               JZ DIS1 ;WAIT FOR DIFFERENCE
319D C9                   RET
                  ;OUTPUT CHARACTER IN A WHEN DEVICE READY.
319E 47           SOUT:   MOV B,A         ;WAIT TIL READY
319F DB43         SOUT1:  IN CNT
31A1 E601                 ANI 1
```

```
31A3 CA9F31          JZ SOUT1
31A6 78              MOV A,B
31A7 D342            OUT SIO         ;CHAR OUT
31A9 C9              RET
             ;INPUT A CHAR WHEN READY.  IF AN ERROR
             ;OCCURS, PUT PE,OE,FE,RRDY,TRDY IN 4 TO 0.
31AA DB43    SINP:   IN CNT          ;SEE IF READY ON ERROR
31AC E60A            ANI 0AH
31AE C8              RZ
31AF EE0A            XRI 0AH         ;YES, TEST ERROR
31B1 CABA31          JZ SIN1
31B4 EE02            XRI 2           ;SEE IF OLD ERROR FLAG
31B6 C8              RZ              ;IF SO,RETURN
31B7 DB42            IN SIO          ;NO ERROR, GET CHAR
31B9 C9              RET
31BA 3E80    SIN1:   MVI A,80H       ;GET ERROR BITS
31BC D343            OUT CNT         ;PARITY ERROR
31BE DB43            IN CNT
31C0 E608            ANI 8
31C2 07              RLC
31C3 47              MOV B,A
31C4 3EC0            MVI A,0C0H      ;FRAMING ERROR
31C6 D343            OUT CNT
31C8 DB43            IN CNT
31CA E608            ANI 8
31CC 0F              RRC
31CD 80              ADD B
31CE 47              MOV B,A
31CF 3E40            MVI A,40H       ;OVERUN,RRDY AND TRDY
31D1 D343            OUT CNT
31D3 DB43            IN CNT
31D5 E60B            ANI 0BH
31D7 80              ADD B
31D8 47              MOV B,A
31D9 DB42            IN SIO          ;CLEAR CHARACTER
31DB AF              XRA A   ;RESET CONTROL FOR ERROR FLAG
31DC D343            OUT CNT
31DE F680            ORI 80H
31E0 78              MOV A,B
31E1 C9              RET
             ;INPUT SENSE SWITCHES-DELAY IF DIFFERENT
31E2 DBFF    SSIN:   IN SSPT         ;GET THEM
31E4 47              MOV B,A
31E5 3AFA31          LDA SSAV        ;COMPARE WITH PAST
31E8 A8              XRA B
31E9 78              MOV A,B
31EA C8              RZ
31EB CDFB31          CALL DLA5       ;DIFFERENT WAIT FOR A WHILE
31EE CDFB31          CALL DLA5
31F1 CDFB31          CALL DLA5
31F4 DBFF            IN SSPT         ;GET NEW VALUE
31F6 32FA31          STA SSAV
31F9 C9              RET
31FA 00      SSAV:   DB 0
             ;DELAY 5 SECONDS - REQUIRES 10 MILLION CYCLES (APPROXIMATELY)
31FB 3E00    DLA5:   MVI A,0
31FD 0EC9            MVI C,201
31FF CD0B32  DLA51:  CALL DONE
```

```
3202 0C                 INR C
3203 C2FF31             JNZ DLA51
3206 3C                 INR A
3207 C2FF31             JNZ DLA51
320A C9                 RET
320B E5         DONE:   PUSH H          ;TAKE 121 CYCLES
320C E1                 POP H
320D E5                 PUSH H
320E E1                 POP H
320F E5                 PUSH H
3210 E1                 POP H
3211 E5                 PUSH H
3212 E1                 POP H
3213 E5                 PUSH H
3214 E1                 POP H
3215 7F                 MOV A,A
3216 C9                 RET
                ;PIO TEST 1     READ SENSE SWITCHES AND OUTPUT
                ;               TO BOTH PORTS.
3217 0E01       PIO1:   MVI C,1         ;SET TEST 1 FLAG
3219 DBFF       PIO11:  IN SSPT         ;GET VALUE
321B 2F         PIO12:  CMA             ;FOR PROPER LIGHTS
321C D3FF               OUT SSPT        ;OUTPUT TO LIGHTS
321E 2F                 CMA             ;FOR PROGRAM USE
321F 47                 MOV B,A
3220 AF                 XRA A           ;SET FOR PORT 1
3221 D343               OUT CNT
3223 78                 MOV A,B
3224 D341               OUT PIO
3226 3E80               MVI A,80H       ;NOW FOR PORT 2
3228 D343               OUT CNT
322A 78                 MOV A,B
322B D341               OUT PIO
322D 0C                 INR C           ;SEE WHICH TEST IT IS
322E 0D                 DCR C
322F FA3532             JM PIO13
3232 C21932             JNZ PIO11
3235 79         PIO13:  MOV A,C         ;TEST 2 OR 3
3236 D343               OUT CNT         ;SET TO READ PROPER INPUT PORT
3238 DB41               IN PIO
323A C31B32             JMP PIO12
                ;PIO TEST 2             READ PIO PORT 1 AND OUTPUT
                ;               TO PORTS 1 AND 2 AND SENSE LIGHTS
323D 0E00       PIO2:   MVI C,0         ;FLAG FOR PORT 1 IN
323F C33532             JMP PIO13
                ;PIO TEST 3             READ PIO PORT 2 AND OUTPUT
                ;               TO PORTS 1 AND 2 AND SENSE LIGHTS.
3242 0E80       PIO3:   MVI C,80H       ;FLAG FOR PORT 2 IN
3244 C33532             JMP PIO13
                ;CRI WRITE TEST         WRITE A BLOCK OF 256 BYTES
                ;               WITH EACH BYTE CONTAINING ITS ADDRESS
                ;               WITHIN THE BLOCK.
3247 310036     CRIWT:  LXI SP,STACK
324A 210036             LXI H,BUFR      ;FILL BUFFER WITH ADDRESS
324D AF                 XRA A
324E 77         CRIW1:  MOV M,A
324F 23                 INX H
3250 3C                 INR A
```

```
3251 C24E32            JNZ CRIW1
3254 210036            LXI H,BUFR      ;SET PARAMATERS
3257 1E00              MVI E,0         ;256 BYTES
3259 3E3F              MVI A,3FH
325B D3FF              OUT SSPT        ;GIVE LIGHTS AN INITIAL VALUE
325D CD9E32            CALL WRIT       ;DO THE WRITE
3260 AF        CRIW2:  XRA A           ;ALL DONE LOOP
3261 2F                CMA             ;FOR PROPER LIGHTS
3262 D3FF              OUT SSPT
3264 C36032            JMP CRIW2
               ;CRI READ TEST          READ A BLOCK OF 256 BYTES.
               ;                 CHECK THAT EACH BYTE CONTAINS ITS ADDRESS
               ;                 WITHIN THE BLOCK.  CRC ERROR IS ALSO DETECTED BY
               ;                 READ HANDLER.
3267 310036    CRIRT:  LXI SP,STACK
326A 210036            LXI H,BUFR      ;SET PARAMATERS
326D 1E00              MVI E,0         ;256 BYTES
326F 3E3F              MVI A,3FH       ;INITIAL VALUE FOR LIGHTS
3271 D3FF              OUT SSPT
3273 CD0133            CALL READ       ;READ THE BLOCK
3276 CA7E32            JZ CRIR1        ;JUMP IF NO CRC ERROR
3279 3EFF              MVI A,0FFH      ;ELSE,DISPLAY IT
327B CD8E31            CALL DISP
327E 1E00      CRIR1:  MVI E,0         ;DO A BYTE BY BYTE COMPARE
3280 210036            LXI H,BUFR
3283 7E        CRIR3:  MOV A,M
3284 BB                CMP E           ;COMPARE A BYTE
3285 CA9532            JZ CRIR2
3288 3EFE              MVI A,0FEH      ;DISPLAY THE ERROR
328A CD8E31            CALL DISP
328D 7B                MOV A,E         ;CORRECT VALUE
328E CD8E31            CALL DISP
3291 7E                MOV A,M         ;ACTUAL VALUE
3292 CD8E31            CALL DISP
3295 23        CRIR2:  INX H
3296 1C                INR E           ;LOOP COUNT
3297 C28332            JNZ CRIR3
329A C36032            JMP CRIW2       ;IF ALL DONE
               ;GENERAL HANDLERS FOR TARBELL OR BYTE/LANCASTER ON
               ;CRI.  USE 1-255 BYTE BLOCK AND STANDARD CRC ROUTINE
               ;FOR ERROR CHECKING.
329D 00        TYPF:   DB 0      ;SET 0 FOR TARBELL, NONZERO FOR BYTE/LANCASTER
               ;WRITE ROUTINE - THE FOLLOWING PARAMATERS ARE EXPECTED ON ENTRY
               ;       HL - CONTAIN THE MEMORY LOCATION FOR THE BLOCK
               ;        E - CONTAINS THE BLOCK SIZE, 1-256. (0=256)
329E 3E10      WRIT:   MVI A,10H
32A0 D343              OUT CNT
32A2 D340              OUT CRI         ;RESET BYTE COUNTER
32A4 3E3C              MVI A,03CH      ;START BYTE
32A6 CDCB32            CALL WRBYT      ;OUTPUT IT
32A9 3EE6              MVI A,0E6H      ;SYNC BYTE
32AB CDCB32            CALL WRBYT      ;WRITE A BYTE WHEN READY
32AE 01FFFF            LXI B,0FFFFH    ;INITIALIZE CRC VALUE
32B1 7E        WRIT1:  MOV A,M         ;GET A BYTE
32B2 CD8F33            CALL CRC        ;ADD TO CRC
32B5 7E                MOV A,M         ;GET THE BYTE AGAIN
32B6 CDCB32            CALL WRBYT      ;WRITE IT WHEN READY
32B9 23                INX H
```

```
32BA 1D                 DCR E           ;LOOP COUNT
32BB C2B132             JNZ WRIT1       ;LOOP TIL DONE
32BE 78                 MOV A,B         ;WRITE CRC BYTE 1
32BF CDCB32             CALL WRBYT
32C2 79                 MOV A,C         ;BYTE 2
32C3 CDCB32             CALL WRBYT
32C6 AF                 XRA A           ;TRAILING ZERO BYTE
32C7 CDCB32             CALL WRBYT
32CA C9                 RET
32CB 57         WRBYT:  MOV D,A         ;SAVE THE BYTE
32CC DB43       WRBY2:  IN CNT          ;WAIT TIL READY
32CE E604               ANI 4
32D0 CACC32             JZ WRBY2
32D3 3A9D32             LDA TYPF        ;SEE WHICH TYPE
32D6 A7                 ANA A
32D7 C2DE32             JNZ WRBY3
32DA 7A                 MOV A,D         ;TARBELL
32DB D340               OUT CRI
32DD C9                 RET
32DE E5         WRBY3:  PUSH H          ;BYTE/LANCASTER-SERIALIZE BYTE
32DF 2608               MVI H,8
32E1 CDF432     WRBY7:  CALL WRBY5      ;WRITE A BIT
32E4 25                 DCR H           ;BIT COUNTER
32E5 C2EA32             JNZ WRBY6
32E8 E1                 POP H           ;DONE, RESTORE H
32E9 C9                 RET
32EA DB43       WRBY6:  IN CNT
32EC E604               ANI 4
32EE CAEA32             JZ WRBY6
32F1 C3E132             JMP WRBY7
32F4 7A         WRBY5:  MOV A,D
32F5 17                 RAL
32F6 57                 MOV D,A         ;CARRY HAS FIRST BIT
32F7 3EFF               MVI A,0FFH      ;FOR A ONE
32F9 DAFE32             JC WRBY4
32FC 3EAA               MVI A,0AAH      ;FOR A ZERO
32FE D340       WRBY4:  OUT CRI
3300 C9                 RET
                ;READ ROUTINE READS IN TARBELL OR BYTE/LANCASTER AS
                ;A FUNCTION OF TYPF.  INPUT PARAMATERS ARE:
                ;       HL - CONTAIN ADDRESS OF INPUT BUFFER
                ;        E - CONTAINS BLOCK SIZE, 1-256 (0=256)
                ;RETURNS WITH ZERO FLAG SET OF NO CRC ERROR OCCURS.
3301 3E60       READ:   MVI A,60H       ;SET TO RECOGNIZE SYNC
3303 D343               OUT CNT
3305 CD3233             CALL RBSN       ;SYNC ON BYTE BASIS OF BYTE/LANCASTER
3308 CD5633     READ1:  CALL GBIT       ;GET BYTE ON NEXT SHIFT
330B FEE6               CPI 0E6H        ;SEE IF SYNC
330D C20833             JNZ READ1
3310 3E20               MVI A,20H       ;OKAY, GO TO BYTE READY
3312 D343               OUT CNT
3314 01FFFF             LXI B,0FFFFH    ;SET INITIAL CRC VALUE
3317 CD6A33     READ2:  CALL GBYT       ;GET AA BYTE
331A 77                 MOV M,A ;STORE IT
331B CD8F33             CALL CRC        ;ADD TO CRC
331E 23                 INX H
331F 1D                 DCR E           ;LOOP COUNT
3320 C21733             JNZ READ2
```

```
3323 CD6A33           CALL GBYT       ;CRC BYTE 1
3326 CD8F33           CALL CRC        ;FORM VALUE
3329 CD6A33           CALL GBYT       ;BYTE 2
332C CD8F33           CALL CRC        ;FOR THE LAST TIME!
332F 79               MOV A,C         ;SET FLAG
3330 B0               ORA B
3331 C9               RET
3332 3A9D32   RBSN:   LDA TYPF        ;SYNC IF REQUIRED
3335 A7               ANA A
3336 C8               RZ
3337 1600             MVI D,0         ;SET FOR BIT SYNC RECOGNITION
3339 CD4C33   RBS2:   CALL RBS1       ;NOW WAIT FOR A ZERO BIT FOLLOWED
333C 17               RAL             ;BY EIGHT ONES SO HAVE
333D DA3933           JC RBS2         ;TARBELL BYTE SAME AS LANCASTER BIT.
3340 CD4C33           CALL RBS1       ;HAVE A ZERO LOOK FOR ONES
3343 3C               INR A
3344 C23933           JNZ RBS2
3347 3E20             MVI A,20H
3349 D343             OUT CNT         ;ALL SET, SET TO GET BYTES FROM NOW ON
334B C9               RET
334C DB43     RBS1:   IN CNT
334E E604             ANI 4
3350 CA4C33           JZ RBS1
3353 DB40             IN CRI
3355 C9               RET
3356 DB43     GBIT:   IN CNT          ;GET BYTE AFTER NEXT BIT SHIFT
3358 E604             ANI 4
335A CA5633           JZ GBIT
335D 3A9D32           LDA TYPF
3360 A7               ANA A
3361 DB40             IN CRI
3363 C8               RZ              ;RETURN ON TARBELL
3364 C601             ADI 1           ;CONVERT TO 1 OR 0 BIT
3366 7A               MOV A,D
3367 17               RAL             ;ADD TO BYTE
3368 57               MOV D,A
3369 C9               RET
336A DB43     GBYT:   IN CNT          ;WAIT TIL READY
336C E604             ANI 4
336E CA6A33           JZ GBYT
3371 3A9D32           LDA TYPF        ;CHECK MODE
3374 A7               ANA A
3375 C27B33           JNZ GBYT1
3378 DB40             IN CRI          ;TARBELL, JUST READ BYTE
337A C9               RET
337B E5       GBYT1:  PUSH H          ;LANCASTER NEED TO ASSEMBLE A BYTE
337C 2607             MVI H,7
337E DB40             IN CRI          ;FIRST BIT
3380 57               MOV D,A
3381 CD4C33   GBYT2:  CALL RBS1       ;GET NEXT TARBELL BYTE=LANCASTER BIT
3384 C601             ADI 1
3386 7A               MOV A,D
3387 17               RAL
3388 57               MOV D,A ;ADD TO BYTE
3389 25               DCR H
338A C28133           JNZ GBYT2
338D E1               POP H
338E C9               RET
```

```
                    ;GENERAL CRC ROUTINE.  COMPUTE FOR ONE BYTE
338F E5             CRC:    PUSH H
3390 D5                     PUSH D
3391 A8                     XRA B
3392 67                     MOV H,A
3393 07                     RLC
3394 07                     RLC
3395 07                     RLC
3396 07                     RLC
3397 AC                     XRA H
3398 6F                     MOV L,A
3399 E6F0                   ANI 0F0H
339B 57                     MOV D,A
339C 85                     ADD L
339D 5F                     MOV E,A
339E 7A                     MOV A,D
339F CE00                   ACI 0
33A1 A9                     XRA C
33A2 47                     MOV B,A
33A3 7C                     MOV A,H
33A4 E6F0                   ANI 0F0H
33A6 67                     MOV H,A
33A7 AB                     XRA E
33A8 4F                     MOV C,A
33A9 7C                     MOV A,H
33AA 0F                     RRC
33AB 0F                     RRC
33AC 0F                     RRC
33AD A8                     XRA B
33AE 47                     MOV B,A
33AF D1                     POP D
33B0 E1                     POP H
33B1 C9                     RET
33B2                        END
```

APPENDIX C

MIOB LISTING

APPENDIX C

```
                ;MIO BOARD CRI INITIALIZATION PROGRAMS
                ;ADDRESS DEFINITIONS FOR MIO BOARD CONFIGURED
                ;AS DEFINED IN MIO USER GUIDE - SECTION I.2
0042 =          SIO     EQU 42H
0041 =          PIO     EQU 41H
0043 =          CNT     EQU 43H
0040 =          CRI     EQU 40H
00FF =          SSPT    EQU 0FFH        ;SENSE LIGHTS AND SWITCHES
3100 =          BASA    EQU 3100H
3000 =          BASB    EQU 3000H
3600 =          BUFR    EQU 3600H
3600 =          STACK   EQU 3600H
                ;SYNC RECOGNITION PROGRAM - FINDS INITIAL SYNC
                ;AND THEN SETS ALL SENSE LIGHTS FOR EACH SYNC
                ;BYTE THEREAFTER.  IF A SYNC BYTE IS MISSED SETS SENSE
                ;SENSE SWITCHES TO ZERO AND LOOKS FOR SYNC AGAIN.
3000                    ORG BASB
3000 3E60       SYNR:   MVI A,60H
3002 D343               OUT CNT         ;ENABLE READ AND READY BY BIT
3004 AF                 XRA A
3005 2F                 CMA             ;FOR PROPER LIGHTS
3006 D3FF               OUT SSPT        ;CLEAR LIGHTS
3008 DB43       SYNR1:  IN CNT          ;WAIT FOR READY
300A E604               ANI 4
300C CA0830             JZ SYNR1
300F DB40               IN CRI          ;SEE IF SYNC BYTE
3011 D6E6               SUI 0E6H
3013 C20030             JNZ SYNR        ;IF NOT, RELOOP
3016 D3FF               OUT SSPT        ;YES SET LIGHTS TO ONES
3018 3E20               MVI A,20H       ;SET TO READ BYTES
301A D343               OUT CNT
301C C30830             JMP SYNR1       ;GO LOOK AT NEXT BYTE
                ;SYNC GENERATION PROGRAM - WRITES SYNC BYTE CONTINUOUSLY
301F 3E10       SYNG:   MVI A,10H       ;SET WRITE ENABLE
3021 D343               OUT CNT
3023 3EE6       SYNG2:  MVI A,0E6H      ;OUTPUT SYNC CHAR
3025 D340               OUT CRI
3027 DB43       SYNG1:  IN CNT          ;WAIT TIL READY AGAIN
3029 E604               ANI 4
302B CA2730             JZ SYNG1
302E C32330             JMP SYNG2       ;THEN DO ANOTHER
                ;BOOTSTRAP PROGRAM FOR TARBELL CODE
3031 210031     BOOT:   LXI H,BASA      ;GET STARTING ADDRESS
3034 3E60               MVI A,60H       ;SET READ AND READY BY BIT
3036 D343               OUT CNT
3038 DB43       BOOT1:  IN CNT          ;LOOK FOR SYNC CHAR
303A E604               ANI 4
303C CA3830             JZ BOOT1
303F DB40               IN CRI          ;GET CHAR
3041 FEE6               CPI 0E6H
3043 C23830             JNZ BOOT1
3046 3E20               MVI A,20H       ;GO TO BYTE
3048 D343               OUT CNT
304A DB43       BOOT2:  IN CNT          ;WAIT FOR BYTE
304C E604               ANI 4
304E CA4A30             JZ BOOT2
```

```
3051 DB40              IN CRI          ;GET IT
3053 77                MOV M,A ;STORE IT
3054 23                INX H
3055 C34A30            JMP BOOT2       ;GET NEXT BYTE
               ;DUMP PROGRAM FOR FORMING TAPE FOR LATER REBOOT
3058 210031    DUMP:   LXI H,BASA
305B 3E10              MVI A,10H
305D D343              OUT CNT         ;SET CONTROL FOR WRITE
305F D340              OUT CRI         ;TO CLEAR COUNTERS
3061 DB43      DUMP1:  IN CNT          ;WAIT UNTIL READY
3063 E604              ANI 4
3065 CA6130            JZ DUMP1
3068 3E3C              MVI A,03CH
306A D340              OUT CRI         ;WRITE START CHARACTER
306C 06E6              MVI B,0E6H      ;SYNC CHARACTER
306E DB43      DUMP2:  IN CNT          ;WAIT UNTIL READY
3070 E604              ANI 4
3072 CA6E30            JZ DUMP2
3075 78                MOV A,B         ;GET CHARACTER
3076 D340              OUT CRI         ;WRITE IT
3078 46                MOV B,M         ;NEXT CHARACTER
3079 23                INX H
307A C36E30            JMP DUMP2
307D                   END
```

APPENDIX D

DEBUGGING INFORMATION

If the problem still persists, it will be necessary to use the MIO Schematic Drawing as a guide in trouble-shooting. While it may seem very complex at first glance, it is much easier to understand once it has been broken down into FUNCTIONAL BLOCKS (e.g., Board Enable Circuits, SIO Port Circuits, CRI Port Circuits, PIO Port Circuits, Control Port Circuits, Input Receivers, Output Drivers, etc.).

The User is encouraged to familiarize him/herself with the Schematic Drawings if s/he is to do any further debugging.

1. The first step in debugging is to narrow down the problem as specifically as possible. EXAMPLE: If the SIO Test fails, does it fail in Transmit or Receive mode? If it fails only in Transmit mode, does it fail for all characters or just one in particular? How does it fail? Is a bit being dropped, or is any input being received at all? Etc.

   TRY TO WORK ON ONE SPECIFIC PROBLEM AT A TIME.

2. Armed with this information, the User should use the following reference sources to understand the logic flow for the operation which fails (determined in Step 1 above).

   1. Schematic Drawing:
   2. Theory of Operation Chapter;
   3. A reference such as the <u>TTL DATA BOOK</u>; and
   4. Test Program Listings.

3. Follow the Logic Flow, determined in Step 2, above, by checking circuit points with an Oscilloscope or Logic Probe. It is usually easier to start checking at the logical end-point and work back towards the source.

   You will be looking for:

   1. incorrect signal levels;
   2. missing signals;
   3. incorrect voltage levels of signals; and
   4. signals which occur at the wrong time.

   Once an inconsistency or problen has been located, trace back towards the source of the signal to locate the source of the problem.

The problem can usually be traced to:

1. a defective chip;
2. a solder cross or bad solder joint or
3. a misplaced or incorrectly oriented component.

A BRIEF LIST OF PROBLEMS WITH SUGGESTED POINTS TO CHECK ARE GIVEN BELOW.

NONE OF THE PORTS RESPOND

1. Check the jumpering of the EXTERNAL ADDRESS JUMPER AREA.

2. Check the Board Enable Circuits. U18-8 goes low when the processor executes an output instruction to the MIO Board Addrsss.

ONE OF THE PORTS DOES NOT RESPOND

1. Check the jumpering of the Internal Address Jumper Area.

2. Check U23: The outputs of U23 are the Internal Port Select Signals. There are four Register Load Signals and four Read Enables.

NO INPUT FROM ANY INPUT PORTS

1. Check the Input Bus Drivers U42 and U43. Check for Enables U43-1, U43-15, and U42-15, going low.

NO OUTPUT TO ANY OUTPUT PORT

1. Check the Output Bus Drivers U41 and U42. Check for Enables U42-1 and U41-1 going low.

SIO PORTS

NO OUTPUT FROM SIO TO EXTERNAL DEVICE

1. Check U7-25 UART Transmit Data Line. If Data is present here, carefully check the jumpering of the OJA and/or OJA Line Drivers.

   If Data is not present, check SIO Configuration Jumpers and check all Control Inputs to the UART U7 (especially U7-23, UART Data Load).

NO INPUT FROM EXTERNAL DEVICE TO SIO

1. Check U7-20, the UART Receive Data Line. If Data is not present here, carefully check the jumpering of the IJA and/or the IJA Receivers.

   If Data is present, check the SIO Configuration Jumpers and check all Control Inputs to the UART U7 (especially U7-4, UART Read Enable).

PIO PORTS

NO INPUT FROM PIO INPUT PORTS

1. Check the STB from the external device. It should set /INT low.

2. Check the jumpering of the PIO Strobe Select.

3. Check the jumpering of the IJA.

4. Check the PIO Port enable (/DS1)(DS2). It is active when the Processor reads the Port. /INT should be reset to a high at this time.

NO OUTPUT TO PIO OUTPUT PORTS

1. Check the OCDR Line from the external device. It should set /INT Low.

2. Check the jumpering of the IJA.

3. Check the Port Enable (/DS1)(DS2). It is active when the Processor accesses the Port. /INT should be reset to a high at this time.

CRI PORT

NO INPUT FROM CRI

1. Check the settings of the recorder and refer to the CRI Initialization Procedures in the User Guide.

2. Check the jumpering of the IJA for CRIS.

3. Check the CRI Rate Jumpers. Refer to the User Guide.

4. Check to insure that Input Data appears at U25-2. If Data appears, check the operation of the shift registers at U24 and U25. If no Data appears, check the zero crossing detector at U34.

   Refer to the CRI Theory of Operation for further timing problems in this area.

5. Check the setting of U19-7 (Read Phase Jumper).

Table 14

TEST PROGRAM ADDRESSING AND CONTROL

TEST	ENTRY IN HEX	SENSE SWITCHES CONTROL	SENSE LIGHTS DISPLAY
SIO 1	31ØØ	Output Character	Input Character
SIO 2	31Ø3	----------	-----------
SIO 3	31Ø6	Transmit Bit Mask	Error Code
PIO 1	31Ø9	Output Character	Output Character
PIO 2	31ØC	Output Character	Output Character
PIO 3	31ØF	Output Character	Output Character
CRI Write	3112	----------	Error Code
CRI Read	3115	Sense Light Display	Error Code

Figure 9

Jumper Settings for Test Programs

Address Selection (II)

External: Jumper 2 address 40H to 43H
Internal: Jumpers 1 and 6

Input Jumper Area (III.1)

Interrupts are not used.
Data input as follows:

Bit 7 - REIA2
Bit 6 - REIA3
Bit 5 - REIA4
Bit 4 - PIOS
Bit 3 - SIOS
Bit 2 - CRIS
Bit 1 - RRDY
Bit 0 - TRDY

Output Jumper Area (III.2)

CRO - DEIA2
CR1 - DEIA3
CR2 - DEIA4

Parallel IO Port Input Strobe (V)

PIO1 - No jumper
PIO2 - No jumper

SIO Configuation Jumper Area (IV.)

No jumper

SIO Baud Jumper Area (IV.2)

Jumpered for 1200 Baud

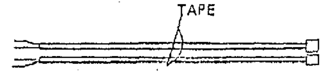
Cassette Recorder Interface

Read and Write Phase VI - as determined by initial procedure
Bit rate (VI.I) - set for 1500 bits/second

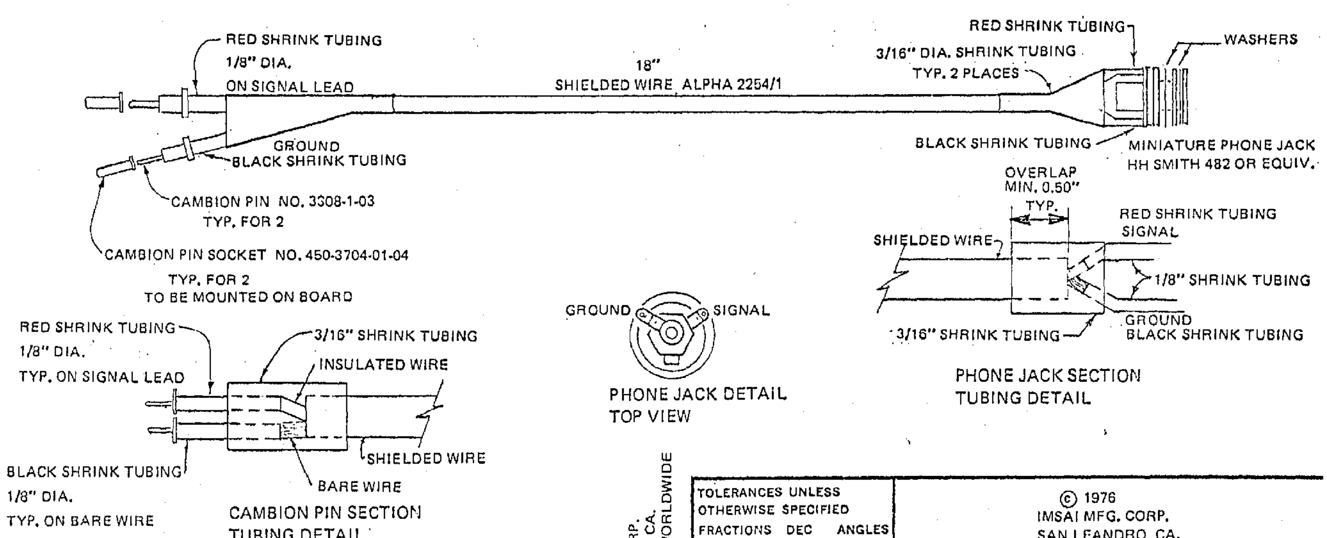
The above configuration provides the user with all the status information required to run a full RS-232-C EIA interface, a cassette recorder and two parallel input/output ports under program control.

REVISIONS

LTR	DESCRIPTION	DATE	APPR
0	ORIGINAL	12/76	ajK
1	MOD. SHRINK TUBING SPECS. ECN 76–0067	12/76	
2	ADDITION JACK DETAIL MISC. ADDT'L CALLOUTS FOR CLARIFICATION ECN 77–0007	1/77	MSJ



CABLE M IS A PAIR OF
MIO CABLES TAPED TOGETHER



CAMBION PIN SECTION
TUBING DETAIL

PHONE JACK DETAIL
TOP VIEW

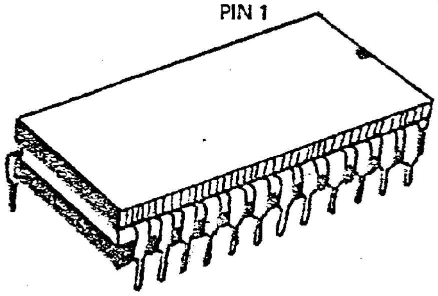
PHONE JACK SECTION
TUBING DETAIL

TOLERANCES UNLESS OTHERWISE SPECIFIED			© 1976 IMSAI MFG. CORP. SAN LEANDRO, CA.		
FRACTIONS ±	DEC ±	ANGLES ±	MIO CABLE M ASSEMBLY		91-0400017
APPROVALS	DATE				
DRAWN PRU	10/15/76				
CHECKED	12/6/76		SCALE	SIZE	DRAWING NO. 91–1040017
	12/7/76		DO NOT SCALE DRAWING		SHEET 1 O

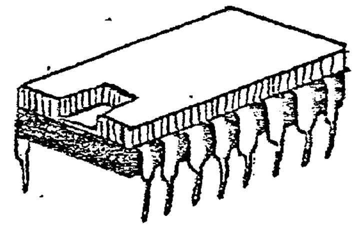
APPENDIX E

COMPONENT ILLUSTRATIONS

PIN 1



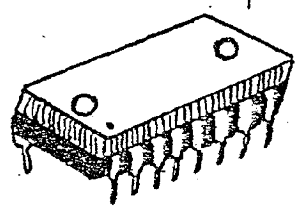
24 PIN I.C. 8212



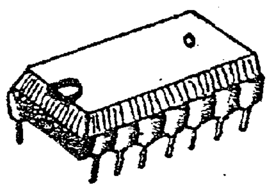
PIN 1

16 PIN I.C.

74LS123 (or 74123)	74LS175
74LS153	74LS293
74LS155	74367
74LS163 (or 74LS161)	74LS395
	8T20



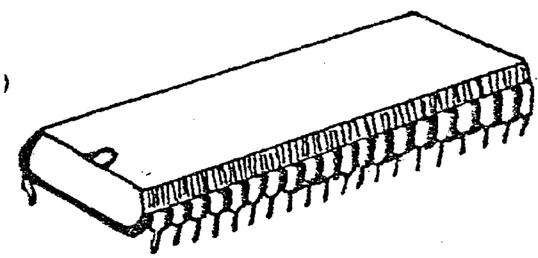
PIN 1



PIN 1

14 PIN I.C.

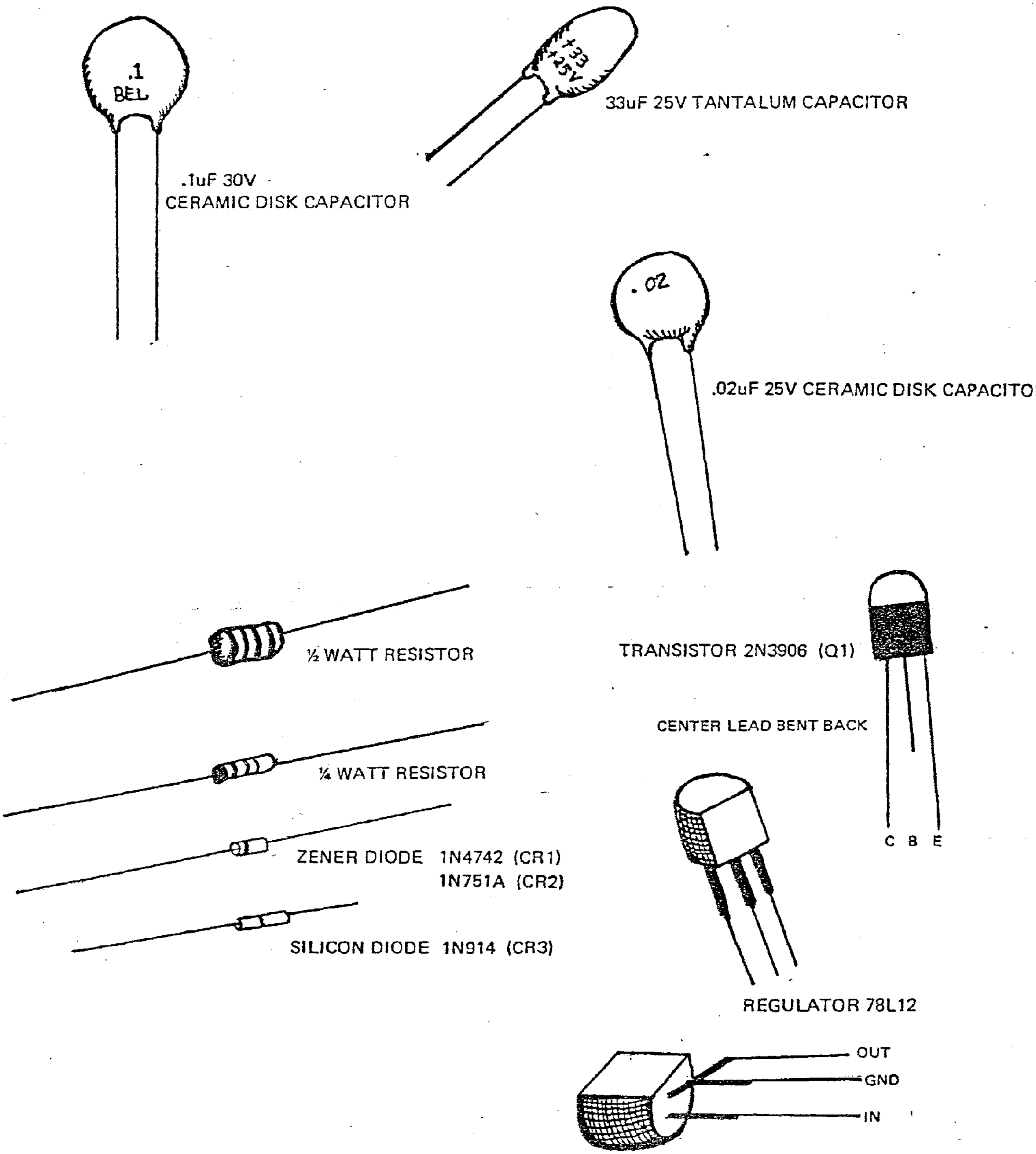
74LS00	74LS32	74LS123 (or 74123)
74LS04	74LS51	74LS153
74LS05	74LS74	74LS155
7406	74LS86	74LS163 (or 74LS161)
74LS30	75188	
7432	75189	



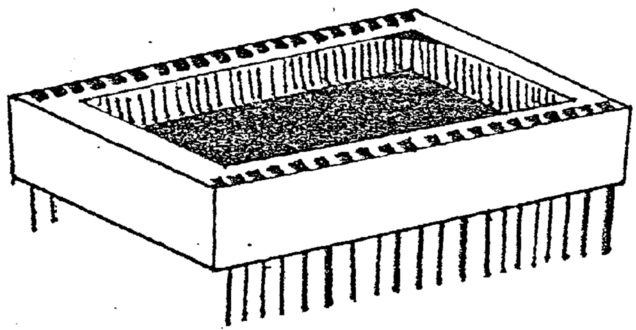
PIN 1

40 PIN UART

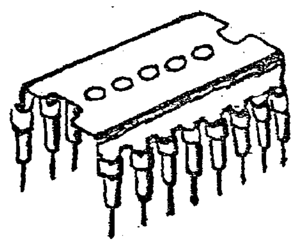
INTEGRATED CIRCUITS/CHIPS



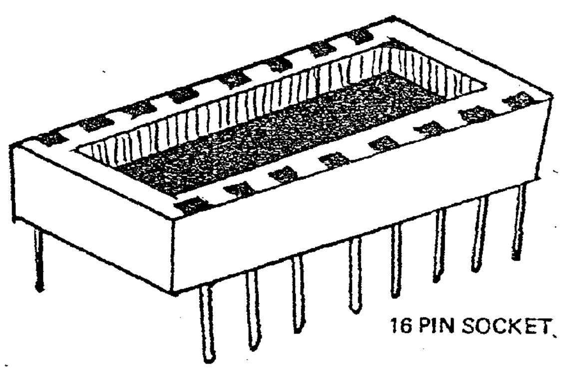
DISCRETE COMPONENTS



40 PIN SOCKET

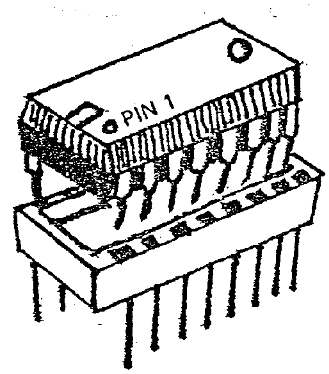


LEAD CARRIER SOCKET

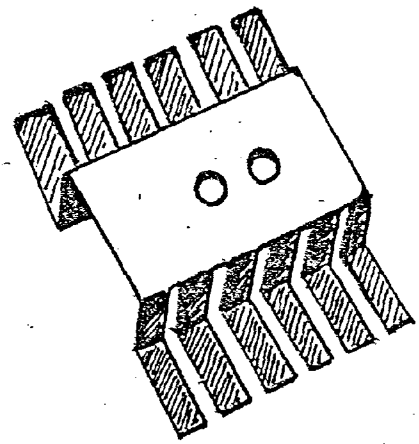


16 PIN SOCKET

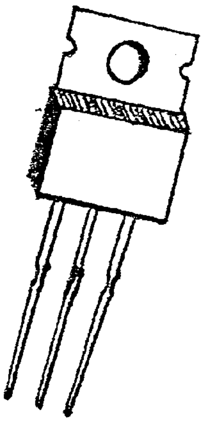
SOCKETS



I.C. INSTALLATION
INTO SOCKET



6 PRONG
HEAT SINK



7805 5V POSITIVE VOLTAGE REGULATOR

HEAT SINK & REGULATOR

IMSAI

SCS

IMSAI 8080
Self-Contained System
Acknowledgement
Revision 2

The IMSAI 8080 Monitor, Assembler, and Text Editor, supplied by IMSAI Manufacturing Corporation free of charge, is a modified version of software written by Microtec of Sunnyvale, California for Processor Technology of Berkeley, California who distributed the package free of charge.

IMSAI 8080 SELF-CONTAINED SYSTEM

OPERATING SYSTEM

The IMSAI 8080 Self-Contained System is a software system designed to run on the IMSAI 8080 computer. Included in the package is an Executive to handle memory files, an Assembler, and a line oriented Editor.

To use the system 6K,of memory must be available for use by the system. This memory is allocated as follows:

ØØ4Ø - ØDAB	Operating Program
1ØØØ - 1119	Special System RAM
111A - 17FF	Symbol Table (Assembler Only)

In addition, other memory must be available for source and object files necessary for the user's program.

I/O within the program interacts with I/O ports addressed as follows:

PORT	FUNCTION
2	TTY Data
3	TTY Status Bit Ø indicates TBE Bit 1 indicates DAV
FF	Sense Switch Input ADDRESS - PROGRAMMED INPUT switch seven is used to control file listing.

IMSAI 8080
Self-Contained System
Operating System
Revision 2

Executive Commands

CONTROL-X	Kill current line
ENTR	Enter data to memory
DUMP	Display memory data
FILE	Create, assign or display file information
EXEC	Execute a program
ASSM	Assemble a source file to object code
LIST	List file
DELT	Delete lines of file
1111	Any four numeric digits enters editor
PAGE	Move a page of data
BREK	Set or clear break points
PROC	Proceed from break point
CUST	Optional user command at location 2ØØØ

To initialize the system, start it at ØØØØ. To restart the system without initializing it, start at ØØØ3.

The executive has one error message ....WHAT?.... indicating an improper command or an error on parameters following the command.

Command Format

ENTR AAAA --- Enter data to memory

This command is used to enter data to memory starting at address AAAA and continuing until a slash (/) followed by a carriage return is entered. Data is entered in hexadecimal format.

<u>Example</u>:

```
ENTR 500
0 0A 30 44 FF FE/ (cr)
```

DUMP AAAA BBBB --- Dump contents of memory

This command is used to examine the contents of memory. The values contained in memory from locations AAAA to BBBB are displayed in hexadecimal. Each line of display consists of the contents of up to 16 memory locations. If BBBB is not specified, only locations AAAA will be displayed.

FILE /NAME/ AAAA

This command is used to enter, examine or modify parameters of files created in the system. Up to six files can exist simultaneously with any one of the files "current". Depending on the form of the command, the following parameters the following functions are performed.

FILE /NAME/ AAAA	Create a file with the name, NAME starting at address AAAA and make it current. If a file with the same name already exists, output error message NO NO.
FILE /NAME/ O	Delete file with name NAME and make no file current. Note: No file can start at location 0.
FILE /NAME/	Get file NAME and make it current. Save all parameters of existing current file.
FILE	Display parameters of the "current" file in the following format with AAAA and BBBB being the beginning of file and end of file addresses: NAME AAAA BBBB
FILES	Display the parameters of all files currently saved by the system.

EXEC AAAA----Execute a program.

This command is used to execute a program at address AAAA.

LIST N----List file

This command is used to display the lines entered by the user into the file. The output consists of the lines in the file starting at line number N. If N is not specified, the display starts at the beginning of the file. The user can terminate the display by raising ADDRESS-PROGRAMMED INPUT switch 7.

DELT L1 L2----Delete line(s) from file

This command is used to delete lines entered by the user from the file. All lines starting at line L1 and continuing up to and including L2 are deleted from the file. If L2 is not specified, only L1 is deleted.

PAGE AAAA BBBB----Move page of data

This command is used to move one page (256 bytes) of data from address AAAA to BBBB.

CUST----Optional user command at location 2000

This command allows any routine to be placed at location 2000 by the user. If the command is terminated by a RET and proper stack operations are used, the system will return in an orderly manner.

BREK or BREK AAAA

This command is used to set or clear break points. If called without the argument AAAA, all break points are cleared.

If called with the argument AAAA, a break point is set at location AAAA. When the break point is encountered in the course of execution, the break point is cleared, all registers are saved, the A register is displayed in the PROGRAMMED OUTPUT on the front panel, the message "AAAA BREAK" is typed and control returns to the executive. The registers are saved in the following locations, and may be examined or modified using the DUMP or ENTR commands.

Location	Register
1000	PSW
1001	A
1002	C
1003	B
1004	E
1005	D
1006	SP (low)
1007	SP (high)
1008	L
1009	H
100A	PC (low)
100B	PC (high)

Restrictions: (1) A maximum of 8 break points may be set.

(2) Break points may not be set below location 000B.

(3) Setting a break point causes information to be stored into locations 0008-000A, destroying any information already there.

PROC or PROC AAAA

This command is used to proceed from a break point. All registers are restored from the locations specified above, and execution continues from the location specified by the PC, unless the argument AAAA is given, in which case execution begins at location AAAA.

ASSM AAAA BBBB --- Assemble a source file to object code.

This command is used to assemble a source program written by the user and located in the file area. The assembler performs the assembly, assigning addresses to the object code starting at AAAA. On the second pass the object code is placed in memory starting at location BBBB. If BBBB is not specified, it assumes the same value as AAAA. During pass one certain errors are displayed, and during pass two a complete listing is produced.

ASSME AAAA BBBB --- Assemble and list errors only.

This command is the same as ASSM, except that only lines with errors are displayed. Object code is produced just as in ASSM.

TEXT EDITOR

Editor

The editor is a line oriented editor which enables the user to easily create program files in the system. Each line is prefaced by a fixed line number which provides for stable line referencing. Since line numbers can range from 0000 to 9999 (decimal), up to 10,000 lines can exist in each file. As the user types lines on the input device, they are entered into the file area. The editor places all line numbers in sequence and automatically over-writes an existing line in the file, if a new line with the same line number is entered by the user. A feature of the editor is that the file area never contains any wasted space.

Note: The Editor ALWAYS operates on the current file.

The editor does not automatically assign line numbers. The user must first, when entering a line of data, enter a decimal number which will be interpreted as being the line number. Valid line numbers must contain four digits; preceding zeros must be included. An entry to the editor is terminated by the carriate return key. No more than 80 characters may be input for one line.

All lines are ordered by the ascending numeric sequence of their line numbers. If the user wishes to insert lines after the initial entry is made, it is suggested that s/he input the original lines with line numbers at least five units apart.

ASSEMBLER

When the Assembler is given control by the executive, it proceeds to translate the Symbolic 8080 Assembly Language (Source) program into 8080 machine (object) code. The Assembler is a two pass assembler which operates on the "current" file. Features of the Assembler include:

- free format source input.
- symbolic addressing, including forward references and relative symbolic references.
- complex expressions may be used as arguments.
- self defining constants.
- multiple constant forms.
- up to 256 five character symbols.
- reserved names for 8080 registers
- ASCII character code generation
- 6 Pseudo Operations (assembler directives)

The assembler translates those lines contained in the current file into object code. The second character following the line number is considered to be the first source code character position. Hence, the character immediately following the line number should normally be a space. Line numbers are not processed by the assembler; they are merely reproduced on the listing.

The assembler will assemble a source program file composed of STATEMENTS, COMMENTS, and PSEUDO OPERATIONS.

During Pass 1, the assembler allocates all storage necessary for the translated program and defines the values of all symbols used, by creating a symbol table. The storage allocated for the object code will begin at the byte indicated by the 1st parameter in the original Executive ASSM command.

During Pass 2, all expressions, symbols and ASCII constants are evaluated to absolute values and are placed in allocated memory in the appropriate locations. The listing, also produced during Pass 2, indicates exactly what data is in each location of memory.

Statements

Statements may contain either symbolic 8080 machine instructions or pseudo-ops. The structure of such a statement is:

NAME OPERATION OPERAND COMMENT

The name-field, if present, must begin in assembler character position one. The symbol in the name field can contain as many characters as the user wants; however, only the first 5 characters are used in the symbol table to uniquely define a symbol. All symbols in this field must begin with an alphabetic character and may contain no special characters.

The operation field contains either a 8080 operation mnemonic or a system pseudo-operation code.

The operand field contains parameters pertaining to the operation in the operation field. If two arguments are present, they must be separated by a comma. Example:

```
0015 FLOP  MOV M,B   COMMENT
0020 * COMMENT
0025       JMP  BEG
0030       CALL FLOP
0035 BEG   ADI  8+6-4
0040       MOV  A,B
```

All fields are separated and distinguished from one another by the presence of one or more spaces or tabs.

The comment field is for explanatory remarks. It is reproduced on the listing without processing. See example 0015. Comment lines must start with an asterisk (*) in character position 1. See example 0020.

Symbolic Names

To assign a symbolic name to a statement, one merely places the symbol in the name field. To leave off the name field, the user skips two or more spaces after the line number and begins the operation field. If a name is attached to a statement, the assembler assigns it the value of the current Location Counter. The Location Counter always holds the address of the next byte to be assembled. The only exception to this is the EQU pseudo-op. In this case a symbol in the name field is assigned a value which is contained in the operand field of the EQU pseudo-of statement.

Example:

0057 POTTS EQU 128

assigns the value 128 to the name POTTS. This data can then be used elsewhere in the program, as in ADI POTTS.

Names are defined when they appear in the name field. All defined names may be used as symbolic arguments in the argument field. See examples 0015, 0025, 0030 and 0035.

In addition to user defined names, the assembler has reserved several symbols, the value of which is predetermined. These names may not be used by the user except in the operand field. They are (with their value in parenthesis):

A - the accumulator	(7)
B - Register B	(0)
C - Register C	(1)
D - Register D	(2)
E - Register E	(3)
H - Register H	(4)
L - Register L	(5)
M - Memory (through H,L)	(6)
P - Program Status Word	(6)
S - Stack Pointer	(6)

In addition to the above reserved symbols, there is the single special character symbol ($). This symbol changes in value as the assembly progresses. It is always equated with the value of the program counter after the current instruction is assembled. It may only be used in the operand field.

Examples:

```
JMP  $       means jump to the location
MOV  A,B       after this instruction;
               that is, the MOV instruction

LDA  $+5     means load the data at the
DB   0         fifth location after this
DB   1         location.  In this case,
DB   2         the data has the value 5.
DB   3
DB   4
DB   5
```

Relative Symbolic Addressing

If the name of a particular location is known, a nearby location may be specified using the known name and a numeric offset. Example:

```
     JMP   BEG
     JPE   BEG+4
     CC    SUB
     CALL  $+48
BEG  MOV   A,B
     HLT
     MVI   C, 'B'
     INR   B
```

In this example the instruction JMP BEG refers to the MOV A,B instruction. The instruction JPE BEG+4 refers to the INR B instruction. BEG+4 means the address BEG plus four bytes. This form of addressing can be used to locate several bytes before or after a named location.

Constants

The Assembler allows the user to write positive or negative numbers directly in a statement. They will be regarded as decimal constants and their binary equivalents will be used appropriately. All unsigned numbers are considered positive. Decimal constants can be defined using the descriptor "D" after the numeric value. (This is not required, as the default is decimal.)

Hexadecimal constants may be defined using the descriptor "H" after a numeric value. IE. +10H, 10H, 3AH, 0F4H.

Note that a hexadecimal constant cannot start with the digits A-F. In this case, a leading 0 must be included. This enables the assembler to differentiate between a numeric value and a symbol.

ASCII constants may be defined by enclosing the ASCII character within single quote marks, i.e., 'C'. For double word constants, two characters may be defined within one quote string.

Expressions

An expression is a sequence of one or more symbols, constants or other expressions separated by the arithmetic operators plus or minus.

```
PAM +3
ISAB-'A'+52
LOOP+32H-5
```

Expressions are calculated using 16 bit arithmetic. All arithmetic is done modulo 65536. Single byte data cannot contain a value greater than 255 or less than -256. Any value outside this range will result in an assembler error.

Pseudo-Operations

The pseudo-operations are written as ordinary statements, but they direct the assembler to perform certain functions which do not always develop 8080 machine code. The following section describes the pseudo-ops.

ORG----Set Program Origin

Format is

```
label ORG expression
```

where the label is optional but if present will be equaled to the given expression.

END----End of Assembly

The pseudo-op informs the assembler that the last source statement has been read. The assembler will then start on pass 2 and terminate the assembly and pass control back to the executive. This pseudo-op is not needed when assembling from a memory file since the assembler will stop when an end of file indicator has been reached.

EQU----Equal Symbolic Value

Format is

label EQU expression

where label is a symbol the value of which will be determined from the expression, and expression is an expression which when evaluated will be assigned to the symbol given in the name field.

DS----Define Storage

Format is

label DS expression.

The DS causes the assembler to advance the Assembly Program Counter, effectively skipping past a given number of memory bytes.

DB----Define Byte

Format is

label DB expression.

This pseudo-op is used to reserve one byte of storage. The content of the byte is specified in the argument field.

DW----Define Word

This pseudo-op is used to define two bytes of storage. The evaluated argument will be placed in the two bytes; high order 8 bits in the low order byte, and the low order 8 bits in the high order byte. This conforms to the Intel format for two byte addresses.

Assembler Errors

The following error flags are output on the assembler listing when the error occurs. Some of the errors are only output during pass 1.

O Opcode Error
L Label Error
D Duplicate Label Error
M Missing Label Error
V Value Error
U Undefined Symbol
S Snytax Error
R Register Error
A Argument Error.

OBJECT TAPE FORMAT

The IMSAI Self-Contained System is supplied on paper tape in a blocked hexadecimal format. The data on the tape is blocked into discrete records, each record containing record length, record type, memory address and checksum information in addition to data. A frame-by-frame description is as follows:

Frame 0	Record Mark. Signals the start of a record. The ASCII character colon (":" HEX 3A) is used as the record mark.
Frames 1,2 (0-9,A-F)	Record Length. Two ASCII characters representing a hexadecimal number in the range 0 to 'FF' (0 to 255). This is the count of actual data bytes in the record type or checksum. A record length of 0 indicates end of file.
Frames 3 to 6	Load Address. Four ASCII characters that represent the initial memory location where the data following will be loaded. The first data byte is stored in the location pointed to by the load address; succeeding data bytes are loaded into ascending addresses.
Frames 7, 8	Record Type. Two ASCII characters. Currently all records are type 0. This field is reserved for future expansion.
Frames 9 to 9+2* (Record Length) -1	Data. Each 8 bit memory word is represented by two frames containing the ASCII characters (0 to 9, A to F) to represent a hexadecimal value 0 to 'FF'H (0 to 255).

Frames 9+2* (Record Length) to 9+2*(Record Length) +1	Checksum. The checksum is the negative of the sum of all 8 bit bytes in the record since the record mark (":") evaluated modulus 256. That is, if you add together all the 8 bit bytes, ignoring all carries out of an 8-bit sum, then add the checksum, the result is zero.

Example: If memory locations 1 through 3 contain 53F8EC, the format of the hex file produced when these locations are punched is:

:0300010053F8ECC5

SAVING AND RESTORING PROGRAMS

While the system has no explicit provision for saving and restoring programs, it is possible to do so with an ASR style teletype. The procedure is as follows:

1. Make the file you want to save the current file.
2. Type 'LIST', but don't type the carriage return.
3. Turn on the paper tape punch.
4. Type carriage return. The program will be listed on the teletype and simultaneously punched on the paper tape punch.
5. When the 'LIST' is completed, turn off the punch.

The procedure for restoring the file is as follows:

1. Make the file you want to restore into the current file.
2. Mount the tape in the paper tape reader.
3. Start the paper tape reader. The program will be automatically read in.

An analogous procedure, using the DUMP and ENTR commands, may be used to save and restore object code.

```
                ; REVISION 2             06 OCT 76
                ;
                ; ******** SELF CONTAINED SYSTEM ********
                ;
0000                    ORG     00H
0000 C34000             JMP     INITA   ;DEAD START
0003 C36700             JMP     EOR     ;RESTART MONITOR
                ;
0006                    ORG     08H
0008 C32E0D             JMP     BRKP    ;BREAKPOINT RESTART
                ;
0008                    ORG     40H
                ;
                ; THIS ROUTINE SETS UP THE SIO BOARD
                ;
0040 3EAA       INITA:  MVI     A,0AAH  ;GET DUMMY MODE WORD
0042 D303               OUT     TTS     ;OUTPUT IT
0044 3E40               MVI     A,40H   ;GET RESET BIT
0046 D303               OUT     TTS     ;RESET SIO BOARD
0048 3ECE               MVI     A,0CEH  ;GET REAL MODE WORD
004A D303               OUT     TTS     ;SET THE MODE FOR REAL
004C 3E37               MVI     A,37H   ;GET THE COMMAND
004E D303               OUT     TTS     ;OUTPUT IT
                ;
                ; THIS ROUTINE INITIALIZES THE FILE AREA FOR SUBSEQUENT
                ; PROCESSING
                ;
0050 212410             LXI     H,FILE0
0053 0E4E               MVI     C,MAXFIL*FELEN
0055 AF                 XRA     A
0056 77         INIT2:  MOV     M,A
0057 23                 INX     H
0058 0D                 DCR     C
0059 C25600             JNZ     INIT2
                ;
                ; CLEAR THE BREAKPOINT TABLE
                ;
005C 0618               MVI     B,NBR*3
005E 210C10             LXI     H,BRT
0061 77         INIT3:  MOV     M,A
0062 23                 INX     H
0063 05                 DCR     B
0064 C26100             JNZ     INIT3
                ;
                ; THIS IS THE STARTING POINT OF THE SELF CONTAINED
                ; SYSTEM ONCE THE SYSTEM HAS BEEN INITIALIZED.  COMMANDS
                ; ARE READ FROM THE USER, EXECUTED, AND CONTROL RETURNS
                ; BACK TO THIS POINT TO READ ANOTHER COMMAND.
                ;
0067 31B210     EOR:    LXI     SP,AREA+18
006A CD0E01             CALL    CRLF    ;PRINT C/R, LINE FEED
006D CD8000             CALL    READ    ;READ INPUT LINE
0070 23                 INX     H
0071 7E                 MOV     A,M     ;FETCH FIRST CHARACTER
0072 FE3A               CPI     '9'+1   ;COMMAND OR LINE NUMBER?
0074 DAB504             JC      LINE    ;JUMP IF LINE FOR FILE
0077 CD7301             CALL    VALC    ;GET COMMAND VALUES
007A CD2B01             CALL    COMM    ;CHECK LEGAL COMMANDS
007D C36700             JMP     EOR
                ;
                ; THIS ROUTINE READS IN A LINE FROM THE TTY AND PLACES
                ; IT IN AN INPUT BUFFER.
                ; THE FOLLOWING ARE SPECIAL CHARACTERS
                ;   CR            TERMINATES READ ROUTINE
```

```
                ;   LF          NOT RECOGNIZED BY ROUTINE
                ;   CTRL X      DELETE CURRENT LINE
                ;   DEL         DELETE CHARACTER
                ; ALL DISPLAYABLE CHARACTERS BETWEEN BLANK & Z AND THE
                ; ABOVE ARE RECOGNIZED BY THE READ ROUTINE, ALL OTHERS
                ; ARE SKIPPED OVER.  THE ROUTINE WILL NOT ACCEPT MORE
                ; CHARACTERS THAN THE INPUT BUFFER WILL HOLD.
                ;
0080 21C710     READ:   LXI     H,IBUF  ;GET INPUT BUFFER ADDRESS
0083 227410             SHLD    ADDS    ;SAVE ADDRESS
0086 1E02               MVI     E,2     ;INITIALIZE CHARACTER COUNT
0088 CDF600     NEXT:   CALL    IN8     ;READ A LINE
008B 78                 MOV     A,B
008C FE18               CPI     24      ;CHECK FOR CTRL X
008E C29700             JNZ     CR
0091 CD0E01             CALL    CRLF    ;OUTPUT A CRLF
0094 C38000             JMP     READ
0097 FE0D       CR:     CPI     ASCR    ;GET AN ASCII CR
0099 C2B200             JNZ     DEL
009C 7D                 MOV     A,L
009D FEC7               CPI     IBUF AND 0FFH   ;CHECK FOR FIRST CHAR
009F CA8000             JZ      READ
00A2 360D               MVI     M,ASCR  ;PLACE CR AT END OF LINE
00A4 23                 INX     H
00A5 3601               MVI     M,1     ;PLACE EOF INDICATOR IN LINE
00A7 23                 INX     H
00A8 3E1A               MVI     A,IBUF+83 AND 0FFH
00AA CDE100             CALL    CLER    ;CLEAR REMAINING BUFFER
00AD 21C610             LXI     H,IBUF-1
00B0 73                 MOV     M,E     ;SAVE CHARACTER COUNT
00B1 C9                 RET
00B2 FE7F       DEL:    CPI     127     ;CHECK FOR DELETE CHARACTER
00B4 C2C700             JNZ     CHAR
00B7 3EC7               MVI     A,IBUF AND 0FFH
00B9 BD                 CMP     L       ;IS THIS 1ST CHARACTER
00BA CA8800             JZ      NEXT
00BD 2B                 DCX     H       ;DECREMENT POINTER
00BE 1D                 DCR     E       ;DECREMENT COUNT
00BF 065F       BSPA:   MVI     B,5FH
00C1 CD0301             CALL    OUT8
00C4 C38800             JMP     NEXT
00C7 FE20       CHAR:   CPI     ' '     ;CHECK FOR LEGAL CHARACTER
00C9 DA8800             JC      NEXT
00CC FE5B               CPI     'Z'+1
00CE D28800             JNC     NEXT
00D1 47                 MOV     B,A
00D2 CD0301             CALL    OUT8    ;ECHO CHARACTER
00D5 70                 MOV     M,B
00D6 3E18               MVI     A,IBUF+81 AND 0FFH
00D8 BD                 CMP     L       ;CHECK FOR END OF LINE
00D9 CABF00             JZ      BSPA
00DC 23                 INX     H
00DD 1C                 INR     E       ;INCREMENT CHARACTER COUNT
00DE C38800             JMP     NEXT
                ;
                ;
                ;
                ; THIS ROUTINE IS USED TO BLANK OUT A PORTION OF MEMORY
                ;
00E1 BD         CLER:   CMP     L
00E2 C8                 RZ
00E3 3620               MVI     M,' '   ;PLACE BLANK IN MEMORY
00E5 23                 INX     H
00E6 C3E100             JMP     CLER
                ;
                ; SEE IF TTY INPUT READY AND CHECK FOR CTRL X.
```

```
                  ;
                  ; RETURN WITH ZERO SET IFF CTRL X SEEN.
                  ;
00E9 DB03         INK:    IN      TTS      ;GET TTY STATUS
00EB 2F                   CMA              ;INVERT STATUS
00EC E602                 ANI     TTYDA    ;IS DATA AVAILABLE?
00EE C0                   RNZ              ;RETURN IF NOT
00EF DB02                 IN      TTI      ;GET THE CHAR
00F1 E67F                 ANI     07FH     ;STRIP OFF PARITY
00F3 FE18                 CPI     'X'-40H  ;IS IT A CTRL X?
00F5 C9                   RET
                  ;
                  ; THIS ROUTINE READS A BYTE OF DATA FROM THE USART
                  ;
00F6 DB03         IN8:    IN      TTS      ;READ USART STATUS
00F8 E602                 ANI     TTYDA
00FA CAF600               JZ      IN8
00FD DB02                 IN      TTI      ;READ DATA
00FF E67F                 ANI     127      ;STRIP OFF PARITY
0101 47                   MOV     B,A
0102 C9                   RET
                  ;
                  ; THIS ROUTINE OUTPUTS A BYTE OF DATA TO THE USART
                  ;
0103 DB03         OUT8:   IN      TTS      ;READ STATUS
0105 E601                 ANI     TTYTR
0107 CA0301               JZ      OUT8
010A 78           OK:     MOV     A,B
010B D302                 OUT     TTO      ;TRANSMIT DATA
010D C9                   RET
                  ;
                  ; THIS ROUTINE WILL OUTPUT A CARRIAGE RETURN AND
                  ; LINE FEED FOLLOWED BY TWO DELETE CHARACTERS WHICH
                  ; PROVIDE TIME FOR PRINT HEAD TO RETURN.
                  ;
010E 060D         CRLF:   MVI     B,13     ;CR
0110 CD0301               CALL    OUT8
0113 060A         LF:     MVI     B,10     ;LF
0115 CD0301               CALL    OUT8
0118 067F                 MVI     B,127
011A CD0301               CALL    OUT8
011D CD0301               CALL    OUT8
0120 C9                   RET
                  ;
                  ; THIS ROUTINE JUMPS TO A LOCATION IN MEMORY GIVEN BY
                  ; THE INPUT COMMAND AND BEGINS EXECUTION OF PROGRAM
                  ; THERE.
                  ;
0121 CD0003       EXEC:   CALL    VCHK     ;CHECK FOR PARAMETER
0124 CD0E01               CALL    CRLF
0127 2A8A10               LHLD    BBUF     ;FETCH ADDRESS
012A E9                   PCHL             ;JUMP TO PROGRAM
                  ;
                  ;
                  ;
                  ;
                  ; THIS ROUTINE CHECKS THE INPUT COMMAND AGAINST ALL
                  ; LEGAL COMMANDS STORED IN A TABLE.  IF A LEGAL COMMAND
                  ; IS FOUND, A JUMP IS MADE TO THAT ROUTINE.  OTHERWISE
                  ; AN ERROR MESSAGE IS OUTPUT TO THE USER.
                  ;
012B 11BE02       COMM:   LXI     D,CTAB   ;COMMAND TABLE ADDRESS
012E 060B                 MVI     B,NCOM   ;NUMBER OF COMMANDS
0130 3E04                 MVI     A,4      ;LENGTH OF COMMAND
0132 329510               STA     NCHR     ;SAVE
0135 CD3C01               CALL    COMS     ;SEARCH TABLE
0138 C25A04               JNZ     WHAT     ;JUMP IF ILLEGAL COMMAND
```

```
0138 E9                  PCHL            ;BE HERE NOW
                 ;
                 ;
                 ;
                 ;
                 ;
                 ; THIS ROUTINE CHECKS TO SEE IF A BASE CHARACTER STRING
                 ; IS EQUAL TO ANY OF THE STRINGS CONTAINED IN A TABLE
                 ; POINTED TO BY D,E.  THE TABLE CONSISTS OF ANY NUMBER
                 ; OF CHARS, WITH 2 BYTES CONTAINING VALUES ASSOCIATED
                 ; WITH IT.  REG B CONTAINS THE # OF STRINGS TO COMPARE.
                 ; THIS ROUTINE CAN BE USED TO SEARCH THROUGH A COMMAND
                 ; OR SYMBOL TABLE.  ON RETURN, IF THE ZERO FLAG IS SET,
                 ; A MATCH WAS FOUND; IF NOT, NO MATCH WAS FOUND.  IF
                 ; A MATCH WAS FOUND, D,E POINT TO THE LAST BYTE
                 ; ASSOCIATED WITH THE CHARACTER STRING.  IF NOT, D,E
                 ; POINT TO THE NEXT LOCATION AFTER THE END OF THE TABLE.
                 ;
013C 2A7410      COMS:   LHLD    ADDS     ;FETCH COMPARE ADDRESS
013F 3A9510              LDA     NCHR     ;GET LENGTH OF STRING
0142 4F                  MOV     C,A
0143 CD5301              CALL    SEAR     ;COMPARE STRINGS
0146 1A                  LDAX    D        ;FETCH VALUE
0147 6F                  MOV     L,A
0148 13                  INX     D
0149 1A                  LDAX    D        ;FETCH VALUE
014A 67                  MOV     H,A
014B C8                  RZ
014C 13                  INX     D        ;SET TO NEXT STRING
014D 05                  DCR     B        ;DECREMENT COUNT
014E C23C01              JNZ     COMS
0151 04                  INR     B        ;CLEAR ZERO FLAG
0152 C9                  RET
                 ;
                 ;
                 ;
                 ; THIS ROUTINE CHECKS TO SEE IF TWO CHARACTER STRINGS IN
                 ; MEMORY ARE EQUAL.  THE STRINGS ARE POINTED TO BY D,E
                 ; AND H,L.  ON RETURN, THE ZERO FLAG SET INDICATES A
                 ; MATCH.  REG C INDICATES THE LENGTH OF THE STRINGS. ON
                 ; RETURN, THE POINTERS POINT TO THE NEXT ADDRESS AFTER
                 ; THE CHARACTER STRINGS.
                 ;
0153 1A          SEAR:   LDAX    D        ;FETCH CHARACTER
0154 BE                  CMP     M        ;COMPARE CHARACTERS
0155 C25F01              JNZ     INCA
0158 23                  INX     H
0159 13                  INX     D
015A 0D                  DCR     C        ;DECREMENT CHARACTER COUNT
015B C25301              JNZ     SEAR
015E C9                  RET
015F 13          INCA:   INX     D
0160 0D                  DCR     C
0161 C25F01              JNZ     INCA
0164 0C                  INR     C        ;CLEAR ZERO FLAG
0165 C9                  RET
                 ;
                 ; THIS ROUTINE ZEROES OUT A BUFFER IN MEMORY WHICH IS
                 ; THEN USED BY OTHER SCANNING ROUTINES.
                 ;
0166 AF          ZBUF:   XRA     A        ;GET A ZERO
0167 118A10              LXI     D,ABUF+12 ;BUFFER ADDRESS
016A 060C                MVI     B,12     ;BUFFER LENGTH
016C 1B          ZBU1:   DCX     D        ;DECREMENT ADDRESS
016D 12                  STAX    D        ;ZERO BUFFER
016E 05                  DCR     B
```

```
016F C26C01            JNZ     ZBU1
0172 C9                RET
                ;
                ; THIS ROUTINE CALLS ETRA TO OBTAIN THE INPUT PARAMETER
                ; VALUES AND CALLS AN ERROR ROUTINE IF AN ERROR OCCURRED
                ; IN THAT ROUTINE.
                ;
0173 CD7A01     VALC:  CALL    ETRA     ;GET INPUT PARAMETERS
0176 DA5A04            JC      WHAT     ;JUMP IF ERROR
0179 C9                RET
                ;
                ; THIS ROUTINE EXTRACTS THE VALUES ASSOCIATED WITH A
                ; COMMAND FROM THE INPUT STREAM AND PLACES THEM IN THE
                ; ASCII BUFFER (ABUF).  IT ALSO CALLS A ROUTINE TO
                ; CONVERT THE ASCII HEXADECIMALS TO BINARY AND STORES
                ; THEM IN THE BINARY BUFFER (BBUF).  ON RETURN, CARRY
                ; SET INDICATES AN ERROR IN INPUT PARAMETERS.
                ;
017A 210000     ETRA:  LXI     H,0      ;GET A ZERO
017D 228C10            SHLD    BBUF+2   ;ZERO VALUE
0180 227610            SHLD    FBUF     ;SET NO FILE NAME
0183 CD6601            CALL    ZBUF     ;ZERO BUFFER
0186 21C610            LXI     H,IBUF-1
0189 23         VAL1:  INX     H
018A 7E                MOV     A,M      ;FETCH INPUT CHARACTER
018B FE20              CPI     ' '      ;LOOK FOR FIRST CHARACTER
018D 3F                CMC
018E D0                RNC              ;RETURN IF NO CARRY
018F C28901            JNZ     VAL1     ;JUMP IF NO BLACK
0192 229610            SHLD    PNTR     ;SAVE POINTER
0195 CD0D09            CALL    SBLK     ;SCAN TO FIRST PARAMETER
0198 3F                CMC
0199 D0                RNC              ;RETURN IF CR
019A FE2F              CPI     '/'
019C C2C401            JNZ     VAL5     ;NO FILE NAME
019F 117610            LXI     D,FBUF   ;NAME FOLLOWS PUT IN FBUF
01A2 0E05              MVI     C,NMLEN
01A4 23         VAL2:  INX     H
01A5 7E                MOV     A,M
01A6 FE2F              CPI     '/'
01A8 CAB401            JZ      VAL3
01AB 0D                DCR     C
01AC FA5A04            JM      WHAT
01AF 12                STAX    D        ;STORE FILE NAME
01B0 13                INX     D
01B1 C3A401            JMP     VAL2
01B4 3E20       VAL3:  MVI     A,' '    ;GET AN ASCII SPACE
01B6 0D         VAL4:  DCR     C
01B7 FABF01            JM      DONE
01BA 12                STAX    D        ;FILL IN WITH SPACES
01BB 13                INX     D
01BC C3B601            JMP     VAL4
01BF CD1409     DONE:  CALL    SBL2
01C2 3F                CMC
01C3 D0                RNC
01C4 117E10     VAL5:  LXI     D,ABUF
01C7 CD750B            CALL    ALPS     ;PLACE PARAMETER IN BUFFER
01CA 78                MOV     A,B      ;GET DIGIT COUNT
01CB FE05              CPI     5        ;CHECK NUMBER OF DIGITS
01CD 3F                CMC
01CE D8                RC               ;RETURN IF TOO MANY DIGITS
01CF 017E10            LXI     B,ABUF
01D2 CD1802            CALL    AHEX     ;CONVERT VALUE
01D5 D8                RC               ;ILLEGAL CHARACTER
01D6 228A10            SHLD    BBUF     ;SAVE   IN BINARY BUFFER
01D9 217E10            LXI     H,ABUF
```

```
01DC CD8D05          CALL    NORM     ;NORMALIZE ASCII VALUE
01DF CD0D09          CALL    SBLK     ;SCAN TO NEXT PARAMETER
01E2 3F              CMC
01E3 D0              RNC              ;RETURN IF CR
01E4 118210          LXI     D,ABUF+4
01E7 CD7508          CALL    ALPS     ;PLACE PARAMETER IN BUFFER
01EA 78              MOV     A,B      ;GET DIGIT COUNT
01EB FE05            CPI     5        ;CHECK NUMBER OF DIGITS
01ED 3F              CMC
01EE D8              RC               ;RETURN IF TOO MANY DIGITS
01EF 018210          LXI     B,ABUF+4
01F2 CD1B02          CALL    AHEX     ;CONVERT VALUE
01F5 D8              RC               ;ILLEGAL VALUE
01F6 228C10          SHLD    BBUF+2   ;SAVE IN BINARY BUFFER
01F9 218210          LXI     H,ABUF+4
01FC CD8D05          CALL    NORM     ;NORMALIZE ASCII VALUE
01FF B7              ORA     A        ;CLEAR CARRY
0200 C9              RET
                ;
                ; THIS ROUTINE FETCHES DIGITS FROM THE BUFFER ADDRESSED
                ; BY B,C AND CONVERTS THE ASCII DECIMAL DIGITS INTO
                ; BINARY.  UP TO A 16-BIT VALUE CAN BE CONVERTED.  THE
                ; SCAN STOPS WHEN A BINARY ZERO IS FOUND IN THE BUFFER.
                ;
0201 210000     ADEC:  LXI   H,0      ;GEP A 16 BIT ZERO
0204 0A         ADE1:  LDAX  B        ;FETCH ASCII DIGIT
0205 B7              ORA     A        ;SET ZERO FLAG
0206 C8              RZ               ;RETURN IFF FINISHED
0207 54              MOV     D,H      ;SAVE CURRENT VALUE
0208 5D              MOV     E,L      ;SAVE CURRENT VALUE
0209 29              DAD     H        ;TIMES TWO
020A 29              DAD     H        ;TIMES TWO
020B 19              DAD     D        ;ADD IN ORIGINAL VALUE
020C 29              DAD     H        ;TIMES TWO
020D D630            SUI     48       ;ASCII BIAS
020F FE0A            CPI     10       ;CHECK FOR LEGAL VALUE
0211 3F              CMC
0212 D8              RC               ;RETURN IF ERROR
0213 5F              MOV     E,A
0214 1600            MVI     D,0
0216 19              DAD     D        ;ADD IN NEXT DIGIT
0217 03              INX     B        ;INCREMENT POINTER
0218 C30402          JMP     ADE1
                ;
                ;
                ; THIS ROUTINE FETCHES DIGITS FROM THE BUFFER ADDRESSED
                ; BY B,C AND CONVERTS THE ASCII HEXADECIMAL DIGITS INTO
                ; BINARY.  UP TO A 16-BIT VALUE CAN BE CONVERTED.  THE
                ; SCAN STOPS WHEN A BINARY ZERO IS FOUND IN THE BUFFER.
                ;
021B 210000     AHEX:  LXI   H,0      ;GET A 16 BIT ZERO
021E 0A         AHE1:  LDAX  B        ;FETCH ASCII DIGIT
021F B7              ORA     A        ;SET ZERO FLAG
0220 C8              RZ               ;RETURN IF DONE
0221 29              DAD     H        ;LEFT SHIFT
0222 29              DAD     H        ;LEFT SHIFT
0223 29              DAD     H        ;LEFT SHIFT
0224 29              DAD     H        ;LEFT SHIFT
0225 CD3202          CALL    AHS1     ;CONVERT TO BINARY
0228 FE10            CPI     10H      ;CHECK FOR LEGAL VALUE
022A 3F              CMC
022B D8              RC               ;RETURN IF ERROR
022C 85              ADD     L
022D 6F              MOV     L,A
022E 03              INX     B        ;INCREMENT POINTER
022F C31E02          JMP     AHE1
```

```
                ;
                ; THIS SUBROUTINE CONVERTS ASCII HEX DIGITS INTO BINARY
                ;
0232 D630       AHSI:   SUI     48      ;ASCII BIAS
0234 FE0A               CPI     10      ;DIGIT 0-10
0236 D8                 RC
0237 D607               SUI     7       ;ALPHA BIAS
0239 C9                 RET
                ;
                ;
                ; THIS ROUTINE CONVERTS A BINARY VALUE TO ASCII
                ; HEXADECIMAL AND OUTPUTS THE CHARACTERS TO THE TTY.
                ;
023A CD8602     HOUT:   CALL    BINH    ;CONVERT VALUE
023D 217410             LXI     H,HCON  ;CONVERSION AREA
0240 46         CHOT:   MOV     B,M     ;FETCH OUTPUT CHARACTER
0241 CD0301             CALL    OUT8    ;OUTPUT CHARACTER
0244 23                 INX     H
0245 46                 MOV     B,M     ;FETCH CHARACTER
0246 CD0301             CALL    OUT8    ;OUTPUT CHARACTER
0249 C9                 RET
                ;
                ; THIS ROUTINE DOES THE SAME AS ABOVE BUT OUTPUTS A
                ; BLANK AFTER THE LAST CHARACTER
                ;
024A CD3A02     HOTB:   CALL    HOUT    ;CONVERT AND OUTPUT
024D CD5D02             CALL    BLKI    ;OUTPUT A BLANK
0250 C9                 RET
                ;
                ;
                ; THIS ROUTINE CONVERTS A BINARY VALUE TO ASCII
                ; DECIMAL DIGITS AND OUTPUTS THE CHARACTERS TO THE TTY
                ;
0251 CDA302     DOUT:   CALL    BIND    ;CONVERT VALUE
0254 CD3D02             CALL    HOUT+3  ;OUTPUT VALUE (2 DIGITS)
0257 23                 INX     H
0258 46                 MOV     B,M     ;GET LAST DIGIT
0259 CD0301             CALL    OUT8    ;OUTPUT
025C C9                 RET
                ;
                ; THIS ROUTINE OUTPUTS A BLANK
                ;
025D 0620       BLKI:   MVI     B,' '   ;GET A BLANK
025F CD0301             CALL    OUT8
0262 C9                 RET
                ;
                ;
                ; THIS ROUTINE IS USED BY OTHER ROUTINES TO INCREMENT
                ; THE STARTING ADDRESS IN A COMMAND AND COMPARE IT WITH
                ; THE FINAL ADDRESS IN THE COMMAND.  ON RETURN, THE
                ; CARRY FLAG SET INDICATES THAT THE FINAL ADDRESS HAS
                ; BEEN REACHED.
                ;
0263 2A8A10     ACHK:   LHLD    BBUF    ;FETCH START ADDRESS
0266 3A8D10             LDA     BBUF+3  ;STOP ADRESS (HIGH)
0269 BC                 CMP     H       ;COMPARE ADDRESSES
026A C27502             JNZ     ACH1
026D 3A8C10             LDA     BBUF+2  ;STOP ADDRESS (LOW)
0270 BD                 CMP     L       ;COMPARE ADDRESSES
0271 C27502             JNZ     ACH1
0274 37                 STC             ;SET CARRY IF EQUAL
0275 23         ACH1:   INX     H       ;INCREMENT START ADDRESS
0276 228A10             SHLD    BBUF    ;STORE START ADDRESS
0279 C9                 RET
                ;
                ;
```

```
                ;
                ; THIS ROUTINE OUTPUTS CHARACTERS OF A STRING
                ; UNTIL A CARRIAGE RETURN IS FOUND.
                ;
027A 46         SCRN:   MOV     B,M       ;FETCH CHARACTER
027B 3E0D               MVI     A,13      ;CARRIAGE RETURN
027D B8                 CMP     B         ;CHARACTER = CR?
027E C8                 RZ
027F CD0301             CALL    OUT8      ;OUTPUT CHARACTER
0282 23                 INX     H         ;INCREMENT ADDRESS
0283 C37A02             JMP     SCRN
                ;
                ;
                ; THIS ROUTINE CONVERTS THE BINARY VALUE IN REG A INTO
                ; ASCII HEXADECIMAL DIGITS AND STORES THEM IN MEMORY.
                ;
0286 217410     BINH:   LXI     H,HCON    ;CONVERSION
0289 47                 MOV     B,A       ;SAVE VALUE
028A 1F                 RAR
028B 1F                 RAR
028C 1F                 RAR
028D 1F                 RAR
028E CD9902             CALL    BIN1
0291 77                 MOV     M,A
0292 23                 INX     H
0293 78                 MOV     A,B
0294 CD9902             CALL    BIN1      ;CONVERT TO ASCII
0297 77                 MOV     M,A
0298 C9                 RET
                ;
                ; THIS ROUTINE CONVERTS A VALUE TO HEXADECIMAL
                ;
0299 E60F       BIN1:   ANI     0FH       ;LOW 4 BITS
029B C630               ADI     48        ;CONVERT TO ASCII
029D FE3A               CPI     58        ;DIGIT 0-9
029F D8                 RC
02A0 C607               ADI     7         ;MODIFY FOR A-F
02A2 C9                 RET
                ;
                ;
                ; THIS ROUTINE CONVERTS THE BINARY VALUE IN REG A INTO
                ; ASCII DECIMAL DIGITS AND STORES THEM IN MEMORY
                ;
02A3 217410     BIND:   LXI     H,HCON    ;CONVERSION ADDRESS
02A6 0664               MVI     B,100
02A8 CDB402             CALL    BID1      ;CONVERT HUNDREDS DIGIT
02AB 060A               MVI     B,10
02AD CDB402             CALL    BID1      ;CONVERT TENS DIGIT
02B0 C630               ADI     '0'       ;GET UNITS DIGIT
02B2 77                 MOV     M,A       ;STORE IN MEMORY
02B3 C9                 RET
                ;
                ; THIS ROUTINE CONVERTS A VALUE TO DECIMAL
                ;
02B4 362F       BID1:   MVI     M,'0'-1 ;INITIALIZE DIGIT COUNT
02B6 34                 INR     M
02B7 90                 SUB     B         ;CHECK DIGIT
02B8 D2B602             JNC     BID1+2
02BB 80                 ADD     B         ;RESTORE VALUE
02BC 23                 INX     H
02BD C9                 RET
                ;
                ;
                ; LEGAL COMMAND TABLE
                ;
02BE 44554D50   CTAB:   DB      'DUMP'    ;DUMP COMMAND
02C2 0803               DW      DUMP      :COMMAND ADDRESS
```

```
02C4 45584543          DB      'EXEC'  ;EXECUTE COMMAND
02C8 2101              DW      EXEC    ;COMMAND ADDRESS
02CA 454E5452          DB      'ENTR'  ;ENTER COMMAND
02CE 7604              DW      ENTR
02D0 46494C45          DB      'FILE'  ;FILE COMMAND
02D4 3E03              DW      FILE    ;COMMAND ADDRESS
02D6 4C495354          DB      'LIST'  ;LIST COMMAND
02DA D005              DW      LIST    ;COMMAND ADDRESS
02DC 44454C54          DB      'DELT'  ;DELETE COMMAND
02E0 E705              DW      DELL    ;COMMAND ADDRESS
02E2 4153534D          DB      'ASSM'  ;ASSEMBLE COMMAND
02E6 5E06              DW      ASSM    ;COMMAND ADDRESS
02E8 50414745          DB      'PAGE'  ;PAGE TRANSFER COMMAND
02EC 2203              DW      PAGE    ;COMMAND ADDRESS
02EE 43555354          DB      'CUST'  ;CUSTOMER COMMAND
02F2 0020              DW      2000H   ;COMMAND ADDRESS
02F4 4252454B          DB      'BREK'  ;BREAKPOINT COMMAND
02F8 D20C              DW      BREAK   ;COMMAND ADDRESS
02FA 50524F43          DB      'PROC'  ;;PROCEED COMMAND
02FE 8F0D              DW      PROC    ;COMMAND ADDRESS
                ;
                ;
                ; THIS ROUTINE CHECKS IF ANY PARAMETERS WERE ENTERED
                ; WITH THE COMMAND, IF NOT AN ERROR MESSAGE IS ISSUED
                ;
0300 3A7E10     VCHK:  LDA     ABUF    ;FETCH PARAMETER BYTE
0303 B7                ORA     A       ;SET FLAGS
0304 CA5A04            JZ      WHAT    ;NO PARAMETER
0307 C9                RET
                ;
                ;
                ; THIS ROUTINE DUMPS OUT THE CONTENTS OF MEMORY FROM
                ; THE START TO FINAL ADDRESSES GIVEN IN THE COMMAND.
                ;
0308 CD0003     DUMP:  CALL    VCHK    ;CHECK FOR PARAMETERS
030B CD0E01     DUMS:  CALL    CRLF    ;START NEW LINE
030E 2A8A10     DUM1:  LHLD    BBUF    ;FETCH MEMORY ADDRESS
0311 7E                MOV     A,M
0312 CD4A02            CALL    HOTB    ;OUTPUT VALUE
0315 CD6302            CALL    ACHK    ;CHECK ADDRESS
0318 D8                RC              ;RETURN IF FINISHED
0319 7D                MOV     A,L     ;IS NEXT ADDRESS
031A E60F              ANI     0FH     ;  DIVISIBLE BY 16?
031C C20E03            JNZ     DUM1
031F C30B03            JMP     DUMS
                ;
                ;
                ; THIS ROUTINE WILL MOVE 256 BYTES FROM 1ST ADDRESS
                ; GIVEN IN COMMAND TO 2ND ADDRESS IN COMMAND.
                ;
0322 CD0003     PAGE:  CALL    VCHK    ;CHECK FOR PARAMETER
0325 3A8210            LDA     ABUF+4  ;FETCH 2ND PARAMETER
0328 B7                ORA     A       ;DOES 2ND PARAMETER EXIST?
0329 CA5A04            JZ      WHAT
032C 2A8A10            LHLD    BBUF    ;FETCH MOVE TO ADDRESS
032F EB                XCHG
0330 2A8C10            LHLD    BBUF+2  ;FETCH MOVE TO ADDRESS
0333 0600              MVI     B,0     ;SET COUNTER
0335 1A         PAG1:  LDAX    D
0336 77                MOV     M,A
0337 23                INX     H
0338 13                INX     D
0339 05                DCR     B       ;DECREMENT COUNT
033A C23503            JNZ     PAG1
033D C9                RET
                ;
```

```
                ;
                ;
                ; THIS ROUTINE INITIALIZES THE BEGINNING OF FILE ADDRESS
                ; AND END OF FILE ADDRESS AS WELL AS THE FILE AREA
                ; WHEN THE FILE COMMAND IS USED
                ;
033E CD0E01     FILE:   CALL    CRLF
                ; CHECK FOR FILE PARAMETERS
0341 3A7610             LDA     FBUF
0344 B7                 ORA     A
0345 CA8903             JZ      FOUT    ;NO - GO LIST
0348 CD1804             CALL    FSEA    ;LOOK UP FILE
034B EB                 XCHG            ;PNTR IN DE
034C C26303             JNZ     TEST
                ; NO ENTRY
034F 3A7E10             LDA     ABUF    ;CHECK FOR PARAM
0352 B7                 ORA     A
0353 CA5D04             JZ      WHA1    ;NO?? - ERROR
                ; CHECK FOR ROOM IN DIRECTORY
0356 3A7D10             LDA     FEF
0359 B7                 ORA     A
035A C27803             JNZ     ROOM
035D 216B04             LXI     H,EMES1
0360 C36004             JMP     MESS
                ; ENTRY FOUND ARE THESE PARAMETERS
0363 3A7E10     TEST:   LDA     ABUF
0366 B7                 ORA     A
0367 CA8B03             JZ      SWAPS
036A 2A8A10             LHLD    BBUF
036D 7C                 MOV     A,H
036E B5                 ORA     L
036F CA8B03             JZ      SWAPS
0372 217004             LXI     H,EMES2 ;NO-NO CAN'T DO
0375 C36004             JMP     MESS    ;IT - DELETE FIRST
                ; MOVE FILE NAME TO BLOCK POINTED TO BY FREAD
0378 2A7B10     ROOM:   LHLD    FREAD
037B EB                 XCHG
037C 217610             LXI     H,FBUF  ;FILE NAME POINTER IN H,L
037F D5                 PUSH    D
0380 0E05               MVI     C,NMLEN ;NAME LENGTH COUNT
0382 7E         MOV23:  MOV     A,M
0383 12                 STAX    D
0384 13                 INX     D
0385 0D                 DCR     C       ;TEST COUNT
0386 23                 INX     H
0387 C28203             JNZ     MOV23
038A D1                 POP     D       ;RESTORE ENTRY POINTER
                ; MAKE FILE POINTED TO BY D,E CURRENT
038B 212410     SWAPS:  LXI     H,FILE0
038E 0E0D               MVI     C,FELEN ;ENTRY LENGTH
0390 1A         SWAP:   LDAX    D
0391 46                 MOV     B,M
0392 77                 MOV     M,A     ;EXCHANGE
0393 78                 MOV     A,B
0394 12                 STAX    D
0395 13                 INX     D
0396 23                 INX     H       ;BUMP POINTERS
0397 0D                 DCR     C       ;TEST COUNT
0398 C29003             JNZ     SWAP
                ; CHECK FOR 2ND PARAMETER
039B 3A7E10             LDA     ABUF
039E B7                 ORA     A
039F CAC303             JZ      FOOT    ;NO SECOND PARAMETER
                ; PROCESS 2ND PARAMETER
03A2 2A8A10             LHLD    BBUF    ;GET ADDRESS
03A5 222910             SHLD    BOFP    ;SET BEGIN
```

```
03A8 222B10             SHLD    EOFP     ;SET END
03AB 7D                 MOV     A,L      ;IS ADDRESS ZERO?
03AC B4                 ORA     H
03AD CAB203             JZ      FIL35    ;YES
03B0 3601      FIL30:   MVI     M,1      ;NON-ZERO - SET EOF
03B2 AF        FIL35:   XRA     A        ;AND MAX LINE #
03B3 322D10             STA     MAXL
03B6 C3C303             JMP     FOOT     ;OUTPUT PARAMETERS
03B9 3AC810    FOUT:    LDA     IBUF+4
03BC FE53               CPI     'S'      ;IS COMMAND FILES?
03BE 0E06               MVI     C,MAXFIL
03C0 CAC503             JZ      FOUL
03C3 0E01      FOOT:    MVI     C,1
               ; OUTPUT THE # OF ENTRIES IN C
03C5 212410    FOUL:    LXI     H,FILE0
03C8 79                 MOV     A,C
03C9 327D10    FINE:    STA     FOCNT    ;SAVE COUNT
03CC E5                 PUSH    H
03CD 110500             LXI     D,NMLEN
03D0 19                 DAD     D
03D1 7E                 MOV     A,M
03D2 B7                 ORA     A
03D3 C2E303             JNZ     FOOD     ;NON ZERO, OK TO OUTPUT
03D6 23                 INX     H
03D7 86                 ADD     M
03D8 23                 INX     H
03D9 C2E303             JNZ     FOOD
03DC 33                 INX     SP
03DD 33                 INX     SP
03DE 23                 INX     H
03DF 23                 INX     H
03E0 C3F803             JMP     FEET
               ; HAVE AN ENTRY TO OUTPUT
03E3 E1        FOOD:    POP     H        ;PTR
03E4 0E05               MVI     C,NMLEN
03E6 46        FAST:    MOV     B,M      ;LOAD CHARACTER TO B
03E7 CD0301             CALL    OUT8
03EA 0D                 DCR     C
03EB 23                 INX     H
03EC C2E603             JNZ     FAST     ;DO THE REST
               ; NOW OUTPUT BEGIN-END PTRS
03EF CD0404             CALL    FOOL     ;OUTPUT BEGIN
03F2 CD0404             CALL    FOOL     ;OUTPUT END
03F5 CD0E01             CALL    CRLF     ;AND C/R
               ; TEST COUNT, H,L POINTS PAST EOFP
03F8 110400    FEET:    LXI     D,FELEN-NMLEN-4
03FB 19                 DAD     D        ;MOVE TO NEXT ENTRY
03FC 3A7D10             LDA     FOCNT
03FF 3D                 DCR     A        ;TEST COUNT
0400 C2C903             JNZ     FINE     ;MORE TO DO
0403 C9                 RET              ;DONE!
               ; OUTPUT NUMBER POINTED TO BY H,L
               ; ON RET, H,L POINT 2 WORDS LATER
0404 CD5D02    FOOL:    CALL    BLK1     ;SPACE
0407 23                 INX     H
0408 7E                 MOV     A,M
0409 2B                 DCX     H
040A E5                 PUSH    H
040B CD3A02             CALL    HOUT     ;OUTPUT
040E E1                 POP     H
040F 7E                 MOV     A,M
0410 23                 INX     H
0411 23                 INX     H
0412 E5                 PUSH    H
0413 CD4A02             CALL    HOTB     ;OUTPUT
0416 E1                 POP     H        ;RESTORE H,L
```

```
0417 C9                 RET
                ; SEARCH THE FILE DIRECTORY FOR THE FILE
                ; WHOSE NAME IS IN FBUF.
                ; RETURN IF FOUND, ZERO IS OFF, H,L POINT TO
                ; ENTRY WHILE SEARCHING, ON ENTRY FOUND WITH ADDR
                ; ZERO, SET FEF TO >0 AND FREAD TO THE ADDR OF ENTRY
                ;
0418 AF         FSEA:   XRA     A
0419 327D10             STA     FEF     ;CLAIM NO FREE ENTRIES
041C 0606               MVI     B,MAXFIL        ;COUNT OF ENTRIES
041E 112410             LXI     D,FILE0 ;TABLE ADDRESS
0421 217610     FSE10:  LXI     H,FBUF
0424 0E05               MVI     C,NMLEN
0426 CD5301             CALL    SEAR    ;TEST STRINGS
0429 F5                 PUSH    PSW     ;SAVE FLAG
042A D5                 PUSH    D
042B 1A                 LDAX    D       ;GET BOFP
042C B7                 ORA     A       ;EMPTY ENTRY?
042D C24E04             JNZ     FSE20
0430 13                 INX     D       ;STORE OTHER WORD
0431 1A                 LDAX    D
0432 B7                 ORA     A
0433 C24E04             JNZ     FSE20   ;NOPE-GO TEST FOR MATCH
0436 EB                 XCHG
0437 11FAFF             LXI     D,-NMLEN-1
043A 19                 DAD     D       ;MOV TO BEGINNING
043B 227B10             SHLD    FREAD   ;SAVE ADDR
043E 7A                 MOV     A,D
043F 327D10             STA     FEF     ;SET FREE ENTRY FOUND
0442 E1                 POP     H       ;RESTOR INTERIM PTR
0443 F1                 POP     PSW     ;UNJUNK STACK
                ; MOVE TO NEXT ENTRY
0444 110800     FSE15:  LXI     D,FELEN-NMLEN
0447 19                 DAD     D
0448 EB                 XCHG            ;NEXT ENTRY ADDR IN DE
0449 05                 DCR     B       ;TEST COUNT
044A C8                 RZ              ;DONE--NOPE
044B C32104             JMP     FSE10   ;TRY NEXT
                ;ENTRY WASN'T FREE, TEST FOR MATCH
044E E1         FSE20:  POP     H
044F F1                 POP     PSW
0450 C24404             JNZ     FSE15   ;IF ZERO CLEAR, NO MATCH
                ;ENTRY FOUND
0453 11FBFF             LXI     D,-NMLEN        ;BACKUP
0456 19                 DAD     D       ;H,L POINTS TO ENTRY
0457 7A                 MOV     A,D
0458 B7                 ORA     A       ;CLEAR ZERO
0459 C9                 RET             ;THAT'S ALL
                ;
                ;
                ; OUTPUT ERROR MESSAGE FOR ILLEGAL COMMAND
                ;
045A CD0E01     WHAT:   CALL    CRLF    ;OUT CRLF
045D 216604     WHA1:   LXI     H,EMES  ;MESSAGE ADDRESS
0460 CD7A02     MESS:   CALL    SCRN
0463 C36700             JMP     EOR
                ;
0466 57484154 EMES:     DB      'WHAT'
046A 0D                 DB      13      ;CARRIAGE RETURN
046B 46554C4C EMES1:    DB      'FULL',13
046F 0D
0470 4E4F204E EMES2:    DB      'NO NO',13
0474 4F0D
                ;
                ;
                ; CALL ROUTINE TO ENTER DATA INTO MEMORY
```

```
                ; AND CHECK FOR ERROR ON RETURN
                ;
                ; THIS ROUTINE IS USED TO ENTER DATA VALUES INTO MEMORY.
                ; EACH VALUE IS ONE BYTE AND IS WRITTEN IN HEXADECIMAL
                ; VALUES GREATER THAT 255 WILL CAUSE CARRY TO BE SET
                ; AND RETURN TO BE MADE TO CALLING PROGRAM
                ;
0476 CD0003     ENTR:   CALL    VCHK    ;CHECK FOR PARAMETERS
0479 CD8304             CALL    ENTS
047C DA5A04             JC      WHAT
047F CD0E01             CALL    CRLF
0482 C9                 RET
                ;
                ;
002F            EEND    EQU     '/'     ;TERMINATION CHAR
0483 CD0E01     ENTS:   CALL    CRLF
0486 CD8000             CALL    READ    ;READ INPUT DATA
0489 21C710             LXI     H,IBUF  ;SET LINE POINTER
048C 229610             SHLD    PNTR    ;SAVE POINTER
048F CD6601     ENT1:   CALL    ZBUF    ;CLEAR BUFFER
0492 CD0D09             CALL    SBLK    ;CAN TO FIRST VALUE
0495 DA8304             JC      ENTS    ;JUMP IF CR FOUND
0498 FE2F               CPI     EEND
049A C8                 RZ              ;RETURN CARRY IS ZERO
049B CD750B             CALL    ALPS    ;PLACE VALUE IN BUFFER
049E 78                 MOV     A,B     ;GET DIGIT COUNT
049F FE03               CPI     3       ;CHECK NUR OF DIGITS
04A1 3F                 CMC
04A2 D8                 RC              ;RETURN IF MORE THAN    2 DIGITS
04A3 017E10             LXI     B,ABUF  ;CONVERSION ADDRESS
04A6 CD1B02             CALL    AHEX    ;CONVERT VALUE
04A9 D8                 RC              ;ERROR IN HEX CHARACTER
04AA 7D                 MOV     A,L
04AB 2A8A10             LHLD    BBUF    ;FETCH MEMORY ADDRESS
04AE 77                 MOV     M,A     ;PUT IN MEMORY
04AF CD7502             CALL    ACH1    ;INCREMENT MEMORY LOCATION
04B2 C38F04             JMP     ENT1
                ;
                ;
                ;
                ; THIS ROUTINE IS USED TO ENTER LINES INTO THE FILE
                ; AREA. THE LINE NUMBER IS FIRST CHECKED TO SEE IF IT IS
                ; A VALID NUMBER (0000-9999). NEXT IT IS CHECKED TO SEE
                ; IF IT IS GREATER THAN THE MAXIMUM CURRENT LINE NUMBER.
                ; IF IT IS, THE NEXT LINE IS INSERTED AT THE END OF THE
                ; CURRENT FILE AND THE MAXIMUM LINE NUMBER IS UPDATED AS
                ; WELL AS THE END OF FILE POSITION. LINE NUMBERS THAT
                ; ALREADY EXIST ARE INSERTED INTO THE FILE AREA AT THE
                ; APPROPRIATE PLACE AND ANY EXTRA CHARACTERS IN THE OLD
                ; LINE ARE DELETED.
                ;
04B5 3A2410     LINE:   LDA     FILE0   ;IS A FILE DEFINED?...
04B8 B7                 ORA     A
04B9 CA5A04             JZ      WHAT    ;ABORT IF NOT
04BC 0E04               MVI     C,4     ;NO OF DIGITS TO CHECK
04BE 21C510             LXI     H,IBUF-1        ;INITIALIZE ADDRESS
04C1 23         LICK:   INX     H
04C2 7E                 MOV     A,M     ;FETCH LINE DIGIT
04C3 FE30               CPI     '0'     ;CHECK FOR VALID NUMBER
04C5 DA5A04             JC      WHAT
04C8 FE3A               CPI     '9'+1
04CA D25A04             JNC     WHAT
04CD 0D                 DCR     C
04CE C2C104             JNZ     LICK
04D1 227410             SHLD    ADDS    ;FIND ADDRESS
04D4 113010             LXI     D,MAXL+3        ;GET ADDRESS
```

```
04D7 CDA205          CALL    COMO
04DA D2FA04          JNC     INSR
             ; GET HERE IF NEW LINE IS GREATER THAN MAXIMUM LINE #
04DD 23              INX     H
04DE CD9205          CALL    LODM     ;GET NEW LINE NUMBER
04E1 213010          LXI     H,MAXL+3
04E4 CD9A05          CALL    STOM     ;MAKE IT MAXIMUM LINE NUMBER
04E7 11C610          LXI     D,IBUF-1
04EA 2A2B10          LHLD    EOFP     ;END OF FILE POSITION
04ED 0E01            MVI     C,1
04EF CD8005          CALL    LMOV     ;PLACE LINE IN FILE
04F2 3601    SEOF:   MVI     M,1      ;END OF FILE INDICATOR
04F4 222B10          SHLD    EOFP     ;END OF FILE ADDRESS
04F7 C36700          JMP     EOR
             ; GET HERE IF NEW LINE MUST BE INSERTED INTO ALREADY
             ; EXISTING FILE AREA
04FA CD5205  INSR:   CALL    FINI     ;FIND LINE IN FILE
04FD 0E02            MVI     C,2
04FF CA0305          JZ      EQUL
0502 0D              DCR     C        ;NEW LN NOT EQUAL TO SOME OLD LN
0503 46      EQUL:   MOV     B,M
0504 2B              DCX     H
0505 3602            MVI     M,2      ;MOVE LINE INDICATOR
0507 227210          SHLD    INSP     ;INSERT LINE POSITION
050A 3AC610          LDA     IBUF-1   ;NEW LN COUNT
050D 0D              DCR     C
050E CA1805          JZ      LT       ;NEW LN NOT = OLD LN
0511 90              SUB     B        ;COUNT DIFFERENCE
0512 CA3B05          JZ      ZERO     ;LINE LENGTHS EQUAL
0515 DA2B05          JC      GT
             ; GET HERE IF # OF CHARS IN OLD LINE > # OF CHARS IN
             ; NEW LINE OR NEW LINE # WAS NOT EQUAL TO SOME OLD
             ; LINE #
0518 2A2B10  LT:     LHLD    EOFP     ;END OF FILE ADDRESS
051B 54              MOV     D,H
051C 5D              MOV     E,L
051D CD7B05          CALL    ADR      ;MOVE TO ADDRESS
0520 222B10          SHLD    EOFP     ;NEW END OF FILE ADDRESS
0523 0E02            MVI     C,2
0525 CD8905          CALL    RMOV     ;OPEN UP FILE AREA
0528 C33B05          JMP     ZERO
             ; GET HERE IF # OF CHARS IN OLD LINE < # OF CHARS IN
             ; NEW LINE.
052B 2F      GT:     CMA
052C 3C              INR     A        ;COUNT DIFFERENCE
052D 54              MOV     D,H
052E 5D              MOV     E,L
052F CD7B05          CALL    ADR
0532 EB              XCHG
0533 CD8005          CALL    LMOV     ;DELETE EXCESS CHAR IN FILE
0536 3601            MVI     M,1      ;E-O-F INDICATOR
0538 222B10          SHLD    EOFP     ;E-O-F ADDRESS
             ; GET HERE TO INSERT CURRENT LINE INTO FILE AREA
053B 2A7210  ZERO:   LHLD    INSP     ;INSERT ADDRESS
053E 360D            MVI     M,ASCR
0540 23              INX     H
0541 11C610          LXI     D,IBUF-1         ;NEW LINE ADDRESS
0544 0E01            MVI     C,1      ;CHECK VALUE
0546 CD8005          CALL    LMOV     ;PLACE LINE IN FILE
0549 C36700          JMP     EOR
             ;
             ;
             ; THIS ROUTINE IS USED TO FIND A LN IN THE FILE AREA
             ; WHICH IS GREATER THAN OR EQUAL TO THE CURRENT LINE #
054C 218110  FIND:   LXI     H,ABUF+3         ;BUFFER ADDRESS
054F 227410          SHLD    ADDS     ;SAVE ADDRESS
```

```
0552 2A2910   FIN1:   LHLD    BOFP    ;BEGIN FILE ADDRESS
0555 7C               MOV     A,H     ;RETURN TO MONITOR IF
0556 B5               ORA     L       ;  FILE IS EMPTY...
0557 CA6700           JZ      EOR
055A CD7405   FI1:    CALL    EO1     ;CHECK FOR END OF FILE
055D EB               XCHG
055E 2A7410           LHLD    ADDS    ;FETCH FIND ADDRESS
0561 EB               XCHG
0562 3E04             MVI     A,4
0564 CD7B05           CALL    ADR     ;BUMP LINE ADDRESS
0567 CDA205           CALL    COM0    ;COMPARE LINE NUMBERS
056A D8               RC
056B C8               RZ
056C 7E       FI2:    MOV     A,M
056D CD7B05           CALL    ADR     ;NEXT LINE ADDRESS
0570 C35A05           JMP     FI1
              ;
              ;
              ; WHEN SEARCHING THROUGH THE FILE AREA, THIS ROUTINE
              ; CHECKS TO SEE IF THE CURRENT ADDRESS IS THE END OF
              ; FILE
              ;
0573 23       EOF:    INX     H
0574 3E01     EO1:    MVI     A,1     ;E-O-F INDICATOR
0576 BE               CMP     M
0577 C0               RNZ
0578 C36700           JMP     EOR
              ;
              ;
              ; THIS ROUTINE IS USED TO ADD A VALUE TO AN ADDRESS
              ; CONTAINED IN REGISTER H,L
              ;
057B 85       ADR:    ADD     L
057C 6F               MOV     L,A
057D D0               RNC
057E 24               INR     H
057F C9               RET
              ;
              ;
              ; THIS ROUTINE WILL MOVE CHARACTER STRINGS FROM ONE
              ; LOCATION OF MEMORY TO ANOTHER
              ; CHARACTERS ARE MOVED FROM LOCATION ADDRESSED BY D,E
              ; TO LOCATION ADDRESSED BY H,L.  ADDITIONAL CHARACTERS
              ; ARE MOVED BY BUMPING POINTERS UNTIL THE CHARACTER IN
              ; REG C IS FETCHED.
              ;
0580 1A       LMOV:   LDAX    D       ;FETCH CHARACTER
0581 13               INX     D       ;INCREMENT FETCH ADDRESS
0582 B9               CMP     C       ;TERMINATION CHARACTER
0583 C8               RZ
0584 77               MOV     M,A     ;STORE CHARACTER
0585 23               INX     H       ;INCREMENT STORE ADDRESS
0586 C38005           JMP     LMOV
              ;
              ;
              ; THIS ROUTINE IS SIMILAR TO ABOVE EXCEPT THAT THE
              ; CHARACTER ADDRESS IS DECREMENTED AFTER EACH FETCH
              ; AND STORE
              ;
0589 1A       RMOV:   LDAX    D       ;FETCH CHARACTER
058A 1B               DCX     D       ;DECREMENT FETCH ADDRESS
058B B9               CMP     C       ;TERMINATION CHARACTER
058C C8               RZ
058D 77               MOV     M,A     ;STORE CHARACTER
058E 2B               DCX     H       ;DECREMENT STORE ADDRESS
058F C38905           JMP     RMOV
```

```
                ;
                ;
                ; THIS ROUTINE IS USED TO LOAD FOUR CHARACTERS FROM
                ; MEMORY INTO REGISTERS
                ;
0592 46         LODM:   MOV     B,M     ;FETCH CHARACTER
0593 23                 INX     H
0594 4E                 MOV     C,M     ;FETCH CHARACTER
0595 23                 INX     H
0596 56                 MOV     D,M     ;FETCH CHARACTER
0597 23                 INX     H
0598 5E                 MOV     E,M     ;FETCH CHARACTER
0599 C9                 RET
                ;
                ;
                ; THIS ROUTINE STORES FOUR CHARACTERS FROM THE REGISTERS
                ; INTO MEMORY
                ;
059A 73         STOM:   MOV     M,E     ;STORE CHARACTER
059B 2B                 DCX     H
059C 72                 MOV     M,D     ;STORE CHARACTER
059D 2B                 DCX     H
059E 71                 MOV     M,C     ;STORE CHARACTER
059F 2B                 DCX     H
05A0 70                 MOV     M,B     ;STORE CHARACTER
05A1 C9                 RET
                ;
                ;
                ; THIS ROUTINE IS USED TO COMPARE TWO CHARACTER STRINGS
                ; OF LENGTH 4, ON RETURN ZERO FLAG SET MEANS BOTH
                ; STRINGS ARE EQUAL.  CARRY FLAG =0 MEANS STRING ADDRESS
                ; BY D,E WAS GREATER THAN OR EQUAL TO CHARACTER STRING
                ; ADDRESSED BY H,L
                ;
05A2 0601       COM0:   MVI     B,1     ;EQUAL COUNTER
05A4 0E04               MVI     C,4     ;STRING LENGTH
05A6 B7                 ORA     A       ;CLEAR CARRY
05A7 1A         CO1:    LDAX    D       ;FETCH CHARACTER
05A8 9E                 SBB     M       ;COMPARE CHARACTERS
05A9 CAAD05             JZ      CO2
05AC 04                 INR     B       ;INCREMENT EQUAL COUNTER
05AD 1B         CO2:    DCX     D
05AE 2B                 DCX     H
05AF 0D                 DCR     C
05B0 C2A705             JNZ     CO1
05B3 05                 DCR     B
05B4 C9                 RET
                ;
                ; THIS ROUTINE IS SIMILAR TO THE ABOVE ROUTINE EXCEPT ON
                ; RETURN CARRY FLAG = 0 MEANS THAT CHARACTER STRING
                ; ADDRESSED BY D,E IS ONLY > STRING ADDRESSED BY H,L.
                ;
05B5 0E04       COM1:   MVI     C,4     ;STRING LENGTH
05B7 1A                 LDAX    D       ;TCH CHARACTER
05B8 D601               SUI     1
05BA C3A805             JMP     CO1+1
                ;
                ;
                ; THIS ROUTINE WILL TAKE ASCII CHARACTERS AND ADD ANY
                ; NECESSARY ASCII ZEROES SO THE RESULT IS A 4 CHARACTER
                ; ASCII VALUE
                ;
05BD CD9205     NORM:   CALL    LODM    ;LOAD CHARACTERS
05C0 AF                 XRA     A       ;FETCH A ZERO
05C1 B8                 CMP     B
05C2 C8                 RZ
```

```
05C3 BB          NORI:   CMP     E
05C4 C49A05              CNZ     STOM      ;STORE VALUES
05C7 C0                  RNZ
05C8 5A                  MOV     E,D       ;NORMALIZE VALUE
05C9 51                  MOV     D,C
05CA 48                  MOV     C,B
05CB 0630                MVI     B,'0'
05CD C3C305              JMP     NORI
                 ;
                 ;
                 ; THIS ROUTINE IS USED TO LIST THE CONTENTS OF THE FILE
                 ; AREA STARTING AT THE LINE NUMBER GIVEN IN THE COMMAND
                 ;
05D0 CD0E01      LIST:   CALL    CRLF
05D3 CD4C05              CALL    FIND      ;FIND STARTING LN
05D6 23          LIST0:  INX     H         ;OUTPUT LINE...
05D7 CD7A02              CALL    SCRN
05DA CD0E01              CALL    CRLF
05DD CD7305              CALL    EOF       ;CHECK FOR END OF FILE
05E0 CDE900              CALL    INK       ;CHECK FOR δX
05E3 C2D605              JNZ     LIST0     ;LOOP IF NO δX
05E6 C9                  RET
                 ;
                 ;
                 ; THIS ROUTINE IS USED TO DELETE LINES FROM THE
                 ; FILE AREA.  THE REMAINING FILE AREA IS THEN MOVED IN
                 ; MEMORY SO THAT THERE IS NO EXCESS SPACE.
                 ;
05E7 CD0003      DELL:   CALL    VCHK      ;CHECK FOR PARAMETER
05EA CD4C05              CALL    FIND      ;FIND LINE IN FILE AREA
05ED 227210              SHLD    DELP      ;SAVE DELETE POSITION
05F0 218510              LXI     H,ABUF+7
05F3 7E                  MOV     A,M       ;CHECK FOR 2ND PARAMETER
05F4 B7                  ORA     A         ;SET FLAGS
05F5 C2FB05              JNZ     DEL1
05F8 218110              LXI     H,ABUF+3         ;USE FIRST PARAMETER
05FB 227410      DEL1:   SHLD    ADDS      ;SAVE FIND ADDRESS
05FE EB                  XCHG
05FF 213010              LXI     H,MAXL+3
0602 CDA205              CALL    COM0      ;COMPARE LINE NUMBERS
0605 2A7210              LHLD    DELP      ;LOAD DELETE POSITION
0608 DA4906              JC      NOVR
                 ; GET HERE IF DELETION INVOLVES END OF FILE
060B 222B10              SHLD    EOFP      ;CHANGE E-O-F POSITION
060E 3601                MVI     M,1       ;SET E-O-F INDICATOR
0610 EB                  XCHG
0611 2A2910              LHLD    BOFP      ;GET BEGIN FILE ADDRESS
0614 EB                  XCHG
0615 060D                MVI     B,13      ;SET SCAN SWITCH
0617 2B                  DCX     H         ;CHECK FOR BOF
0618 7D          DEL2:   MOV     A,L
0619 93                  SUB     E
061A 7C                  MOV     A,H
061B 9A                  SBB     D
061C 3E0D                MVI     A,ASCR    ;LOOK FOR CR
061E DA4006              JC      DEL4      ;DECREMENTED PAST BOF
0621 05                  DCR     B
0622 2B                  DCX     H
0623 BE                  CMP     M         ;FIND NEW MAX LN
0624 C21806              JNZ     DEL2
0627 2B                  DCX     H
0628 7D                  MOV     A,L
0629 93                  SUB     E
062A 7C                  MOV     A,H
062B 9A                  SBB     D
062C DA4106              JC      DEL5
```

```
062F BE                 CMP     M        ;END OF PREVIOUS LINE
0630 23                 INX     H
0631 23                 INX     H
0632 CA3606             JZ      DEL3
0635 23                 INX     H
0636 CD9205     DEL3:   CALL    LODM     ;LOAD NEW MAX LN
0639 213010             LXI     H,MAXL+3         ;SET ADDRESS
063C CD9A05             CALL    STOM     ;STORE NEW MAX LN
063F C9                 RET
0640 B8         DEL4:   CMP     B        ;CHECK SWITCH
0641 EB         DEL5:   XCHG
0642 C23506             JNZ     DEL3-1
0645 322D10             STA     MAXL     ;MAKE MAX LN A SMALL NUMBER
0648 C9                 RET
                ; GET HERE IF DELETION IS IN MIDDLE OF FILE AREA
0649 CD5A05     NOVR:   CALL    FI1      ;FIND END OF DELETE AREA
064C CC6C05             CZ      FI2      ;NEXT LINE IF THIS LN EQUAL
064F EB         NOV1:   XCHG
0650 2A7210             LHLD    DELP     ;CHAR MOVE TO POSITION
0653 0E01               MVI     C,1      ;MOVE TERMINATOR
0655 CD8005             CALL    LMOV     ;COMPACT FILE AREA
0658 222B10             SHLD    EOFP     ;SET EOF POSITION
065B 3601               MVI     M,1      ;SET EOF INDICATOR
065D C9                 RET
                ;
                ; STARTING HERE IS THE SELF ASSEMBLER PROGRAM
                ; THIS PROGRAM ASSEMBLES PROGRAMS WHICH ARE
                ; IN THE FILE AREA
                ;
065E CD0003     ASSM:   CALL    VCHK     ;CHECK FOR PARAMETER
0661 3A8210             LDA     ABUF+4   ;GET 2ND PARAMETER
0664 B7                 ORA     A        ;CHECK FOR PARAMETERS
0665 C26E06             JNZ     ASM4
0668 2A8A10             LHLD    BBUF     ;FETCH 1ST PARAMETER
066B 228C10             SHLD    BBUF+2   ;STORE INTO 2ND PARAMETER
066E 3ACB10     ASM4:   LDA     IBUF+4   ;FETCH INPUT CHARACTER
0671 D645               SUI     'E'      ;RESET A IF ERRORS ONLY
0673 328E10             STA     AERR     ;SAVE ERROR FLAG
0676 AF                 XRA     A        ;GET A ZERO
0677 329810             STA     NOLA     ;INITIALIZE LABEL COUNT
067A 329410     ASM3:   STA     PASI     ;SET PASS INDICATOR
067D CD0E01             CALL    CRLF     ;INDICATE START OF PASS
0680 2A8A10             LHLD    BBUF     ;FETCH ORIGIN
0683 229210             SHLD    ASPC     ;INITIALIZE PC
0686 2A2910             LHLD    BOFP     ;GET START OF FILE
0689 227210             SHLD    APNT
068C 2A7210     ASM1:   LHLD    APNT     ;FETCH LINE POINTER
068F 31B210             LXI     SP,AREA+18
0692 7E                 MOV     A,M      ;FETCH CHARACTER
0693 FE01               CPI     1        ;END OF FILE?
0695 CA0109             JZ      EASS     ;JUMP IF END OF FILE
0698 EB                 XCHG
0699 13                 INX     D        ;INCREMENT ADDRESS
069A 21B210             LXI     H,OBUF   ;BLANK START ADDRESS
069D 3EC2               MVI     A,IBUF-5 AND 0FFH ;BLANK END ADDRESS
069F CDE100             CALL    CLER     ;BLANK OUT BUFFER
06A2 0E0D               MVI     C,ASCR   ;STOP CHARACTER
06A4 CD8005             CALL    LMOV     ;MOVE LINE INTO BUFFER
06A7 71                 MOV     M,C      ;PLACE CR IN BUFFER
06A8 EB                 XCHG
06A9 227210             SHLD    APNT     ;SAVE ADDRESS
06AC 3A9410             LDA     PASI     ;FETCH PASS INDICATOR
06AF B7                 ORA     A        ;SET FLAGW
06B0 C2B906             JNZ     ASM2     ;JUMP IF PASS 2
06B3 CDCC06             CALL    PAS1
06B6 C38C06             JMP     ASM1
```

```
0689 CD9307   ASM2:.  CALL   PAS2
068C 21B210           LXI    H,OBUF   ;OUTPUT BUFFER ADDRESS
068F CDC506           CALL   AOUT     ;OUTPUT LINE
06C2 C38C06           JMP    ASM1
              ;
              ; THIS ROUTINE IS USED TO OUTPUT THE LISTING FOR
              ; AN ASSEMBLY.  IT CHECKS THE ERROR SWITCH TO SEE IF
              ; ALL LINES ARE TO BE PRINTED OR JUST THOSE WITH
              ; ERRORS.
              ;
06C5 3A8E10   AOUT:   LDA    AERR     ;FETCH ERROR SWITCH
06C8 B7               ORA    A        ;SET FLAGS
06C9 C2D206           JNZ    AOU1     ;OUTPUT ALL LINES
06CC 3AB210   AOU2:   LDA    OBUF     ;FETCH ERROR INDICATOR
06CF FE20             CPI    ' '      ;CHECK FOR AN ERROR
06D1 C8               RZ              ;RETURN IF NO ERROR
06D2 21B210   AOU1:   LXI    H,OBUF   ;OUTPUT BUFFER ADDRESS
06D5 CD7A02           CALL   SCRN     ;OUTPUT LINE...
06D8 CD0E01           CALL   CRLF
06DB C9               RET
              ;
              ; PASS1 OF ASSEMBLER. USED TO FORM SYMBOL TABLE
              ;
06DC CD6601   PAS1:   CALL   ZBUF     ;CLEAR BUFFER
06DF 329410           STA    PASI     ;SET FOR PASS1
06E2 21C710           LXI    H,IBUF   ;INITIALIZE LINE POINTER
06E5 229610           SHLD   PNTR
06E8 7E               MOV    A,M      ;FETCH CHARACTER
06E9 FE20             CPI    ' '      ;CHECK FOR A BLANK
06EB CA1E07           JZ     OPC      ;JUMP IF NO LABEL
06EE FE2A             CPI    '*'      ;CHECK FOR COMMENT
06F0 C8               RZ              ;RETURN IF COMMENT
              ;
              ; PROCESS LABEL
              ;
06F1 CD2008           CALL   SLAB     ;GET AND CHECK LABEL
06F4 DADF0A           JC     OP5      ;ERROR IN LABEL
06F7 CAC70C           JZ     ERRD     ;DUPLICATE LABEL
06FA CD3507           CALL   LCHK     ;CHECK CHARACTER AFTER LABEL
06FD C2DF0A           JNZ    OP5      ;ERROR IF NO BLANK
0700 0E05             MVI    C,LLAB   ;LENGTH OF LABELS
0702 217E10           LXI    H,ABUF   ;SET BUFFER ADDRESS
0705 7E       MLAB:   MOV    A,M      ;FETCH NEXT CHARACTER
0706 12               STAX   D        ;STORE IN SYMBOL TABLE
0707 13               INX    D
0708 23               INX    H
0709 0D               DCR    C
070A C20507           JNZ    MLAB
070D EB               XCHG
070E 229010           SHLD   TABA     ;SAVE TABLE ADDRESS FOR EQU
0711 3A9310           LDA    ASPC+1   ;FETCH PC (HIGH)
0714 77               MOV    M,A
0715 23               INX    H
0716 3A9210           LDA    ASPC     ;FETCH PC (LOW)
0719 77               MOV    M,A      ;STORE IN TABLE
071A 219810           LXI    H,NOLA
071D 34               INR    M        ;INCREMENT NUMBER OF LABELS
              ;
              ; PROCESS OPCODE
              ;
071E CD6601   OPC:    CALL   ZBUF     ;ZERO WORKING BUFFER
0721 CD0D09           CALL   SBLK     ;SCAN TO OPCODE
0724 DA0608           JC     OERR     ;FOUND CARRIAGE RETURN
0727 CD7508           CALL   ALPS     ;PLACE OPCODE IN BUFFER
072A FE20             CPI    ' '      ;CHECK FOR BLANK AFTER OPCODE
072C DA650A           JC     OPCD     ;CR AFTER OPCODE
```

```
072F C2060B          JNZ     OERR     ;ERROR IF NO BLANK
0732 C3650A          JMP     OPCD     ;CHECK OPCODE
                ;
                ; THIS ROUTINE CHECKS THE CHARACTER AFTER A LABEL
                ; FOR A BLANK OR A COLON.
                ;
0735 2A9610     LCHK:   LHLD    PNTR
0738 7E                 MOV     A,M      ;GET CHARACTER AFTER LABEL
0739 FE20               CPI     ' '      ;CHECK FOR A BLANK
073B C8                 RZ               ;RETURN IF A BLANK
073C FE3A               CPI     ':'      ;CHECK FOR A COLON
073E C0                 RNZ
073F 23                 INX     H
0740 229610             SHLD    PNTR     ;SAVE POINTER
0743 C9                 RET
                ;
                ; PROCESS ANY PSEUDO OPS THAT NEED TO BE IN PASS 1
                ;
0744 CD0D09     PSU1:   CALL    SBLK     ;SCAN TO OPERAND
0747 1A                 LDAX    D        ;FETCH VALUE
0748 B7                 ORA     A        ;SET FLAGS
0749 CA6007             JZ      ORG1     ;ORG OPCODE
074C FA9007             JM      DAT1     ;DATA STATEMENT
074F E27507             JPO     EQU1     ;EQU OPCODE
0752 FE05               CPI     5
0754 DA8807             JC      RES1     ;RES OPCODE
0757 C20109             JNZ     EASS     ;JUMP IF END
                ; DO DW PSEUDO/OP
075A 0E02       AC01:   MVI     C,2      ;2 BYTE INSTRUCTION
075C AF                 XRA     A        ;GET A ZERO
075D C3F50A             JMP     OCN1     ;ADD VALUE TO PROGRAM CNTR
                ; DO ORG PSEUDO-OP
0760 CD970B     ORG1:   CALL    ASCN     ;GET OPERAND
0763 3A8210             LDA     OBUF     ;FETCH ERROR INDICATOR
0766 FE20               CPI     ' '      ;CHECK FOR AN ERROR
0768 C0                 RNZ
0769 229210             SHLD    ASPC     ;STORE NEW ORIGIN
076C 3AC710             LDA     IBUF     ;GET FIRST CHARACTER
076F FE20               CPI     ' '      ;CHECK FOR LABEL
0771 C8                 RZ               ;NO LABEL
0772 C38007             JMP     EQUS     ;CHANGE LABEL VALUE
                ; DO EQU PSEUDO-OP
0775 CD970B     EQU1:   CALL    ASCN     ;GET OPERAND
0778 3AC710             LDA     IBUF     ;FETCH 1ST CHARACTER
077B FE20               CPI     ' '      ;CHECK FOR LABEL
077D CA9F0C             JZ      ERRM     ;MISSING LABEL
0780 EB         EQUS:   XCHG
0781 2A9010             LHLD    TABA     ;SYMBOL TABLE ADDRESS
0784 72                 MOV     M,D      ;STORE LABEL VALUE
0785 23                 INX     H
0786 73                 MOV     M,E
0787 C9                 RET
                ; DO DS PSEUDO-OP
0788 CD970B     RES1:   CALL    ASCN     ;GET OPERAND
078B 44                 MOV     B,H
078C 4D                 MOV     C,L
078D C3ED07             JMP     RES21    ;ADD VALUE TO PROGRAM COUNTER
                ;
                ; DO DB PSEUDO-OP
                ;
0790 C3F407     DAT1:   JMP     DAT2A
                ;
                ; PERFORM PASS 2 OF THE ASSEMBLER
                ;
0793 218410     PAS2:   LXI     H,OBUF+2 ;SET OUTPUT BUFFER ADDRESS
0796 3A9310             LDA     ASPC+1   ;FETCH PC(HIGH)
```

```
0799 CD8902          CALL    BINH+3  ;CONVERT FOR OUTPUT
079C 23              INX     H
079D 3A9210          LDA     ASPC    ;FETCH PC(LOW)
07A0 CD8902          CALL    BINH+3  ;CONVERT FOR OUTPUT
07A3 23              INX     H
07A4 229E10          SHLD    OIND    ;SAVE OUTPUT ADDRESS
07A7 CD6601          CALL    ZBUF    ;CLEAR BUFFER
07AA 21C710          LXI     H,IBUF  ;INITIALIZE LINE POINTER
07AD 229610  PABL:   SHLD    PNTR    ;SAVE POINTER
07B0 7E              MOV     A,M     ;FETCH FIRST CHARACTER
07B1 FE20            CPI     ' '     ;CHECK FOR LABEL
07B3 CA1E07          JZ      OPC     ;GET OPCODE
07B6 FE2A            CPI     '*'     ;CHECK FOR COMMENT
07B8 C8              RZ              ;RETURN IF COMMENT
07B9 CD2008          CALL    SLAB    ;SCAN OFF LABEL
07BC DAC20C          JC      ERRL    ;ERROR IN LABEL
07BF CD3507          CALL    LCHK    ;CHECK FOR A BLANK OR COLON
07C2 C2C20C          JNZ     ERRL    ;ERROR IF NOT A BLANK
07C5 C31E07          JMP     OPC
             ;
             ;
             ;
             ; PROCESS PSEUDO OPS FOR PASS2
07C8 1A      PSU2:   LDAX    D
07C9 B7              ORA     A       ;SET FLAGS
07CA CA0C08          JZ      ORG2    ;ORG OPCODE
07CD FAF107          JM      DAT2    ;DATA OPCODE
07D0 E2FA07          JPO     EQU2    ;EQUATE PSEUDE-OP
07D3 FE05            CPI     5
07D5 DAE107          JC      RES2    ;RES OPCODE
07D8 C20109          JNZ     EASS    ;END OPCODE
             ; DO DW PSEUDO-OP
07DB CDE108  ACO2:   CALL    TYS6    ;GET VALUE
07DE C35A07          JMP     ACO1
             ; DO DS PSEUDO-OP
07E1 CD940B  RES2:   CALL    ASBL    ;GET OPERAND
07E4 44              MOV     B,H
07E5 4D              MOV     C,L
07E6 2A8C10          LHLD    BBUF+2  ;FETCH STORAGE COUNTER
07E9 09              DAD     B       ;ADD VALUE
07EA 228C10          SHLD    BBUF+2
07ED AF      RES21:  XRA     A       ;GET A ZERO
07EE C3F80A          JMP     OCN2
             ; DO DB PSEUDO-OP
07F1 CDA008  DAT2:   CALL    TYS5    ;GET OPERAND
07F4 AF      DAT2A:  XRA     A       ;MAKE A ZERO
07F5 0E01            MVI     C,1     ;BYTE COUNT
07F7 C3F50A          JMP     OCN1
             ;
             ; HANDLE EQUATES ON 2ND PASS.
             ;
07FA CD940B  EQU2:   CALL    ASBL    ;GET OPERAND INTO HL AND
                                     ;  FALL INTO NEXT ROUTINE
             ;
             ; STORE CONTENTS OF HL AS HEX ASCII AT OBUF+2.
             ;   ON RETURN, DE HOLDS VALUE WHICH WAS IN HL.
             ;
07FD EB      BINAD:  XCHG            ;PUT VALUE INTO DE
07FE 218410          LXI     H,OBUF+2 ;POINTER TO ADDR IN OBUF
0801 7A              MOV     A,D     ;STORE HI BYTE....
0802 CD8902          CALL    BINH+3
0805 23              INX     H
0806 7B              MOV     A,E     ;STORE LO BYTE...
0807 CD8902          CALL    BINH+3
080A 23              INX     H
080B C9              RET
```

```
                   ; DO ORG PSEUDO-OP
080C CD9408      ORG2:   CALL    ASBL      ;GET NEW ORIGIN
080F 3A8210              LDA     OBUF      ;GET ERROR INDICATOR
0812 FE20                CPI     ' '       ;CHECK FOR AN ERROR
0814 C0                  RNZ               ;DON'T MODIFY PC IF ERROR
0815 CDFD07              CALL    BINAD     ;STORE NEW ADDR IN OBUF
0818 2A9210              LHLD    ASPC      ;FETCH PC
081B EB                  XCHG
081C 229210              SHLD    ASPC      ;STORE NEW PC
081F 7D                  MOV     A,L
0820 93                  SUB     E         ;FORM DIFFERENCE OF ORIGINS
0821 5F                  MOV     E,A
0822 7C                  MOV     A,H
0823 9A                  SBB     D
0824 57                  MOV     D,A
0825 2A8C10              LHLD    BBUF+2    ;FETCH STORAGE POINTER
0828 19                  DAD     D         ;MODIFY
0829 228C10              SHLD    BBUF+2    ;SAVE
082C C9                  RET
                   ;
                   ; PROCESS 1 BYTE INSTRUCTIONS WITHOUT OPERANDS
                   ;
082D CDEE08      TYP1:   CALL    ASTO      ;STORE VALUE IN MEMORY
0830 C9                  RET
                   ;
                   ; PROCESS STAX AND LDAX INSTRUCTIONS
                   ;
0831 CD9408      TYP2:   CALL    ASBL      ;FETCH OPERAND
0834 C4810C              CNZ     ERRR      ;ILLEGAL REGISTER
0837 7D                  MOV     A,L       ;GET LOW ORDER OPERAND
0838 B7                  ORA     A         ;SET FLAGS
0839 CA5508              JZ      TY31      ;OPERAND = 0
083C FE02                CPI     2         ;OPERAND = 2
083E C4810C              CNZ     ERRR      ;ILLEGAL REGISTER
0841 C35508              JMP     TY31
                   ;
                   ; PROCESS PUSH,POP,INX,DCX,DAD INSTRUCTIONS
                   ;
0844 CD9408      TYP3:   CALL    ASBL      ;FETCH OPERAND
0847 C4810C              CNZ     ERRR      ;ILLEGAL REGISTER
084A 7D                  MOV     A,L       ;GET LOW ORDER OPERAND
084B 0F                  RRC               ;CHECK LOW ORDER BIT
084C DC810C              CC      ERRR      ;ILLEGAL REGISTER
084F 17                  RAL               ;RESTORE
0850 FE08                CPI     8
0852 D4810C              CNC     ERRR      ;ILLEGAL REGISTER
0855 07          TY31:   RLC               ;MULTIPLY BY 8
0856 17                  RAL
0857 17                  RAL
0858 47          TY32:   MOV     B,A
0859 1A                  LDAX    D         ;FETCH OPCODE BASE
085A 80                  ADD     B         ;FORM OPCODE
085B FE76                CPI     118       ;CHECK FOR MOV M,M
085D CC810C              CZ      ERRR      ;ILLEGAL REGISTER
0860 C32D08              JMP     TYP1
                   ;
                   ; PROCESS ACCUMULATOR, INR,DCR,MOV,RST INSTRUCTIONS
                   ;
0863 CD9408      TYP4:   CALL    ASBL      ;FETCH OPERAND
0866 C4810C              CNZ     ERRR      ;ILLEGAL REGISTER
0869 7D                  MOV     A,L       ;GET LOW ORDER OPERAND
086A FE08                CPI     8
086C D4810C              CNC     ERRR      ;ILLEGAL REGISTER
086F 1A                  LDAX    D         ;FETCH OPCODE BASE
0870 FE40                CPI     64        ;CHECK FOR MOV INSTRUCTION
0872 CA8108              JZ      TY41
```

```
0875 FEC7               CPI     199
0877 7D                 MOV     A,L
0878 CA5508             JZ      TY31     ;RST INSRUCTION
087B FA5808             JM      TY32     ;ACCUMULATOR INSTRUCTION
087E C35508             JMP     TY31     ;INR,DCR
              ; PROCESS MOV INSTRUCTION
0881 29       TY41:     DAD     H        ;MULTIPLY OPERAND BY 8
0882 29                 DAD     H
0883 29                 DAD     H
0884 85                 ADD     L        ;FORM OPCODE
0885 12                 STAX    D        ;SAVE OPCODE
0886 CD8F08             CALL    MPNT
0889 CD970B             CALL    ASCN
088C C4810C             CNZ     ERRR     ;INCREMENT POINTER
088F 7D                 MOV     A,L      ;FETCH LOW ORDER OPERAND
0890 FE08               CPI     8
0892 D4810C             CNC     ERRR     ;ILLEGAL REGISTER
0895 C35808             JMP     TY32
              ;
              ; PROCESS IMMEDIATE INSTRUCTIONS
              ; IMMEDIATE BYTE CAN BETWEEN -256 AND +255
              ; MVI INSTRUCTION IS A SPECIAL CASE AND CONTAINS
              ; 2 ARGUMENTS IN OPERAND
0898 FE06     TYP5:     CPI     6        ;CHECK FOR MVI INSTRUCTION
089A CCAD08             CZ      TY56
089D CDEE08             CALL    ASTO     ;STORE OBJECT BYTE
08A0 CD940B   TY55:     CALL    ASBL     ;GET IMMEDIATE ARGUMENT
08A3 3C                 INR     A
08A4 FE02               CPI     2        ;CHECK OPERAND FOR RANGE
08A6 D49A0C             CNC     ERRV     ;OPERAND OUT OF RANGE
08A9 7D                 MOV     A,L
08AA C32D08             JMP     TYP1
              ;
              ; FETCH 1ST ARG FOR MVI AND LXI INSTRUCTIONS
              ;
08AD CD940B   TY56:     CALL    ASBL     ;FETCH ARG
08B0 C4810C             CNZ     ERRR     ;ILLEGAL REGISTER
08B3 7D                 MOV     A,L      ;GET LOW ORDER ARGUMENT
08B4 FE08               CPI     8
08B6 D4810C             CNC     ERRR     ;ILLEGAL REGISTER
08B9 29                 DAD     H
08BA 29                 DAD     H
08BB 29                 DAD     H
08BC 1A                 LDAX    D        ;FETCH OPCODE BASE
08BD 85                 ADD     L        ;FOR OPCODE
08BE 5F                 MOV     E,A      ;SAVE OBJECT BYTE
08BF 2A9610   MPNT:     LHLD    PNTR     ;FETCH POINTER
08C2 7E                 MOV     A,M      ;FETCH CHARACTER
08C3 FE2C               CPI     ','      ;CHECK FOR COMMA
08C5 23                 INX     H        ;INCREMENT POINTER
08C6 229610             SHLD    PNTR
08C9 C28A0C             JNZ     ERRS     ;SYNTAX ERROR IF NO COMMA
08CC 7B                 MOV     A,E
08CD C9                 RET
              ;
              ; PROCESS 3 BYTE INSTRUCTIONS
              ; LXI INSTRUCTION IS A SPECIAL CASE
              ;
08CE FE01     TYP6:     CPI     1        ;CHECK FOR LXI INSTRUCTION
08D0 C2DE08             JNZ     TY6      ;JUMP IF NOT LXI
08D3 CDAD08             CALL    TY56     ;GET REGISTER
08D6 E608               ANI     08H      ;CHECK FOR ILLEGAL REGISTER
08D8 C4810C             CNZ     ERRR     ;REGISTER ERROR
08DB 7B                 MOV     A,E      ;GET OPCODE
08DC E6F7               ANI     0F7H     ;CLEAR BIT IN ERROR
08DE CDEE08   TY6:      CALL    ASTO     ;STORE OBJECT BYTE
```

```
08E1 CD940B    TYS6:   CALL    ASBL     ;FETCH OPERAND
08E4 7D                MOV     A,L
08E5 54                MOV     D,H
08E6 CDEE08            CALL    ASTO     ;STORE 2ND BYTE
08E9 7A                MOV     A,D
08EA C32D08            JMP     TYP1
08ED C9                RET
                ;
                ; THIS ROUTINE IS USED TO STORE OBJECT CODE PRODUCED
                ; BY THE ASSEMBLER DURING PASS 2 INTO MEMORY
                ;
08EE 2A8C10    ASTO:   LHLD    BBUF+2   ;FETCH STORAGE ADDRESS
08F1 77                MOV     M,A      ;STORE OBJECT BYTE
08F2 23                INX     H        ;INCREMENT LOCATION
08F3 228C10            SHLD    BBUF+2
08F6 2A9E10            LHLD    OIND     ;FETCH OUTPUT ADDRESS
08F9 23                INX     H
08FA CD8902            CALL    BINH+3   ;CONVERT OBJECT BYTE
08FD 229E10            SHLD    OIND
0900 C9                RET
                ;
                ; GET HERE WHEN END PSEUDO-OP IS FOUND OR WHEN
                ; END-OF-FILE OCCURS IN SOURCE FILE. CONTROL IS SET
                ; FOR EITHER PASS 2 OR ASSEMBLY TERMINATOR IF FINISHED.
                ;
0901 3A9410    EASS:   LDA     PASI     ;FETCH PASS INDICATOR
0904 B7                ORA     A        ;SET FLAGS
0905 C26700            JNZ     EOR      ;JUMP IF FINISHED
0908 3E01              MVI     A,1      ;PASS INDICATOR FOR 2ND PASS
090A C37A06            JMP     ASM3     ;DO 2ND PASS
                ;
                ; THIS ROUTINE SCANS THROUGH A CHARACTER STRING UNTIL
                ; THE FIRST NON-BLANK CHARACTER IS FOUND
                ;
                ; ON RETURN, CARRY SET INDICATES A CARRIAGE RETURN
                ; AS FIRST NON-BLANK CHARACTER.
                ;
090D 2A9610    SBLK:   LHLD    PNTR     ;FETCH ADDRESS
0910 7E        SBL1:   MOV     A,M      ;FETCH CHARACTER
0911 FE20              CPI     ' '      ;CHECK FOR A BLANK
0913 C0                RNZ              ;RETURN IF NON-BLANK
0914 23        SBL2:   INX     H        ;INCREMENT
0915 229610            SHLD    PNTR     ;SAVE POINTER
0918 C31009            JMP     SBL1
                ;
                ;
                ; THIS ROUTINE IS USED TO CHECK THE CONDITION
                ; CODE MNEMONICS FOR CONDITIONAL JUMPS, CALLS,
                ; AND RETURNS.
                ;
091B 217F10    COND:   LXI     H,ABUF+1
091E 227410            SHLD    ADDS
0921 0602              MVI     B,2      ;2 CHARACTERS
0923 CD500A            CALL    COPC
0926 C9                RET
                ;
                ;
                ; THE FOLLOWING IS THE OPCODE TABLE
                ;
                ;
0927 4F5247    OTAB:   DB      'ORG'
092A 00                DB      0
092B 00                DB      0
092C 455155            DB      'EQU'
092F 00                DB      0
0930 01                DB      1
```

```
0931 4442          DB    'DB'
0933 00            DB    0
0934 00            DB    0
0935 FF            DB    -1
0936 4453          DB    'DS'
0938 00            DB    0
0939 00            DB    0
093A 03            DB    3
093B 4457          DB    'DW'
093D 00            DB    0
093E 00            DB    0
093F 05            DB    5
0940 454E44        DB    'END'
0943 00            DB    0
0944 06            DB    6
0945 00            DB    0
0946 484C54        DB    'HLT'
0949 76            DB    118
094A 524C43        DB    'RLC'
094D 07            DB    7
094E 525243        DB    'RRC'
0951 0F            DB    15
0952 52414C        DB    'RAL'
0955 17            DB    23
0956 524152        DB    'RAR'
0959 1F            DB    31
095A 524554        DB    'RET'
095D C9            DB    201
095E 434D41        DB    'CMA'
0961 2F            DB    47
0962 535443        DB    'STC'
0965 37            DB    55
0966 444141        DB    'DAA'
0969 27            DB    39
096A 434D43        DB    'CMC'
096D 3F            DB    63
096E 4549          DB    'EI'
0970 00            DB    0
0971 FB            DB    251
0972 4449          DB    'DI'
0974 00            DB    0
0975 F3            DB    243
0976 4E4F50        DB    'NOP'
0979 00            DB    0
097A 00            DB    0
097B 58434847      DB    'XCHG'
097F EB            DB    235
0980 5854484C      DB    'XTHL'
0984 E3            DB    227
0985 5350484C      DB    'SPHL'
0989 F9            DB    249
098A 5043484C      DB    'PCHL'
098E E9            DB    233
098F 00            DB    0
0990 53544158      DB    'STAX'
0994 02            DB    2
0995 4C444158      DB    'LDAX'
0999 0A            DB    10
099A 00            DB    0
099B 50555348      DB    'PUSH'
099F C5            DB    197
09A0 504F50        DB    'POP'
09A3 00            DB    0
09A4 C1            DB    193
09A5 494E58        DB    'INX'
09A8 00            DB    0
```

```
09A9 03          DB    3
09AA 444358      DB    'DCX'
09AD 00          DB    0
09AE 0B          DB    11
09AF 444144      DB    'DAD'
09B2 00          DB    0
09B3 09          DB    9
09B4 00          DB    0
09B5 494E52      DB    'INR'
09B8 04          DB    4
09B9 444352      DB    'DCR'
09BC 05          DB    5
09BD 4D4F56      DB    'MOV'
09C0 40          DB    64
09C1 414444      DB    'ADD'
09C4 80          DB    128
09C5 414443      DB    'ADC'
09C8 88          DB    136
09C9 535542      DB    'SUB'
09CC 90          DB    144
09CD 534242      DB    'SBB'
09D0 98          DB    152
09D1 414E41      DB    'ANA'
09D4 A0          DB    160
09D5 585241      DB    'XRA'
09D8 A8          DB    168
09D9 4F5241      DB    'ORA'
09DC B0          DB    176
09DD 434D50      DB    'CMP'
09E0 B8          DB    184
09E1 525354      DB    'RST'
09E4 C7          DB    199
09E5 00          DB    0
09E6 414449      DB    'ADI'
09E9 C6          DB    198
09EA 414349      DB    'ACI'
09ED CE          DB    206
09EE 535549      DB    'SUI'
09F1 D6          DB    214
09F2 534249      DB    'SBI'
09F5 DE          DB    222
09F6 414E49      DB    'ANI'
09F9 E6          DB    230
09FA 585249      DB    'XRI'
09FD EE          DB    238
09FE 4F5249      DB    'ORI'
0A01 F6          DB    246
0A02 435049      DB    'CPI'
0A05 FE          DB    254
0A06 494E        DB    'IN'
0A08 00          DB    0
0A09 DB          DB    219
0A0A 4F5554      DB    'OUT'
0A0D D3          DB    211
0A0E 4D5649      DB    'MVI'
0A11 06          DB    6
0A12 00          DB    0
0A13 4A4D50      DB    'JMP'
0A16 00          DB    0
0A17 C3          DB    195
0A18 43414C4C    DB    'CALL'
0A1C CD          DB    205
0A1D 4C5849      DB    'LXI'
0A20 00          DB    0
0A21 01          DB    1
0A22 4C4441      DB    'LDA'
```

```
0A25 00            DB      0
0A26 3A            DB      58
0A27 535441        DB      'STA'
0A2A 00            DB      0
0A2B 32            DB      50
0A2C 53484C44      DB      'SHLD'
0A30 22            DB      34
0A31 4C484C44      DB      'LHLD'
0A35 2A            DB      42
0A36 00            DB      0
             ; CONDITION CODE TABLE
0A37 4E5A          DB      'NZ'
0A39 00            DB      0
0A3A 5A            DB      'Z'
0A3B 00            DB      0
0A3C 08            DB      8
0A3D 4E43          DB      'NC'
0A3F 10            DB      16
0A40 43            DB      'C'
0A41 00            DB      0
0A42 18            DB      24
0A43 504F          DB      'PO'
0A45 20            DB      32
0A46 5045          DB      'PE'
0A48 28            DB      40
0A49 50            DB      'P'
0A4A 00            DB      0
0A4B 30            DB      48
0A4C 4D            DB      'M'
0A4D 00            DB      0
0A4E 38            DB      56
0A4F 00            DB      0
             ;
             ; THIS ROUTINE IS USED TO CHECK A GIVEN OPCODE
             ; AGAINST THE LEGAL OPCODES IN THE OPCODE TABLE.
             ;
0A50 2A7410  COPC:  LHLD    ADDS
0A53 1A             LDAX    D         ;FETCH CHARACTER
0A54 B7             ORA     A         ;SET FLAGS
0A55 CA620A         JZ      COP1      ;JUMP IF TERMINATION CHARACTER
0A58 48             MOV     C,B
0A59 CD5301         CALL    SEAR
0A5C 1A             LDAX    D
0A5D C8             RZ                ;RETURN IF MATCH
0A5E 13             INX     D         ; NEXT STRING
0A5F C3500A         JMP     COPC      ;CONTINUE SEARCH
0A62 3C      COP1:  INR     A         ;CLEAR ZERO FLAG
0A63 13             INX     D         ;INCREMENT ADDRESS
0A64 C9             RET
             ;
             ; THIS ROUTINE CHECKS THE LEGAL OPCODES IN BOTH PASS 1
             ; AND PASS 2.  IN PASS 1 THE PROGRAM COUNTER IS INCRE-
             ; MENTED BY THE CORRECT NUMBER OF BYTES.  AN ADDRESS IS
             ; ALSO SET SO THAT AN INDEXED JUMP CAN BE MADE TO
             ; PROCESS THE OPCODE FOR PASS 2.
             ;
             ;
0A65 217E10  OPCD:  LXI     H,ABUF    ;GET ADDRESS
0A68 227410         SHLD    ADDS
0A6B 112709         LXI     D,OTAB    ;OPCODE TABLE ADDRESS
0A6E 0604           MVI     B,4       ;CHARACTER COUNT
0A70 CD500A         CALL    COPC      ;CHECK OPCODES
0A73 CA0E0B         JZ      PSEU      ;JUMP IF A PSEUDO-OP
0A76 05             DCR     B         ;3 CHARACTER OPCODES
0A77 CD500A         CALL    COPC
0A7A CA810A         JZ      OP1
```

```
0A7D 04                 INR     B       ;4 CHARACTER  OPCODES
0A7E CD500A             CALL    COPC
0A81 212D08     OP1:    LXI     H,TYP1  ;TYPE 1 INSTRUCTIONS
0A84 0E01       OP2:    MVI     C,1     ;1 BYTE INSTRUCTIONS
0A86 CAE10A             JZ      OCNT
                ;
0A89 CD500A     OPC2:   CALL    COPC    ;CHECK FOR STAX,LDAX
0A8C 213108             LXI     H,TYP2
0A8F CA840A             JZ      OP2
0A92 CD500A             CALL    COPC    ;CHECK FOR PUSH,POP,INX
                ;               ;       ;DCX AND DAD
0A95 214408             LXI     H,TYP3
0A98 CA840A             JZ      OP2
0A9B 05                 DCR     B       ;3 CHAR OPCODES
0A9C CD500A             CALL    COPC    ;ACCUMULATOR INSTRUCTIONS,
                ;               ;       ;INR,DCR,MOV,RST
0A9F 216308             LXI     H,TYP4
0AA2 CA840A             JZ      OP2
                ;
0AA5 CD500A     OPC3:   CALL    COPC    ;IMMEDIATE INSTRUCTIONS
0AA8 219808             LXI     H,TYP5
0AAB 0E02               MVI     C,2     ;2 BYTE INSTRUCTIONS
0AAD CAE10A             JZ      OCNT
0AB0 04                 INR     B       ;4 CHAR OPCODES
0AB1 CD500A             CALL    COPC    ;JMP,CALL,LXI,LDA,STA,
                ;               ;       ;LHLD,SHLD OPCODES
0AB4 CADC0A             JZ      OP4
0AB7 CD1B09             CALL    COND    ;CONDITIONAL INSTRUCTIONS
0ABA C2060B             JNZ     OERR    ;ILLEGAL OPCODE
0ABD C6C0               ADI     192     ;ADD BASE VALUE TO RETURN
0ABF 57                 MOV     D,A
0AC0 0603               MVI     B,3     ;3 CHARACTER OPCODES
0AC2 3A7E10             LDA     ABUF    ;FETCH FIRST CHARACTER
0AC5 4F                 MOV     C,A     ;SAVE CHARACTER
0AC6 FE52               CPI     'R'     ;CONDITIONAL RETURN
0AC8 7A                 MOV     A,D
0AC9 CA810A             JZ      OP1
0ACC 79                 MOV     A,C
0ACD 14                 INR     D       ;FORM CONDITIONAL JUMP
0ACE 14                 INR     D
0ACF FE4A               CPI     'J'     ;CONDITIONAL JUMP
0AD1 CADB0A             JZ      OPAD
0AD4 FE43               CPI     'C'     ;CONDITIONAL CALL
0AD6 C2060B             JNZ     OERR    ;ILLEGAL OPCODE
0AD9 14                 INR     D       ;FORM CONDIITIONAL CALL
0ADA 14                 INR     D
0ADB 7A         OPAD:   MOV     A,D     ;GET OPCODE
0ADC 21CE08     OP4:    LXI     H,TYP6
0ADF 0E03       OP5:    MVI     C,3     ;3 BYTE INSTRUCTION
0AE1 329D10     OCNT:   STA     TEMP    ;SAVE OPCODE
                ;
                ; CHECK FOR OPCODE ONLY CONTAINING THE CORRECT NUMBER OF
                ; CHARACTERS. THUS ADDQ, SAY, WOULD GIVE AN ERROR
                ;
0AE4 3E7E               MVI     A,ABUF AND 0FFH ;LOAD BUFFER ADDRESS
0AE6 80                 ADD     B       ;ADD LENGTH OF OPCODE
0AE7 5F                 MOV     E,A
0AE8 3E10               MVI     A,ABUF/256
0AEA CE00               ACI     0       ;GET HIGH ORDER ADDRESS
0AEC 57                 MOV     D,A
0AED 1A                 LDAX    D       ;FETCH CHARACTER AFTER OPCODE
0AEE B7                 ORA     A       ;IT SHOULD BE ZERO
0AEF C2060B             JNZ     OERR    ;OPCODE ERROR
0AF2 3A9410             LDA     PASI    ;FETCH PASS INDICATOR
0AF5 0600       OCN1:   MVI     B,0
0AF7 EB                 XCHG
```

```
0AF8 2A9210     OCN2:   LHLD    ASPC     ;FETCH PROGRAM COUNTER
0AFB 09                 DAD     B        ;ADD IN BYTE COUNT
0AFC 229210             SHLD    ASPC     ;STORE PC
0AFF B7                 ORA     A        ;WHICH PASS?
0B00 C8                 RZ               ;RETURN IF PASS 1
0B01 3A9D10             LDA     TEMP     ;FETCH OPCODE
0B04 EB                 XCHG
0B05 E9                 PCHL
                ;
0B06 21AD0C     OERR:   LXI     H,ERRO   ;SET ERROR ADDRESS
0B09 0E03               MVI     C,3      ;LEAVE 3 BYTES FOR PATCH
0B0B C3F20A             JMP     OCN1-3
                ;
0B0E 218210     PSEU:   LXI     H,ABUF+4         ;SET BUFFER ADDRESS
0B11 7E                 MOV     A,M      ;FETCH CHARACTER AFTER OPCODE
0B12 B7                 ORA     A        ;SHOULD BE A ZERO
0B13 C2060B             JNZ     OERR
0B16 3A9410             LDA     PASI     ;FETCH PASS INDICATOR
0B19 B7                 ORA     A
0B1A CA4407             JZ      PSU1
0B1D C3C807             JMP     PSU2
                ;
                ;
                ; THIS ROUTINE IS USED TO PROCESS LABELS.
                ; IT CHECKS TO SEE IF A LABEL IS IN THE SYMBOL TABLE
                ; OR NOT.  ON RETURN, Z=1 INDICATES A MATCH WAS FOUND
                ; AND H,L CONTAIN THE VALUE ASSOCIATED WITH THE LABEL.
                ; THE REGISTER NAMES A, B, C, D, E, H, L, P, AND S ARE
                ; PRE-DEFINED AND NEED NOT BE ENTERED BY THE USER.
                ; ON RETURN, C=1 INDICATES A LABEL ERROR.
                ;
0B20 FE41       SLAB:   CPI     'A'      ;CHECK FOR LEGAL CHAR
0B22 D8                 RC
0B23 FE5B               CPI     'Z'+1    ;CHECK FOR ILLEGAL CHAR
0B25 3F                 CMC
0B26 D8                 RC               ;RETURN IF ILLEGAL CHAR
0B27 CD750B             CALL    ALPS     ;PLACE SYMBOL IN BUFFER
0B2A 217E10             LXI     H,ABUF   ;SET BUFFER ADDRESS
0B2D 227410             SHLD    ADDS     ;SAVE ADDDRESS
0B30 05                 DCR     B        ;CHECK IF ONE CHARACTER
0B31 C2440B             JNZ     SLA1
                ; CHECK IF PREDEFINED REGISTER NAME
0B34 04                 INR     B        ;SET B=1
0B35 11600B             LXI     D,RTAB   ;REGISTER TABLE ADDRESS
0B38 CD500A             CALL    COPC     ;CHECK NAME OF REGISTER
0B3B C2440B             JNZ     SLA1     ;NOT A PREDEFINED REGISTER
0B3E 6F                 MOV     L,A      ;SET VALUE(HIGH)
0B3F 2600               MVI     H,0
0B41 C35A0B             JMP     SLA2
0B44 3A9810     SLA1:   LDA     NOLA     ;FETCH SYMBOL COUNT
0B47 47                 MOV     B,A
0B48 111A11             LXI     D,SYMT   ;SET SYMBOL TABLE ADDRESS
0B4B B7                 ORA     A        ;ARE THERE ANY LABELS?
0B4C CA5D0B             JZ      SLA3     ;JUMP IF NO LABELS
0B4F 3E05               MVI     A,LLAB   ;FETCH LENGTH OF LABEL
0B51 329510             STA     NCHR
0B54 CD3C01             CALL    COMS     ;CHECK TABEL
0B57 4C                 MOV     C,H      ;SWAP H AND L
0B58 65                 MOV     H,L
0B59 69                 MOV     L,C
0B5A 37         SLA2:   STC              ;SET CARRY
0B5B 3F                 CMC              ;CLEAR CARRY
0B5C C9                 RET              ;RETURN
0B5D 3C         SLA3:   INR     A        ;CLEAR ZERO FLAG
0B5E B7                 ORA     A        ;CLEAR CARRY
0B5F C9                 RET
```

```
                ;
                ; PREDEFINE REGISTER VALUES IN THIS TABLE
                ;
0B60 41         RTAB:   DB      'A'
0B61 07                 DB      7
0B62 42                 DB      'B'
0B63 00                 DB      0
0B64 43                 DB      'C'
0B65 01                 DB      1
0B66 44                 DB      'D'
0B67 02                 DB      2
0B68 45                 DB      'E'
0B69 03                 DB      3
0B6A 48                 DB      'H'
0B6B 04                 DB      4
0B6C 4C                 DB      'L'
0B6D 05                 DB      5
0B6E 4D                 DB      'M'
0B6F 06                 DB      6
0B70 50                 DB      'P'
0B71 06                 DB      6
0B72 53                 DB      'S'
0B73 06                 DB      6
0B74 00                 DB      0           ;END OF TABLE INDICATOR
                ;
                ; THIS ROUTINE SCANS THE INPUT LINE AND PLACES THE
                ; OPCODES AND LABELS IN THE BUFFER.  THE SCAN TERMINATES
                ; WHEN A CHARACTER OTHER THAN 0-9 OR A-Z IS FOUND.
                ;
0B75 0600       ALPS:   MVI     B,0         ;SET COUNT
0B77 12         ALP1:   STAX    D           ;STORE CHARACTER IN BUFFER
0B78 04                 INR     B           ;INCREMENT COUNT
0B79 78                 MOV     A,B         ;FETCH COUNT
0B7A FE08               CPI     11          ;MAXIMUM BUFFER SIZE
0B7C D0                 RNC                 ;RETURN IF BUFFER FILLED
0B7D 13                 INX     D           ;INCREMENT BUFFER
0B7E 23                 INX     H           ;INCREMENT INPUT POINTER
0B7F 229610             SHLD    PNTR        ;SAVE LINE POINTER
0B82 7E                 MOV     A,M         ;FETCH CHARACTER
0B83 FE30               CPI     '0'         ;CHECK FOR LEGAL CHARACTERS
0B85 D8                 RC
0B86 FE3A               CPI     '9'+1
0B88 DA770B             JC      ALP1
0B8B FE41               CPI     'A'
0B8D D8                 RC
0B8E FE5B               CPI     'Z'+1
0B90 DA770B             JC      ALP1
0B93 C9                 RET
                ; THIS ROUTINE IS USED TO SCAN THROUGH THE INPUT LINE
                ; TO FETCH THE VALUE OF THE OPERAND FIELD.  ON RETURN,
                ; THE VALUE OF THE OPERAND IS CONTAINED IN REG'S H,L.
                ;
0B94 CD0D09     ASBL:   CALL    SBLK        ;GET FIRST ARGUMENT
0B97 210000     ASCN:   LXI     H,0         ;GET A ZERO
0B9A 229A10             SHLD    OPRD        ;INITIALIZE OPERAND
0B9D 24                 INR     H
0B9E 229B10             SHLD    OPRI-1      ;INITIALIZE OPERAND INDICATOR
0BA1 2A9610     NXT1:   LHLD    PNTR        ;FETCH SCAN POINTER
0BA4 2B                 DCX     H
0BA5 CD6601             CALL    ZBUF        ;CLEAR BUFFER
0BA8 329910             STA     SIGN        ;ZERO SIGN INDICATOR
0BAB 23         NXT2:   INX     H           ;INCREMENT POINTER
0BAC 7E                 MOV     A,M         ;FETCH NEXT CHARACTER
0BAD FE21               CPI     ' '+1
0BAF DA530C             JC      SEND        ;JUMP IF CR OR BLANK
0BB2 FE2C               CPI     ','         ;FIELD SEPARATOR
```

```
0B84 CA530C             JZ       SEND
                ; CHECK FOR OPERATORS
0B87 FE2B               CPI      '+'       ;CHECK FOR PLUS
0B89 CAC40B             JZ       ASC1
0B8C FE2D               CPI      '-'       ;CHECK FOR MINUS
0B8E C2D40B             JNZ      ASC2
0BC1 329910             STA      SIGN
0BC4 3A9C10     ASC1:   LDA      OPRI      ;FETCH OPERAND INDICATOR
0BC7 FE02               CPI      2         ;CHECK FOR TWO OPERATORS
0BC9 CA8A0C             JZ       ERRS      ;SYNTAX ERROR
0BCC 3E02               MVI      A,2
0BCE 329C10             STA      OPRI      ;SET INDICATOR
0BD1 C3AB0B             JMP      NXT2
                ;CHECK FOR OPERANDS
0BD4 4F         ASC2:   MOV      C,A       ;SAVE CHARACTER
0BD5 3A9C10             LDA      OPRI      ;GET INDICATOR
0BD8 B7                 ORA      A         ;CHECK FOR TWO OPERANDS
0BD9 CA8A0C             JZ       ERRS      ;SYNTAX ERROR
0BDC 79                 MOV      A,C
0BDD FE24               CPI      '$'       ;LC EXPRESSION
0BDF C2EC0B             JNZ      ASC3
0BE2 23                 INX      H         ;INCREMENT POINTER
0BE3 229610             SHLD     PNTR      ;SAVE POINTER
0BE6 2A9210             LHLD     ASPC      ;FETCH LOCATION COUNTER
0BE9 C3280C             JMP      AVAL
                ;CHECK FOR ASCII CHARACTERS
0BEC FE27       ASC3:   CPI      27H       ;CHECK FOR SINGLE QUOTE
0BEE C2180C             JNZ      ASC5      ;JUMP IF NOT QUOTE
0BF1 110000             LXI      D,0       ;GET A ZERO
0BF4 0E03               MVI      C,3       ;CHARACTER COUNT
0BF6 23         ASC4:   INX      H         ;BUMP POINTER
0BF7 229610             SHLD     PNTR      ;SAVE
0BFA 7E                 MOV      A,M       ;FETCH NEXT CHARACTER
0BFB FE0D               CPI      ASCR      ;IS IT A CR?
0BFD CAA80C             JZ       ERRA      ;ARGUMENT ERROR
0C00 FE27               CPI      27H       ;IS IT QUOTE
0C02 C20F0C             JNZ      SSTR
0C05 23                 INX      H         ;INCREMENT POINTER
0C06 229610             SHLD     PNTR      ;SAVE
0C09 7E                 MOV      A,M       ;FETCH NEXT CHAR
0C0A FE27               CPI      27H       ;CHECK FOR 2 QUOTES IN A ROW
0C0C C2290C             JNZ      AVAL+1    ;TERMINAL QUOTE
0C0F 0D         SSTR:   DCR      C         ;CHECK COUNT
0C10 CAA80C             JZ       ERRA      ;TOO MANY CHARACTERS
0C13 53                 MOV      D,E
0C14 5F                 MOV      E,A       ;SET CHARACTER IN BUFFER
0C15 C3F60B             JMP      ASC4
0C18 FE30       ASC5:   CPI      '0'       ;CHECK FOR NUMERIC
0C1A DAA80C             JC       ERRA      ;ILLEGAL CHARACTER
0C1D FE3A               CPI      '9'+1
0C1F D2470C             JNC      ALAB
0C22 CD630C             CALL     NUMS      ;GET NUMERIC VALUE
0C25 DAA80C             JC       ERRA      ;ARGUMENT ERROR
0C28 EB         AVAL:   XCHG
0C29 2A9A10             LHLD     OPRD      ;FETCH OPERAND
0C2C AF                 XRA      A         ;GET A ZERO
0C2D 329C10             STA      OPRI      ;STOR IN OPERAND INDICATOR
0C30 3A9910             LDA      SIGN      ;GET SIGN INDICATOR
0C33 B7                 ORA      A         ;SET FLAGS
0C34 C23E0C             JNZ      ASUB
0C37 19                 DAD      D         ;FORM RESULT
0C38 229A10     ASC7:   SHLD     OPRD      ;SAVE RESULT
0C3B C3A10B             JMP      NXT1
0C3E 7D         ASUB:   MOV      A,L
0C3F 93                 SUB      E
0C40 6F                 MOV      L.A
```

```
0C41 7C                  MOV     A,H
0C42 9A                  SBB     D
0C43 67                  MOV     H,A
0C44 C3380C              JMP     ASC7
0C47 CD2008    ALAB:     CALL    SLAB
0C4A CA280C              JZ      AVAL
0C4D DAA80C              JC      ERRA     ;ILLEGAL SYMBOL
0C50 C3950C              JMP     ERRU     ;UNDEFINED SYMBOL
               ;
               ; GET HERE WHEN TERMINATING CHARACTER IS FOUND.
               ; CHECK FOR LEADING FIELD SEPARATOR.
               ;
0C53 3A9C10    SEND:     LDA     OPRI     ;FETCH OPERAND INDICATOR
0C56 B7                  ORA     A        ;SET FLAGS
0C57 C28A0C              JNZ     ERRS     ;SYNTAX ERROR
0C5A 2A9A10              LHLD    OPRD
0C5D 7C        SEN1:     MOV     A,H      ;GET HIGH ORDER BYTE
0C5E 119010              LXI     D,TEMP   ;GET ADDRESS
0C61 B7                  ORA     A        ;SET FLAGS
0C62 C9                  RET
               ;
               ; GET A NUMERIC VALUE WHICH IS EITHER HEXADECIMAL OR
               ; DECIMAL.  ON RETURN, CARRY SET INDICATES AN ERROR.
               ;
0C63 CD750B    NUMS:     CALL    ALPS     ;GET NUMERIC
0C66 1B                  DCX     D
0C67 1A                  LDAX    D        ;GET LAST CHARACTER
0C68 017E10              LXI     B,ABUF   ;SET BUFFER ADDRESS
0C6B FE48                CPI     'H'      ;IS IT HEXADECIMAL?
0C6D CA7B0C              JZ      NUM2
0C70 FE44                CPI     'D'      ;IS IT DECIMAL
0C72 C2770C              JNZ     NUM1
0C75 AF                  XRA     A        ;GET A ZERO
0C76 12                  STAX    D        ;CLEAR D FROM BUFFER
0C77 CD0102    NUM1:     CALL    ADEC     ;CONVERT DECIMAL VALUE
0C7A C9                  RET
0C7B AF        NUM2:     XRA     A        ;GET A ZERO
0C7C 12                  STAX    D        ;CLEAR H FROM BUFFER
0C7D CD1802              CALL    AHEX
0C80 C9                  RET
               ;
               ; PROCESS REGISTER ERROR
0C81 3E52      ERRR:     MVI     A,'R'    ;GET INDICATOR
0C83 210000              LXI     H,0      ;GET A 0
0C86 328210              STA     OBUF     ;SET IN OUTPUT BUFFER
0C89 C9                  RET
               ;PROCESS SYNTAX ERROR
0C8A 3E53      ERRS:     MVI     A,'S'    ;GET INDICATOR
0C8C 328210              STA     OBUF     ;STORE IN OUTPUT BUFFER
0C8F 210000              LXI     H,0
0C92 C35D0C              JMP     SEN1
               ;PROCESS UNDEFINED SYMBOL ERROR
0C95 3E55      ERRU:     MVI     A,'U'    ;GET INDICATOR
0C97 C38C0C              JMP     ERRS+2
               ;PROCESS VALUE ERROR
0C9A 3E56      ERRV:     MVI     A,'V'    ;GET INDICATOR
0C9C C3830C              JMP     ERRR+2
               ;PROCESS MISSING LABEL ERROR
0C9F 3E4D      ERRM:     MVI     A,'M'    ;GET INDICATOR
0CA1 328210              STA     OBUF     ;STORE IN OUTPUT BUFFER
0CA4 CD0206              CALL    AOU1     ;DISPLAY ERROR
0CA7 C9                  RET
               ;PROCESS ARGUMENT ERROR
0CA8 3E41      ERRA:     MVI     A,'A'    ;GET INDICATOR
0CAA C38C0C              JMP     ERRS+2
               ; PROCESS OPCODE ERROR
```

```
                  ; STORE 3 BYTES OF ZERO IN OBJECT CODE TO PROVIDE
                  ; FOR A PATCH.
0CAD 3E4F         ERRO:   MVI     A,'O'    ;GET INDICATOR
0CAF 32B210               STA     OBUF     ;STORE IN OUTPUT BUFFER
0CB2 3A9410               LDA     PASI     ;FETCH PASS INDICATOR
0CB5 B7                   ORA     A        ;WHICH PASS
0CB6 C8                   RZ               ;RETURN IF PASS1
0CB7 0E03                 MVI     C,3      ;NEED 3 BYTES
0CB9 AF           ERO1:   XRA     A        ;GET A ZERO
0CBA CDEE08               CALL    ASTO     ;PUT IN LISTING AND MEMORY
0CBD 0D                   DCR     C
0CBE C2B90C               JNZ     ERO1
0CC1 C9                   RET
                  ; PROCESS LABEL ERROR
0CC2 3E4C         ERRL:   MVI     A,'L'    ;GET INDICATOR
0CC4 C3AF0C               JMP     ERRO+2
                  ;PROCESS DUPLICATE LABEL ERROR
0CC7 3E44         ERRD:   MVI     A,'D'    ;GET ERROR INDICATOR
0CC9 32B210               STA     OBUF     ;STORE IN OUTPUT BUFFER
0CCC CDC506               CALL    AOUT     ;DISPLAY ERROR
0CCF C31E07               JMP     OPC      ;PROCESS OPCODE
                  ;
                  ;
                  ; THIS ROUTINE SETS OR CLEARS BREAKPOINTS
                  ;
0CD2 3A7E10       BREAK:  LDA     ABUF     ;CHECK FOR AN ARG
0CD5 B7                   ORA     A
0CD6 CA140D               JZ      CLRB     ;IF NO ARG, GO CLEAR BREAKPOINTS
0CD9 1608                 MVI     D,NBR    ;ELSE, GET NUMBER OF BREAKPOINTS
0CDB 210C10               LXI     H,BRT    ;AND ADDR OF TABLE
0CDE 7E           B1:     MOV     A,M      ;GET HI BYTE OF ENTRY
0CDF 23                   INX     H.
0CE0 46                   MOV     B,M      ;GET LO BYTE OF ENTRY
0CE1 B0                   ORA     B        ;CHECK FOR EMPTY ENTRY
0CE2 CAEE0C               JZ      B2       ;BRANCH IF EMPTY
0CE5 23                   INX     H        ;ELSE GO ON TO NEXT ENTRY
0CE6 23                   INX     H
0CE7 15                   DCR     D        ;BUMP COUNT
0CE8 C2DE0C               JNZ     B1       ;AND TRY AGAIN
0CEB C35A04               JMP     WHAT     ;OOPS! NO ROOM
0CEE 2B           B2:     DCX     H
0CEF EB                   XCHG
0CF0 2A8A10               LHLD    BBUF     ;GET ADDRESS
0CF3 EB                   XCHG             ;IN D,E
0CF4 7A                   MOV     A,D      ;CHECK FOR ADDR > 11D
0CF5 B7                   ORA     A
0CF6 C2FF0C               JNZ     B3
0CF9 7B                   MOV     A,E
0CFA FE0B                 CPI     11
0CFC DA5A04               JC      WHAT     ;OOPS. TOO LOW
0CFF 72           B3:     MOV     M,D      ;SAVE ADDRESS
0D00 23                   INX     H
0D01 73                   MOV     M,E
0D02 23                   INX     H
0D03 1A                   LDAX    D        ;PICK UP INSTRUCTION
0D04 77                   MOV     M,A      ;SAVE IT
0D05 3ECF                 MVI     A,(RST 1) ;REPLACE IT WITH A
0D07 12                   STAX    D        ;RESTART INSTRUCTION
0D08 3EC3                 MVI     A,0C3H   ;SET UP LO MEMORY
0D0A 320800               STA     8        ;WITH A JUMP TO BRKP
0D0D 212E0D               LXI     H,BRKP
0D10 220900               SHLD    9
0D13 C9                   RET              ;THEN RETURN
                  ;
                  ; THIS ROUTINE CLEARS ALL BREAKPOINTS
                  ;
```

```
0D14 210C10     CLRB:   LXI     H,BRT    ;GET TABLE ADDRESS
0D17 0608               MVI     B,NBR    ;GET NUMBER OF BREAKPOINTS
0D19 AF         CLBL:   XRA     A        ;GET A ZERO
0D1A 56                 MOV     D,M      ;GET HI-BYTE OF ENTRY
0D1B 77                 MOV     M,A
0D1C 23                 INX     H
0D1D 5E                 MOV     E,M      ;GET LO-BYTE OF ENTRY
0D1E 77                 MOV     M,A
0D1F 23                 INX     H
0D20 46                 MOV     B,M      ;GET INST BYTE
0D21 23                 INX     H
0D22 7A                 MOV     A,D      ;WAS THIS A NULL ENTRY
0D23 B3                 ORA     E
0D24 CA290D             JZ      CL2      ;BRANCH IF IT WAS
0D27 78                 MOV     A,B
0D28 12                 STAX    D        ;ELSE, PLUG INST BACK IN
0D29 05         CL2:    DCR     B        ;BUMP COUNT
0D2A C2190D             JNZ     CLBL     ;GO DO NEXT ONE
0D2D C9                 RET              ;RETURN WHEN DONE
                ;
                ; COME HERE WHEN WE HIT A BREAKPOINT
                ;
0D2E 220810     BRKP:   SHLD    HOLD+8   ;SAVE H,L
0D31 E1                 POP     H        ;GET PC
0D32 2B                 DCX     H        ;ADJUST IT
0D33 220A10             SHLD    HOLD+10  ;SAVE IT
0D36 F5                 PUSH    PSW      ;SAVE FLAGS
0D37 E1                 POP     H        ;GET THEM INTO HL
0D38 220010             SHLD    HOLD     ;NOW STORE THEM FOR USER
0D3B 210000             LXI     H,0
0D3E 39                 DAD     SP       ;GET STACK POINTER
0D3F 310810             LXI     SP,HOLD+8 ;SET NEW SP
0D42 E5                 PUSH    H        ;SAVE OLD SP
0D43 D5                 PUSH    D        ;SAVE D,E
0D44 C5                 PUSH    B        ;SAVE B,C
0D45 2F                 CMA              ;COMPLEMENT ACC
0D46 D3FF               OUT     0FFH     ;DISPLAY IT IN THE LIGHTS
0D48 31B210             LXI     SP,AREA+18 ;SET SP AGAIN
0D4B 2A0A10             LHLD    HOLD+10  ;GET PC
0D4E EB                 XCHG             ;INTO D,E
0D4F 210C10             LXI     H,BRT    ;GET ADDR OF TABLE
0D52 0608               MVI     B,NBR    ;AND NUMBER OF ENTRIES
0D54 7E         BL1:    MOV     A,M      ;GET AN ENTRY FROM THE TABLE
0D55 23                 INX     H
0D56 BA                 CMP     D        ;DOES IT MATCH
0D57 C25F0D             JNZ     BL2      ;BRANCH IF NOT
0D5A 7E                 MOV     A,M      ;ELSE GET NEXT BYTE
0D5B BB                 CMP     E        ;CHECK IT
0D5C CA680D             JZ      BL3      ;IT MATCHES!
0D5F 23         BL2:    INX     H        ;BUMP AROUND THIS ENTRY
0D60 23                 INX     H
0D61 05                 DCR     B        ;BUMP COUNT
0D62 CA5A04             JZ      WHAT     ;NOT IN OUR TABLE!
0D65 C3540D             JMP     BL1
                ;
0D68 23         BL3:    INX     H
0D69 7E                 MOV     A,M      ;GET INSTR BYTE
0D6A 12                 STAX    D        ;PUT IT BACK
0D6B AF                 XRA     A        ;CLEAR ENTRY IN TABLE
0D6C 2B                 DCX     H
0D6D 77                 MOV     M,A
0D6E 2B                 DCX     H
0D6F 77                 MOV     M,A
0D70 CD0E01             CALL    CRLF     ;RESTORE THE CARRIAGE
0D73 3A0B10             LDA     HOLD+11  ;GET HI-BYTE OF PC
0D76 CD3A02             CALL    HOUT     ;TYPE IT
```

```
0D79 3A0A10             LDA     HOLD+10 ;GET LO-BYTE OF PC
0D7C CD3A02             CALL    HOUT    ;TYEP IT
0D7F 21880D             LXI     H,BMES  ;TELL USER WHAT IT IS
0D82 CD7A02             CALL    SCRN
0D85 C36700             JMP     EOR     ;GO BACK TO COMMAND LEVEL
                ;
0D88 20425245   BMES:   DB      ' BREAK',13
0D8C 41480D
                ;
                ;
                ; THIS ROUTINE PROCEEDS FROM A BREAKPOINT
                ;
0D8F 3A7E10     PROC:   LDA     ABUF    ;CHECK FOR ARG
0D92 B7                 ORA     A
0D93 CA9C0D             JZ      P1      ;JMP IF NO ARG
0D96 2A8A10             LHLD    BBUF    ;ELSE, GET ARG
0D99 220A10             SHLD    HOLD+10 ;PLUG IT INTO PC SLOT
0D9C 310010     P1:     LXI     SP,HOLD ;SET SP TO POINT AT REG'S
0D9F F1                 POP     PSW     ;RESTORE PSW
0DA0 C1                 POP     B       ;RESTORE B,C
0DA1 D1                 POP     D       ;RESTORE D,E
0DA2 E1                 POP     H       ;GET OLD SP
0DA3 F9                 SPHL            ;RESTORE IT
0DA4 2A0A10             LHLD    HOLD+10 ;GET PC
0DA7 E5                 PUSH    H       ;PUT IT ON STACK
0DA8 2A0810             LHLD    HOLD+8  ;RESTORE H,L
0DAB C9                 RET             ;AND PROCEED
                ;
                ; SYSTEM RAM
                ;
0DAC                    ORG     1000H
                ;
                ; DEFINE BREAKPOINT REGION
                ;
0008            NBR     EQU     8       ;NUMBER OF BREAKPOINTS
1000            HOLD:   DS      12      ;REGISTER HOLD AREA
100C            BRT:    DS      3*NBR   ;BREAKPOINT TABLE
                ;
                ; FILE AREA PARAMETERS
0006            MAXFIL  EQU     6       ;MAX # OF FILES
0005            NMLEN   EQU     5       ;NAME LENGTH
000D            FELEN   EQU     NMLEN+8 ;DIRECTORY ENTRY LENGTH
1024            FILE0:  DS      NMLEN
1029            BOFP:   DS      2
102B            EOFP:   DS      2
102D            MAXL:   DS      4
1031            FILTB:  DS      (MAXFIL-1)*FELEN
1072            INSP:   DS      2       ;INSERT LINE POSITION
1072            DELP    EQU     INSP    ;DELETE LINE POSITION
000D            ASCR    EQU     13      ;ASCII CARRIAGE RETURN VALUE
1074            HCON:   DS      2
1074            ADDS    EQU     HCON    ;FIND ADDRESS
1076            FBUF:   DS      NMLEN   ;FILE NAME BUFFER
107B            FREAD:  DS      2       ;FREE ADDRESS IN DIRECTORY
107D            FEF:    DS      1       ;FREE ENTRY FOUND FLAG
107D            FOCNT   EQU     FEF     ;OUTPUT COUNTER
107E            ABUF:   DS      12      ;ASCII BUFFER
108A            BBUF:   DS      4       ;BINARY BUFFER
108E            SCNT:   DS      1
108F            DCNT:   DS      1       ;DUMP ROUTINE COUNTER
000B            NCOM    EQU     11      ;NUMBER OF COMMANDS
1090            TABA:   DS      2       ;SYMBOL TABLE END ADDRESS
1092            ASPC:   DS      2       ;ASSEMBLER PROGRAM COUNTER
1094            PASI:   DS      1       ;PASS INDICATOR
1095            NCHR:   DS      1       ;LENGTH OF STRING FOR COMPARE
1096            PNTR:   DS      2       ;LINE POINTER STORAGE
```

```
1098        NOLA:   DS      1       ;NUMBER OF LABELS
1099        SIGN:   DS      1       ;SIGN STORAGE FOR SCAN
109A        OPRD:   DS      2       ;OPERAND STORAGE
109C        OPRI:   DS      1       ;OPERAND FOUND INDICATOR
109D        TEMP:   DS      1
1072        APNT    EQU     INSP    ;ASSEMBLE LINE POINTER
108E        AERR    EQU     SCNT    ;ASSEMBLER ERROR PRINT SWITCH
109E        OIND:   DS      2       ;OUTPUT ADDRESS
0005        LLAB    EQU     5       ;LENGTH OF LABELS
10A0        AREA:   DS      18
10B2        OBUF:   DS      16      ;OUTPUT BUFFER AREA
10C2                DS      5
10C7        IBUF:   DS      83
111A        SYMT    EQU     $       ;START OF SYMBOL TABLE
            ;
            ; TELETYPE PARAMETERS
            ;
0003        TTS     EQU     3       ;TTY STATUS PORT
0002        TTI     EQU     2       ;TTY DATA IN PORT
0002        TTO     EQU     2       ;TTY DATA OUT PORT
0002        TTYDA   EQU     2       ;TTY DATA AVAILABLE BIT
0001        TTYTR   EQU     1       ;TTY XMTR READY BIT
00FF        SWCH    EQU     0FFH    ;SWITCH REGISTER
            ;
0000                END
```

BOOTSTRAP LOADER

The IMSAI Bootstrap Loader is a system that allows the user to get a general paper tape loader into any region of RAM using only a 32-byte key-in. It requires an ASR33 teletype. To use this loader, proceed as follows:

1. Key in the basic bootstrap given below starting at location 0000.

   3E CE D3 03 3E 17 D3 03 21 20 00 06 F8 DB 03 E6
   02 CA 0D 00 DB 02 77 3C CA 08 00 23 05 C2 0D 00

2. Mount the bootstrap tape in the paper tape reader on the teletype so that the block of rubouts (frames with all the holes punched out) is in the reader.

3. Set the PROGRAMMED INPUT switches to the high order 8 bits of the address where the paper tape loader is to be located, e.g., to put the loader at 5C00 hex, set the PROGRAMMED INPUT switches to 5C hex. (See the warning below.)

4. Press STOP, RESET and RUN, then manually start the paper tape reader on the teletype.

If all goes well, the tape should go through the reader, stop at the end, then the loader will print an "*" on the teletype. If this is the case, refer to the IMSAI Paper Tape Loader section to use the loader.

If the loader does not type an asterisk after the tape has gone through the reader, this means the loader was not read in correctly. Proceed as follows:

1. Check the basic bootstrap key into it as correct.

2. If the key-in is correct, check the bootstrap tape for tears or distorted holes. (These may usually be fixed with cellophane tape.)

If the key-in and bootstrap tape are correct, the problem may be dirty contacts in the teletype reader. Try repeating the bootstrap procedure from the beginning.

WARNING:

1. Since the bootstrap loader resides in location 20 hex - 120, do not try to load the paper tape loader below 200 hex or it will overlay the bootstrap.

2. Be sure to locate the loader in a region where it will not be overlayed by the program it is loading. For instance, 8K BASIC occupies locations 0000-1FFF hex, so that to load 8K BASIC, the loader should be located at or above 2000 hex.

BOOTSTRAP LOADER PROGRAM LOGIC

The Bootstrap Loader is a system that allows the user to read the Paper Tape Loader into the region of RAM that begins on a 256-word boundary using a specially formatted tape.

1. Bootstrap Tape Format:

   The Bootstrap Tape consists of two sections. The first section consists of a direct core image of the second level bootstrap (described below), preceded by a block of rubouts. In this section of the tape, each frame corresponds directly to one data byte. The second section consists of the Paper Tape Loader in standard object format.

2. Overall Logic:

   The Bootstrap Sequence Procedure is as follows:

   a. The user keys in a simple 32-byte bootstrap, starts it up, then starts the tape reader on the teletype.

   b. The basic bootstrap reads in the second level bootstrap from the first part of the bootstrap tape and starts it up.

   c. The second level bootstrap stops the tape reader then checksums itself to make sure it was loaded correctly. If not, it hangs up.

   d. If the second level bootstrap checksums correctly, it starts the tape reader and reads in the paper tape loader from the second part of the bootstrap tape and locates it in the 256-byte page specified by the PROGRAMMED INPUT switches. If it detects an error in the tape, it stops the reader and hangs up.

   e. When the Paper Tape Loader is completely loaded, it stops the paper tape reader, then starts up the Paper Tape Loader.

3. Basic Bootstrap:

   The Basic Key-In Bootstrap was designed to be as short as possible. It merely reads in characters from the tape and stores them directly into memory. Whenever it reads in a byte of FF hex, it resets its pointer and counter. This allows it to use the block of rubouts at the beginning of the tape to synchronize on.

4. Second Level Bootstrap:

   The second level bootstrap is a modified version of the Paper Tape Loader. The main differences between the two are:

   a. The second level bootstrap checksums itself to make sure it was loaded properly. This is done because the Basic key-in bootstrap, for reasons of brevity, does not error checking.

   b. If it encounters an error, the second level bootstrap turns off the tape and hangs up.

   c. If it encounters a byte of FD hex, it substitutes the contents of the PROGRAMMED INPUT switches. This is done so that the Paper Tape Loader may be located at any 256-byte page in memory. See below.

5. Relocating the Paper Tape Loader

   The Paper Tape Loader that is on the second part of the bootstrap tape was assembled to begin at FD00 hex. Since there is no instruction with op-code FD hex, the only times a byte of FD hex will appear on the tape are:

   a. The high byte of the address field in the paper tape record. (Note that the high byte of the address fields of all records will be FD hex.)

   b. The high byte of the address in a jump instruction.

   Therefore, by substituting another value (in this case, the contents of the PROGRAMMED INPUT switches) for every occurance of FD hex, we can load the Paper Tape Loader into any 256-byte page in memory.

PAPER TAPE LOADER

The IMSAI Paper Tape Loader is a program that will load tapes in the standard object format (see appendix) from the paper tape reader on an ASR33 teletype.

If the paper tape loader is read in with the bootstrap loader (see Bootstrap Loader section), it will start itself up and print an "*" on the teletype. Otherwise, it should be manually started at its beginning address.

When the loader prints an "*" on the teletype, mount the tape to be loaded in the paper tape reader on the teletype. Then, strike any key on the teletype. The paper tape reader should start automatically. While the tape is being read in, the data being loaded will be displayed in the PROGRAMMED OUTPUT lights.

The loader will stop the reader and print an "*" under two conditions:

1. If the PROGRAMMED OUTPUT displays 00 (all lights off), the loader has encountered an End-of-File record, an the program has been successfully loaded. At this point, another tape may be loaded by placing it in the paper tape reader and striking a key on the teletype.

2. If something other than 00 is displayed in the PROGRAMMED OUTPUT lights, a bad record has been encountered in the tape. The record may be re-read as follows:

   o Move the switch on the reader to the "FREE" position

   o Back the tape up about two feet

   o Put the switch back in the "STOP" position

   o Strike a key on the teletype

   If the loader stops again on the same record, inspect the tape for tears or distorted holes (these may usually be fixed with cellophane tape).

PAPER TAPE LOADER PROGRAM LOGIC

The IMSAI Paper Tape Loader is a program designed to load paper tapes in the standard object format from the paper tape reader on an ASR33 teletype. The loader is designed to use no stack or local RAM, thereby allowing it to be executed out of ROM.

1. Object Tape Format:

The standard object format is a blocked hexadecimal format. The data on the tape is blocked into discrete records, each record containing record length, record type, memory address and checksum information in addition to data. A frame-by-frame description is as follows:

Frame 0	Record Mark. Signals the start of a record. The ASCII character colon (":" 3A hex) is used as the record mark.
Frames 1,2 (0-9, A-F)	Record Length. Two ASCII characters representing a hexadecimal number in the range 0 to FF (0 to 255). This is the count of actual data bytes in the record type or checksum. A record length of 0 indicates end-of-file.
Frames 3 to 6	Load Address. Four ASCII characters that represent the initial memory location where the data following will be loaded. The first data byte is stored in the location pointed to by the load address; succeeding data bytes are loaded into ascending addresses.
Frames 7,8	Record Type. Two ASCII characters. Currently all records are type 0. This field is reserved for future expansion.
Frames 9 to 9+2*	Data. Each 8-bit memory word is represented by two frames containing the ASCII characters 0-9, A-F) to represent a hexadecimal value 0 to FF hex (0 to 255).
Frames 9+2* (Record Length) to 9+2* (Record Length + 1	Checksum. The checksum is the negative of the sum of all 8-bit bytes in the record since the record mark (":") evaluated modulus 256. That is, if you add together all the 8-bit bytes, ignoring all carries out of an 8-bit sum then add the checksum, the result is zero.

Example: If memory locations 1 through 3 contain 53F8EC, the format of the hex file produced when these locations are punched is:

:0300010053F8ECC5

2. Register Allocation:

Since this loader uses no RAM, all variables and data are kept in the registers. The registers are assigned as follows:

A - scratch
B - byte count for data field
C - checksum
D - holes the data byte
E - flag register, describes what to do next

If this register contains zero, this program is looking for a ":" to signal the beginning of a block. Otherwise, if bit 7=1, then the next character is the first digit of a byte. If bit 2=0, the next character is the second digit of a byte. Bits 0-6 have the following significance:

1 - next byte is a count
2 - next byte is a high byte of the load address
3 - next byte is a low byte of the load address
4 - next byte is a type byte
5 - next byte is a data byte
6 - next byte is a checksum byte.

H, L - Load Address.

3. Logic:

The program flow is controlled by the flags in the E-register as given above.

```
                        ;
                        ; *** BASIC KEY-IN BOOTSTRAP LOADER ***
                        ;
                        ; THIS SIMPLE LOADER BOOTSTRAPS IN THE SECOND
                        ; LEVEL BOOTSTRAP, WHICH IN TURN LOADS THE
                        ; REAL PAPER TAPE LOADER.
                        ;
                        ; TO USE THIS LOADER, PROCEED AS FOLLOWS:
                        ;       (1) KEY IN THIS LOADER, STARTING AT LOC 1000
                        ;       (2) MOUNT THE BOOTSTRAP TAPE, SO THAT
                        ;           THE BLOCK OF RUBOUTS AT THE BEGINNING
                        ;           OF THE TAPE IS IN THE READER
                        ;       (3) SET THE PROGRAMMED INPUT SWITCHES TO THE
                        ;           HIGH ORDER 8 BITS OF THE ADDRESS WHERE
                        ;           YOU WANT THE PAPER TAPE LOADER TO
                        ;           BE LOADED. (E.G. TO CAUSE THE LOADER
                        ;           TO BE LOADED AT 5C00, SET THE PROGRAMMED
                        ;           INPUT SWITCHES TO 5C.)
                        ;       (4) PRESS THE 'RESET' KEY, FOLLOWED BY THE
                        ;           'RUN' KEY, THEN MANUALLY START THE PAPER
                        ;           TAPE READER ON THE TELETYPE.
                        ;
                        ;  IF EVERYTHING GOES CORRECTLY, THE LOADER WILL STOP
                        ;  THE PAPER TAPE READER, AND PRINT A * ON THE
                        ;  TELETYPE.  AT THIS POINT, MOUNT THE TAPE TO BE
                        ;  LOADED IN THE TELETYPE READER, THEN STRIKE ANY KEY
                        ;  ON THE TELETYPE.  THE LOADER WILL START THE
                        ;  PAPER TAPE READER, AND START LOADING THE TAPE. IF
                        ;  IT FINDS ANYTHING WRONG WITH THE TAPE, IT WILL
                        ;  STOP THE READER.  LOADING MAY BE CONTINUED BY
                        ;  STRIKING A KEY ON THE TELETYPE.
                        ;
                        ;
00F8                    CNT     EQU     0F8H    ;SIZE OF 2ND LEVEL BOOTSTRAP
                        ;
0000 3ECE               BOOT1:  MVI     A,0CEH  ;GET MODE BYTE FOR SIO BOARD.
0002 D303                       OUT     03      ;ISSUE IT
0004 3E17                       MVI     A,17H   ;GET COMMAND BYTE
0006 D303                       OUT     03      ; ISSUE IT
0008 212000             B1RST:  LXI     H,B1END ;GET LOAD ADDRESS
000B 06F8                       MVI     B,CNT   ;GET # OF BYTES
                        ;
000D DB03               LOOP:   IN      03      ;GET STATUS
000F E602                       ANI     2       ;IS THERE A BYTE READY
0011 CA0D00                     JZ      LOOP    ;KEEP WAITING
0014 DB02                       IN      2       ;GET THE BYTE
0016 77                         MOV     M,A     ;STORE IT
0017 3C                         INR     A       ;WAS IT A RUBOUT?
0018 CA0800                     JZ      B1RST   ;IF YES, RESET POINTERS
001B 23                         INX     H       ;ELSE, BUMP POINTER
001C 05                         DCR     B       ;AND DECR COUNT
001D C20D00                     JNZ     LOOP    ;IF NOT DONE, GO GET ANOTHER
                                                ; CHAR. ELSE, FALL THROUGH AND
                                                ; START UP SECOND LEVEL
                                                ; BOOTSTRAP.
0020                    B1END   EQU     $
0000                            END
```

```
                ;
                ; SECOND LEVEL BOOTSTRAP
                ;
                ; THIS LOADER IS PULLED IN BY THE BASIC KEY-IN
                ; LOADER. WHEN STARTED UP BY THE KEY-IN LOADER,
                ; IT CHECKSUMS ITSELF, TO MAKE SURE THAT IT
                ; HAS BEEN LOADED CORRECTLY, THEN PULLS IN AND
                ; RELOCATES THE MAIN PAPERTAPE LOADER.
                ;
                ; NOTE THAT THIS LOADER IS A SLIGHTLY MODIFIED
                ; VERSION OF THE MAIN PAPER TAPE LOADER.
                ;
                ;
0000                    ORG     20H
                ;
0020 3E13       BOOT2:  MVI     A,13H   ;GET STOP CHAR
0022 D302               OUT     2       ;STOP THE READER
0024 06F7               MVI     B,CHKSM-BOOT2 ;GET SIZE OF LDR
0026 212000             LXI     H,BOOT2 ;GET ADDRESS OF LDR
0029 AF                 XRA     A       ;CLEAR A AND CARRY
                ;
                ; PERFORM AN END-AROUND CHECKSUM, TO MAKE SURE
                ; WE WERE LOADED CORRECTLY
                ;
002A 8E         CHECK:  ADC     M       ;ADD IN A BYTE WITH CARRY
002B 23                 INX     H       ;BUMP POINTER
002C 05                 DCR     B       ;DECREMENT COUNT
002D C22A00             JNZ     CHECK   ;KEEP GOING
0030 CE00               ACI     0       ;ADD IN LAST CARRY
0032 BE                 CMP     M       ;COMPARE WITH CHECKSUM
0033 C23300     XXX:    JNZ     XXX     ;HANG UP IF NO GOOD.
                ;
                ; WE DO THE FOLLOWING NONSENSE BECAUSE THE
                ; BASIC KEY-IN BOOTSTRAP WILL NOT LOAD
                ; AN 0FFH CHARACTER.
                ;
0036 21BC00             LXI     H,FF1+1 ;GET ADDRESS OF 'IN 0FEH' INST
0039 34                 INR     M       ;MAKE IT 'IN 0FFH'.
003A 21B100             LXI     H,FF2+1 ;DO IT AGAIN
003D 34                 INR     M
003E 210B01             LXI     H,FF3+1 ;AND AGAIN
0041 34                 INR     M
                ;
                ; NOW WE'RE READY TO LOAD AND RELOCATE THE LOADER
                ;
0042 C35E00             JMP     STR     ;1ST TIME, SKIP RE-INIT STUFF.
                ;
0045 3EAA       START:  MVI     A,0AAH  ;GET DUMMY MODE BYTE
0047 D303               OUT     3
0049 3E40               MVI     A,40H   ;GET RESET COMMAND
004B D303               OUT     3       ;ISSUE IT
004D 3EFA               MVI     A,0FAH
004F D303               OUT     3       ;ISSUE MODE BYTE TO SIO
0051 3E17               MVI     A,17H
0053 D303               OUT     3       ;ISSUE COMMAND BYTE
0055 DB03       SL:     IN      03      ;GET STATUS
0057 E602               ANI     02      ;CHECK FOR CHAR READY
0059 CA5500             JZ      SL      ;KEEP WAITING
005C DB02               IN      02      ;READ CHAR AND IGNOR
005E DB03       STR:    IN      03      ;GET STATUS
0060 E601               ANI     1       ;MAKE SURE WE HAVE XMTR RDY
0062 CA5E00             JZ      STR
0065 3E11               MVI     A,11H   ;GET 'XON' CHAR
0067 D302               OUT     02      ;START READER
```

```
                        ;
0069 1E00       LOOP1:  MVI     E,0         ;CLEAR FLAG
006B 0E00               MVI     C,0         ;CLEAR CHECKSUM
                        ;
006D DB03       LOOP2:  IN      3           ;GET SIO STATUS
006F E602               ANI     2           ;CHECK FOR CHARACTER
0071 CA6D00             JZ      LOOP2       ;KEEP WAITING
0074 7B                 MOV     A,E         ;GET FLAG
0075 B7                 ORA     A           ;IS IT ZERO?
0076 C28700             JNZ     X1          ;NO, GO PROCESS A HEX CHAR
0079 DB02               IN      2           ;YES, WE'RE LOOKING FOR A COLON
007B E67F               ANI     127         ;STRIP OFF PARITY BIT
007D FE3A               CPI     ':'         ;IS IT A COLON?
007F C26D00             JNZ     LOOP2       ;NO, KEEP WAITING
0082 1E81               MVI     E,81H       ;YES, SET FLAG FOR COUNT BYTE
0084 C36D00             JMP     LOOP2       ;AND GET ANOTHER CHAR.
                        ;
                        ; WE'RE PUTTING TOGETHER A BYTE. FLAG BIT 7 = 1 => HIGH
                        ; DIGIT OF BYTE, BIT 7=0 => LOW DIGIT
                        ;
0087 F2A200     X1:     JP      Y1          ;JUMP IF LOW DIGIT
008A E67F               ANI     127         ;ELSE STRIP OFF HIGH BIT
008C 5F                 MOV     E,A         ;PUT FLAG BACK IN E-REG
008D DB02               IN      2           ;GET THE CHAR
008F E67F               ANI     127         ;STRIP OFF THE PARITY BIT
0091 FE3A               CPI     '9'+1       ;IS IT .LE. '9'
0093 FA9800             JM      X2          ;SKIP IT YES
0096 C609               ADI     9           ;IF NOT, ADJUST IT
0098 E60F       X2:     ANI     0FH         ;GET HEX DIGIT
009A 87                 ADD     A           ;SHIFT LEFT ONE BIT
009B 87                 ADD     A           ;     TWO BITS
009C 87                 ADD     A           ;     THREE BITS
009D 87                 ADD     A           ;AND FOUR BITS.
009E 57                 MOV     D,A         ;SAVE NIBBLE IN D REG
009F C36D00             JMP     LOOP2
                        ;
                        ; PROCESS LOW DIGIT OF BYTE, THEN DECIDE WHAT TO DO WITH
                        ;
00A2 DB02       Y1:     IN      2           ;GET THE CHAR
00A4 E67F               ANI     127         ;GET RID OF PARITY BIT
00A6 FE3A               CPI     '9'+1       ;HEX IS SUCH A PAIN.
00A8 FAAD00             JM      Y2
00AB C609               ADI     9
00AD E60F       Y2:     ANI     0FH
00AF B2                 ORA     D           ;MAKE THE BYTE
00B0 D3FE       FF2:    OUT     0FEH        ;PUT IT IN LIGHTS
00B2 57                 MOV     D,A         ;SAVE IT IN D REG
00B3 81                 ADD     C           ;ADD IT INTO CHECKSUM
00B4 4F                 MOV     C,A         ;SAVE RUNNING CHECKSUM
00B5 7A                 MOV     A,D         ;GET BYTE BACK
00B6 FEFD               CPI     0FDH        ;IS IT FELOCATABLE BYTE?
00B8 C2BD00             JNZ     Y3          ;BRANCH IF NOT
00BB DBFE       FF1:    IN      0FEH        ;ELSE SUBSTITUE SWITCHS
00BD 57         Y3:     MOV     D,A         ;PUT BYTE BACK IN D
00BE 7B                 MOV     A,E         ;GET FLAG IN A
00BF 3D                 DCR     A           ;THEN DISPATCH  ON IT
00C0 CA0401             JZ      COUNT
00C3 3D                 DCR     A
00C4 CAFE00             JZ      HADD
00C7 3D                 DCR     A
00C8 CAF800             JZ      LADD
00CB 3D                 DCR     A
00CC CAF300             JZ      TYPE
00CF 3D                 DCR     A
00D0 CAE700             JZ      PUT
00D3 79                 MOV     A,C         ;MUST BE TIME TO CHECK THE
```

```
00D4 B7                 ORA     A       ; CHECKSUM. IS IT ZERO?
00D5 CA6900             JZ      LOOP1   ;YES, GO GET NEXT RECORD
00D8 214500             LXI     H,START ;ELSE, GET RESTART ADDR
00DB 3E13       STOP:   MVI     A,13H   ;GET 'XOFF' CHAR
00DD D302               OUT     2       ;TURN OFF READER
00DF DB03       STPL:   IN      3       ;WAIT TILL XMTR BUFFER EMPTY
00E1 E604               ANI     4
00E3 CADF00             JZ      STPL
00E6 E9                 PCHL            ;GO AWAY.
                ;
                ;
                ; PUT A DATA BYTE INTO CORE
                ;
00E7 72         PUT:    MOV     M,D     ;STORE THE DATA
00E8 23                 INX     H       ;INCREMENT THE H REG
00E9 1E85               MVI     E,85H   ;RESET FLAG FOR NEXT DATA BYTE
00EB 05                 DCR     B       ;DECR COUNT
00EC C26D00             JNZ     LOOP2   ;GO BACK FOR MORE DATA.
00EF 1C                 INR     E       ;OUT OF DATA, SET FLAG FOR
00F0 C36D00             JMP     LOOP2   ;  CHECKSUM.
                ;
                ; IGNORE A TYPE BYTE
                ;
00F3 1E85       TYPE:   MVI     E,85H   ;SET FLAG FOR DATA
00F5 C36D00             JMP     LOOP2   ;GO GET DATA
                ;
                ; GET LOW BYTE OF ADDRESS
                ;
00F8 6A         LADD:   MOV     L,D     ;GET BYTE INTO L-REG
00F9 1E84               MVI     E,84H   ;SET FLAG FOR TYPE BYTE
00FB C36D00             JMP     LOOP2
                ;
                ; GET HIGH BYTE OF ADDRESS
                ;
00FE 62         HADD:   MOV     H,D     ;GET BYTE INTO H
00FF 1E83               MVI     E,83H   ;SET FLAG FOR LOW ADDRESS BYTE
0101 C36D00             JMP     LOOP2
                ;
                ; GET COUNT BYTE
                ;
0104 42         COUNT:  MOV     B,D     ;PUT COUNT INTO B
0105 7A                 MOV     A,D     ;CHECK FOR EOF
0106 B7                 ORA     A
0107 C21201             JNZ     C1      ;IF NOT EOF, CONTINUE
010A DBFE       FF3:    IN      0FEH    ;GET HIGH BYTE OF LOADER
010C 67                 MOV     H,A     ; ADDRESS INTO H
010D 2E00               MVI     L,0     ;AND LOW BYTE
010F C3DB00             JMP     STOP    ;STOP TAPE, THEN GOTO LOADER.
                ;
0112 1E82       C1:     MVI     E,82H   ;SET FLAG FOR ADDRESS BYTE
0114 C36D00             JMP     LOOP2
                ;
                ;
0117 C8         CHKSM:  DB      0C8H    ;SELF-CHECKSUM FOR THIS LOADER
                ;
0000                    END
```

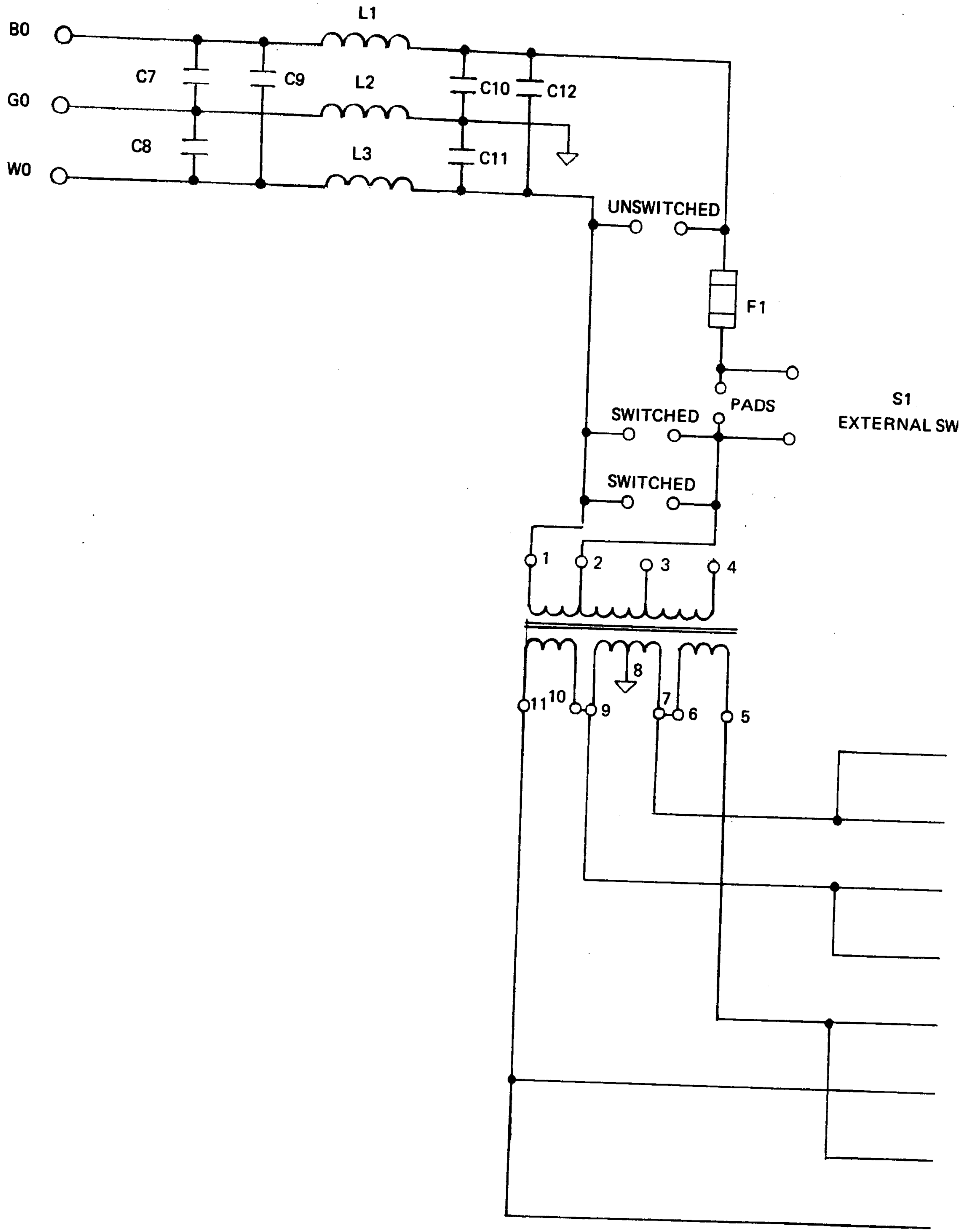
```
                ;
                ; *** IMSAI PAPER TAPE LOADER ***
                ;
                ;       REV 0   3/3/76
                ;
                ; THIS LOADER IS DESIGNED TO LOAD PAPER TAPES IN
                ; THE STANDARD OBJECT FORMAT (SEE THE SOFTWARE
                ; SECTION OF THE 8080 USER MANUAL) FROM AN ASR 33
                ; TELETYPE.  IT USES NO STACK AND NO LOCAL RAM, SO
                ; THAT IT MAY BE RUN FROM PROM WITHOUT REQUIRING
                ; A RAM CARD OF ITS OWN.
                ;
                ; USING THE LOADER:
                ;       IF THIS LOADER IS BROUGHT IN WITH THE
                ; BOOTSTRAP SEQUENCE (DOCUMENTED ELSEWHERE),
                ; IT WILL START ITSELF UP. OTHERWISE, MANUALLY
                ; START IT AT ITS BEGINNING.  IT WILL RESPOND
                ; BY TYPING A * ON THE TELETYPE. MOUNT THE TAPE
                ; TO BE LOADED IN THE READER, AND STRIKE ANY KEY.
                ; THE LOADER WILL START THE READER AUTOMATICALLY.
                ;       THE LOADER WILL STOP THE TAPE AND TYPE A * IN
                ; EITHER OF TWO CASES:
                ;
                ;       (1) IT HAS SEEN AN END OF FILE RECORD. IN
                ;           THIS CASE, ZERO WILL BE DISPLAYED IN
                ;           THE PROGRAMMED OUTPUT LIGHTS.
                ;
                ;       (2) IT ENCOUNTERED A BAD RECORD. IN THIS CASE
                ;           AN NON-ZERO QUANTITY WILL BE DISPLAYED
                ;           IN THE PROGRAMMED OUTPUT LIGHTS.
                ;
                ; IN EITHER CASE, LOADING MAY BE CONTINUED BY STRIKING
                ; A KEY.
                ;
                ;
0000                    ORG     0FD00H
                ;
FD00 110100     START:  LXI     D,1     ;WAIT ABOUT A SECOND SO A
FD03 210000             LXI     H,0     ;  PREVIOUS 'XOFF' CHARACTER
FD06 19         SL0:    DAD     D       ;  HAS TIME TO STOP THE READER
FD07 D206FD             JNC     SL0
                ;
                ; INITIALIZE SIO BOARD.
                ;
FD0A 3EAA               MVI     A,0AAH  ;GET DUMMY MODE BYTE
FD0C D303               OUT     3
FD0E 3E40               MVI     A,40H   ;GET RESET COMMAND
FD10 D303               OUT     3       ;ISSUE IT
FD12 3EFA               MVI     A,0FAH
FD14 D303               OUT     3       ;ISSUE MODE BYTE TO SIO
FD16 3E17               MVI     A,17H
FD18 D303               OUT     3       ;ISSUE COMMAND BYTE
FD1A 3E2A               MVI     A,'*'   ;GET AN ASTERISK
FD1C D302               OUT     02      ;PRINT IT
FD1E DB02               IN      02      ;THROW AWAY ANY CHAR IN BUFFER
FD20 DB03       SL2:    IN      03      ;GET STATUS
FD22 E602               ANI     02      ;CHECK FOR CHAR READY
FD24 CA20FD             JZ      SL2     ;KEEP WAITING
FD27 DB02               IN      02      ;READ CHAR AND IGNOR
FD29 3E11               MVI     A,11H   ;GET 'XON' CHAR
FD2B D302               OUT     02      ;START READER
                ;
FD2D 1E00       LOOP1:  MVI     E,0     ;CLEAR FLAG
FD2F 0E00               MVI     C,0     ;CLEAR CHECKSUM
```

```
                     ;
FD31 DB03     LOOP2:  IN      3         ;GET SIO STATUS
FD33 E602             ANI     2         ;CHECK FOR CHARACTER
FD35 CA31FD           JZ      LOOP2     ;KEEP WAITING
FD38 7B               MOV     A,E       ;GET FLAG
FD39 B7               ORA     A         ;IS IT ZERO?
FD3A C24BFD           JNZ     X1        ;NO, GO PROCESS A HEX CHAR
FD3D DB02             IN      2         ;YES, WE'RE LOOKING FOR A COLON
FD3F E67F             ANI     127       ;STRIP OFF PARITY BIT
FD41 FE3A             CPI     ':'       ;IS IT A COLON?
FD43 C231FD           JNZ     LOOP2     ;NO, KEEP WAITING
FD46 1E81             MVI     E,81H     ;YES, SET FLAG FOR COUNT BYTE
FD48 C331FD           JMP     LOOP2     ;AND GET ANOTHER CHAR.
                     ;
                     ; WE'RE PUTTING TOGETHER A BYTE. FLAG BIT 7 = 1 => HIGH
                     ; DIGIT OF BYTE, BIT 7=0 => LOW DIGIT
                     ;
FD4B F266FD   X1:     JP      Y1        ;JUMP IF LOW DIGIT
FD4E E67F             ANI     127       ;ELSE STRIP OFF HIGH BIT
FD50 5F               MOV     E,A       ;PUT FLAG BACK IN E-REG
FD51 DB02             IN      2         ;GET THE CHAR
FD53 E67F             ANI     127       ;STRIP OFF THE PARITY BIT
FD55 FE3A             CPI     '9'+1     ;IS IT .LE. '9'
FD57 FA5CFD           JM      X2        ;SKIP IT YES
FD5A C609             ADI     9         ;IF NOT, ADJUST IT
FD5C E60F     X2:     ANI     0FH       ;GET HEX DIGIT
FD5E 87               ADD     A         ;SHIFT LEFT ONE BIT
FD5F 87               ADD     A         ;    TWO BITS
FD60 87               ADD     A         ;    THREE BITS
FD61 87               ADD     A         ;AND FOUR BITS.
FD62 57               MOV     D,A       ;SAVE NIBBLE IN D REG
FD63 C331FD           JMP     LOOP2
                     ;
                     ; PROCESS LOW DIGIT OF BYTE, THEN DECIDE WHAT TO DO WITH
                     ;
FD66 DB02     Y1:     IN      2         ;GET THE CHAR
FD68 E67F             ANI     127       ;GET RID OF PARITY BIT
FD6A FE3A             CPI     '9'+1     ;HEX IS SUCH A PAIN.
FD6C FA71FD           JM      Y2
FD6F C609             ADI     9
FD71 E60F     Y2:     ANI     0FH
FD73 B2               ORA     D         ;MAKE THE BYTE
FD74 D3FF             OUT     0FFH      ;PUT IT IN LIGHTS
FD76 57               MOV     D,A       ;SAVE IT IN D REG
FD77 81               ADD     C         ;ADD IT INTO CHECKSUM
FD78 4F               MOV     C,A       ;SAVE RUNNING CHECKSUM
FD79 7B               MOV     A,E       ;GET FLAG IN A
FD7A 3D               DCR     A         ;THEN DISPATCH  ON IT
FD7B CAC1FD           JZ      COUNT
FD7E 3D               DCR     A
FD7F CABBFD           JZ      HADD
FD82 3D               DCR     A
FD83 CAB5FD           JZ      LADD
FD86 3D               DCR     A
FD87 CAB0FD           JZ      TYPE
FD8A 3D               DCR     A
FD8B CAA4FD           JZ      PUT
FD8E 79               MOV     A,C       ;MUST BE TIME TO CHECK THE
FD8F B7               ORA     A         ; CHECKSUM. IS IT ZERO?
FD90 CA2DFD           JZ      LOOP1     ;YES, GO GET NEXT RECORD
FD93 2F       STOP:   CMA               ;DISPLAY REASON FOR STOPPING
FD94 D3FF             OUT     0FFH
FD96 3E13             MVI     A,13H     ;ELSE, GET 'XOFF' CHAR
FD98 D302             OUT     2         ;TURN OFF READER
FD9A DB03     STPL:   IN      3         ;WAIT TILL XMTR BUFFER EMPTY
FD9C E604             ANI     4
```

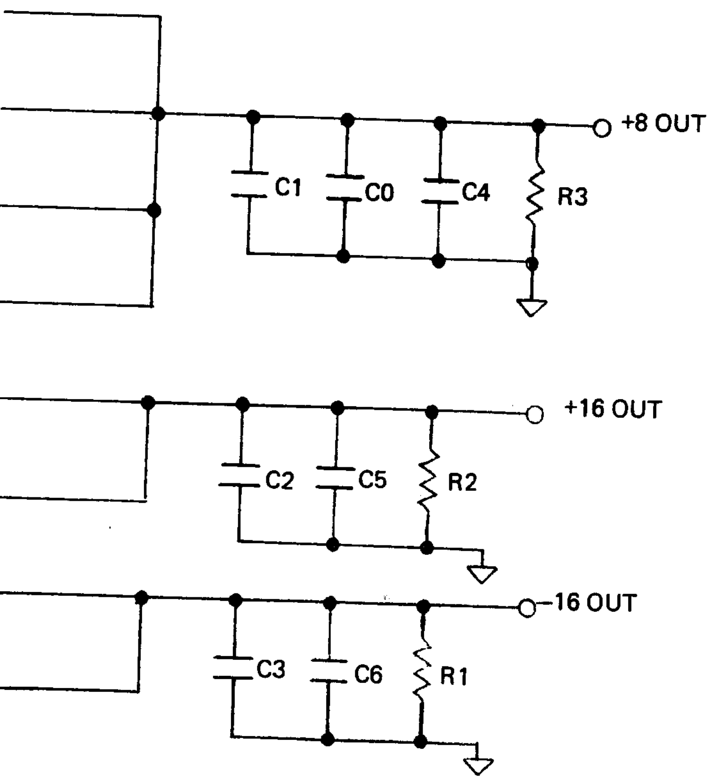
```
FD9E CA9AFD           JZ      STPL
FDA1 C300FD           JMP     START
                ;
                ;
                ; PUT A DATA BYTE INTO CORE
                ;
FDA4 72         PUT:  MOV     M,D       ;STORE THE DATA
FDA5 23               INX     H         ;INCREMENT THE H REG
FDA6 1E85             MVI     E,85H     ;RESET FLAG FOR NEXT DATA BYTE
FDA8 05               DCR     B         ;DECR COUNT
FDA9 C231FD           JNZ     LOOP2     ;GO BACK FOR MORE DATA.
FDAC 1C               INR     E         ;OUT OF DATA, SET FLAG FOR
FDAD C331FD           JMP     LOOP2     ;  CHECKSUM.
                ;
                ; IGNORE A TYPE BYTE
                ;
FDB0 1E85       TYPE: MVI     E,85H     ;SET FLAG FOR DATA
FDB2 C331FD           JMP     LOOP2     ;GO GET DATA
                ;
                ; GET LOW BYTE OF ADDRESS
                ;
FDB5 6A         LADD: MOV     L,D       ;GET BYTE INTO L-REG
FDB6 1E84             MVI     E,84H     ;SET FLAG FOR TYPE BYTE
FDB8 C331FD           JMP     LOOP2
                ;
                ; GET HIGH BYTE OF ADDRESS
                ;
FDBB 62         HADD: MOV     H,D       ;GET BYTE INTO H
FDBC 1E83             MVI     E,83H     ;SET FLAG FOR LOW ADDRESS BYTE
FDBE C331FD           JMP     LOOP2
                ;
                ; GET COUNT BYTE
                ;
FDC1 42         COUNT: MOV    B,D       ;PUT COUNT INTO B
FDC2 7A               MOV     A,D       ;CHECK FOR EOF
FDC3 B7               ORA     A
FDC4 CA93FD           JZ      STOP      ;IF EOF, GO STOP READER
FDC7 1E82             MVI     E,82H     ;ELSE SET FLAG FOR ADDRESS BYTE
FDC9 C331FD           JMP     LOOP2
                ;
                ;
0000                  END
```

APPENDIX — SCHEMATIC DIAGRAMS

POWER SUPPLY
CPA
MPU-A
RAM 4
PROM 4
PIO 4
SIO 2
PIC - 8
CRI


B0
G0
W0
L1
L2
L3
C7
C8
C9
C10
C11
C12
UNSWITCHED
F1
PADS
S1
EXTERNAL SW
SWITCHED
SWITCHED
1
2
3
4
8
11
10
9
7
6
5

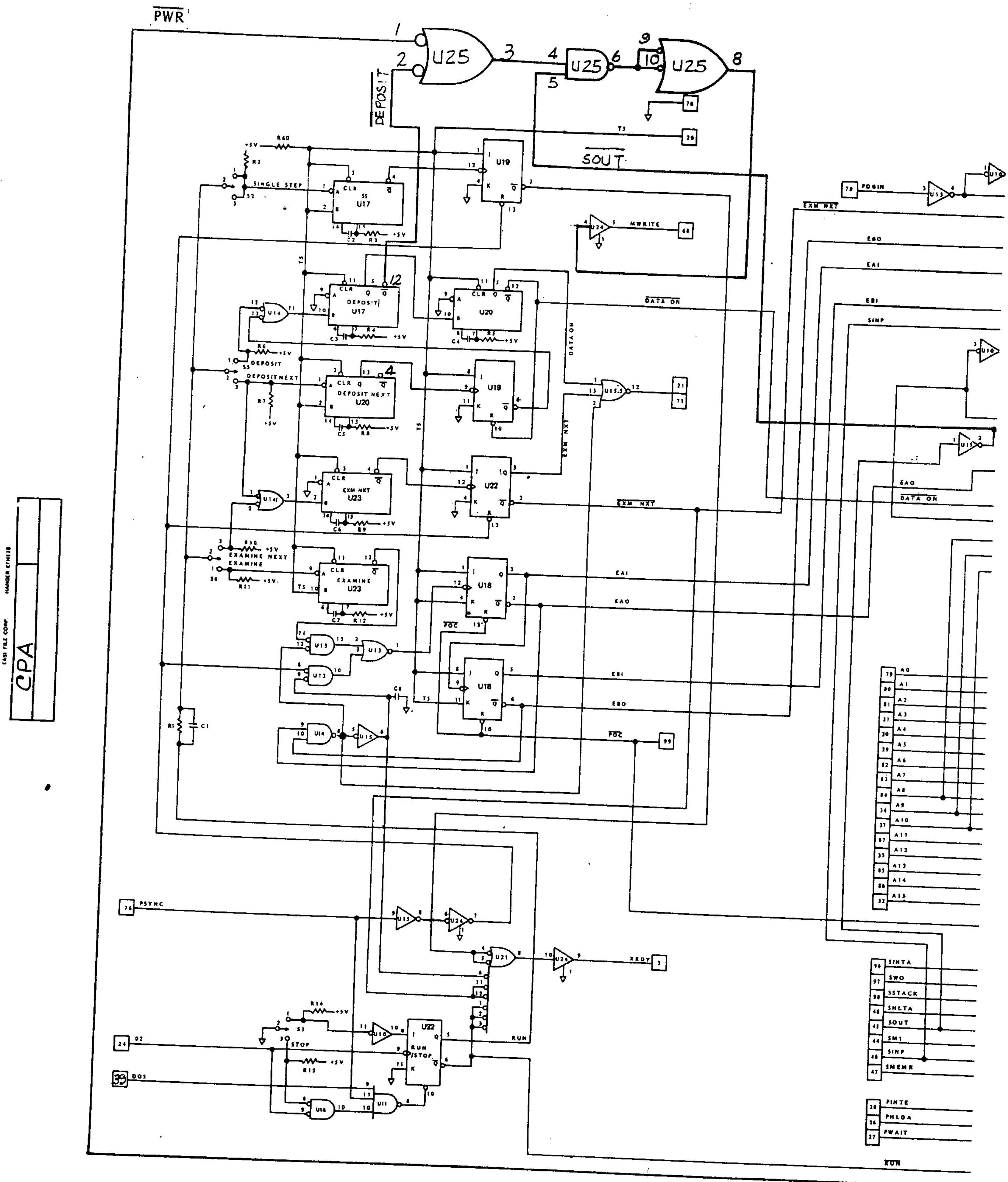
PSC-L



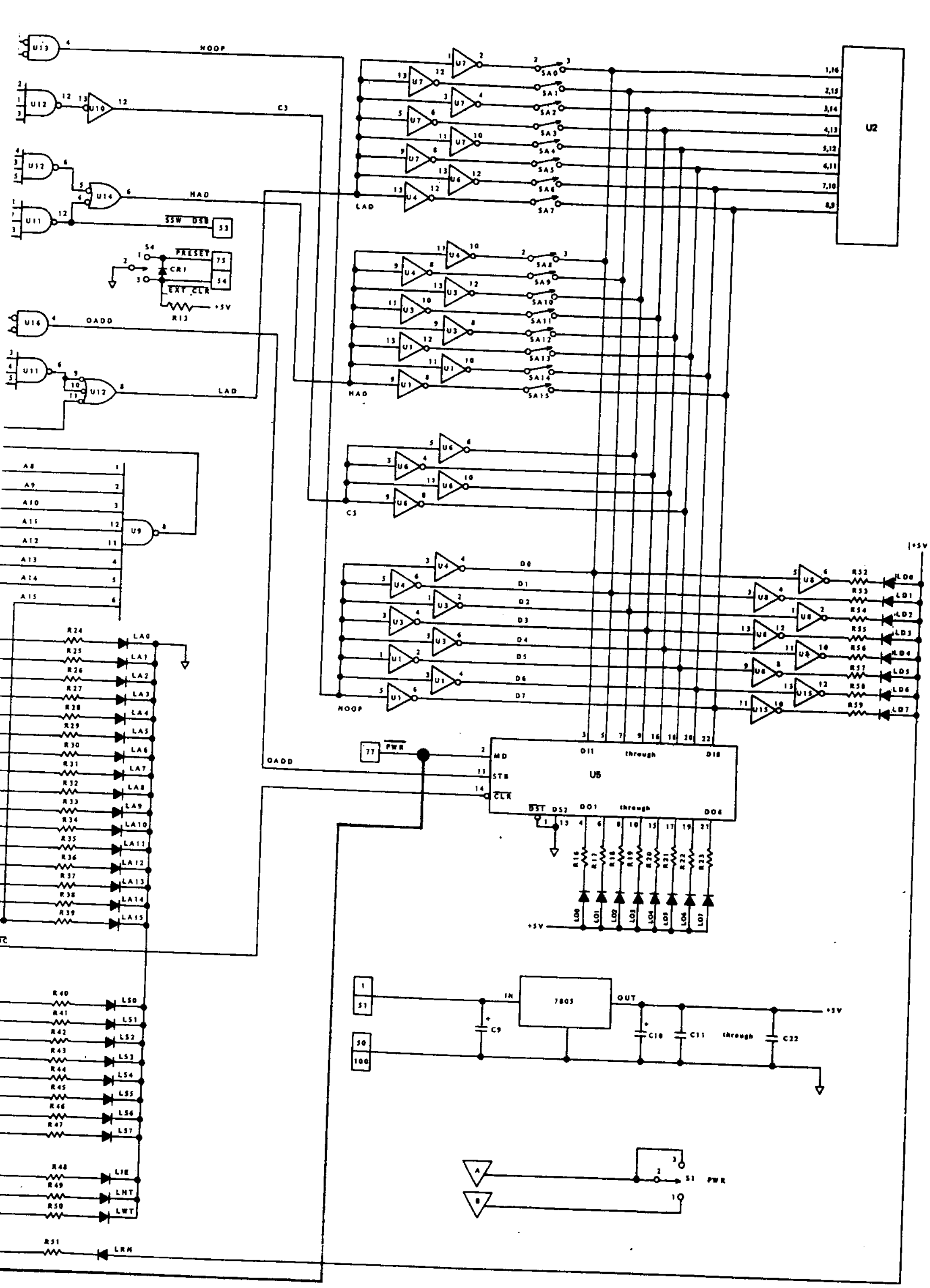
C0, C1	95K uF
C2, C3	10K uF
C4, C5, C6	.1 uF
C7 thru C12	.04 uF 500V
CR0 thru CR3	MR1121
CR4 thru CR7	MR501
F1	5A
L1 thru L3	8uH
R1, R2	1K ½W
R3	470 ½W
S1	NOT SUPPLIED

IMS ASSOCIATES INC.
SCHEMATIC DIAGRAM
PS·C REV. 1 5/76

PS-C-R


PWR
DEPOSIT
U25
SOUT
SINGLE STEP
DEPOSIT NEXT
EXAMINE NEXT
EXAMINE
EXM NXT
DATA ON
MWRITE
PDBIN
EBO
EAI
EBI
SINP
EAO
POC
PSYNC
XRDY
RUN
STOP
RUN /STOP
D2
DOS
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
SINTA
SWO
SSTACK
SHLTA
SOUT
SM1
SINP
SMEMR
PINTE
PHLDA
PWAIT
EASI FILE CORP.
CPA

CPA-L

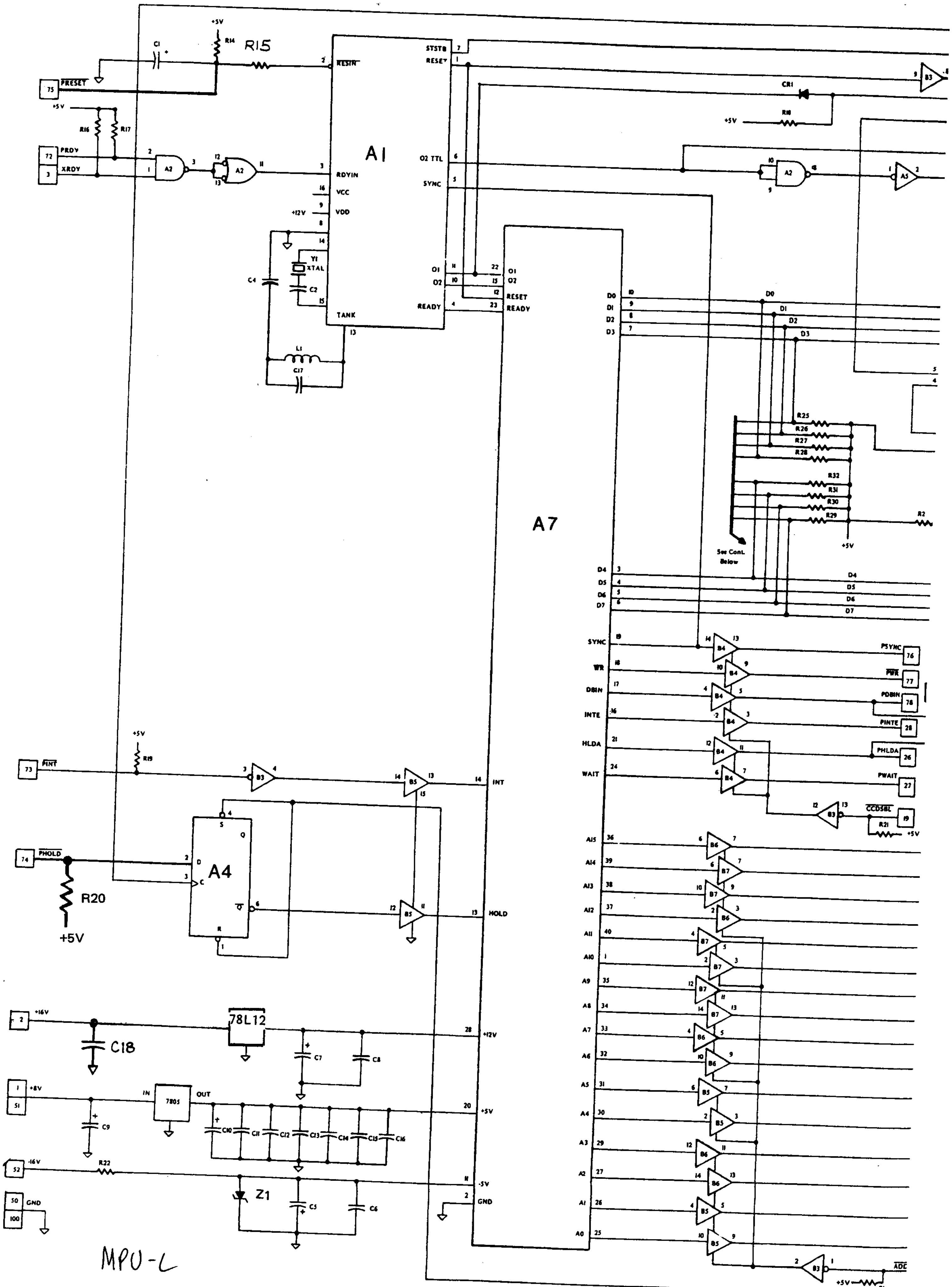


U1, U3, U4, U6, U7	7406
U2	DATA HEADER SOCKET
U5	8212
U8, U15	74LS04
U9	74LS30
U10	7404
U11	74LS10
U12	7410
U13, U16	7402
U14 U25	7400
U15.5	7427
U17, U20, U23	74123
U18, U19, U22	74107
U21	7430
U24	8T97
C1, C2, C5, C6, C7, C11 thru C22	.1mF
C3, C8	.001mF
C4	.01mF
C9, C10	33mF
CR1	1N914
LO0 thru LO7, LA0 thru LA15, LD0 thru LD7, LS0 thru LS7, LIE, LHT, LRN, LWT	LED'S
R1	270KΩ¼W
R2, R6, R7, R10, R11, R13 thru R15, R60	1KΩ¼W
R3, R4, R5, R8, R9, R12	47KΩ¼W
R16 thru R59	220 ¼W

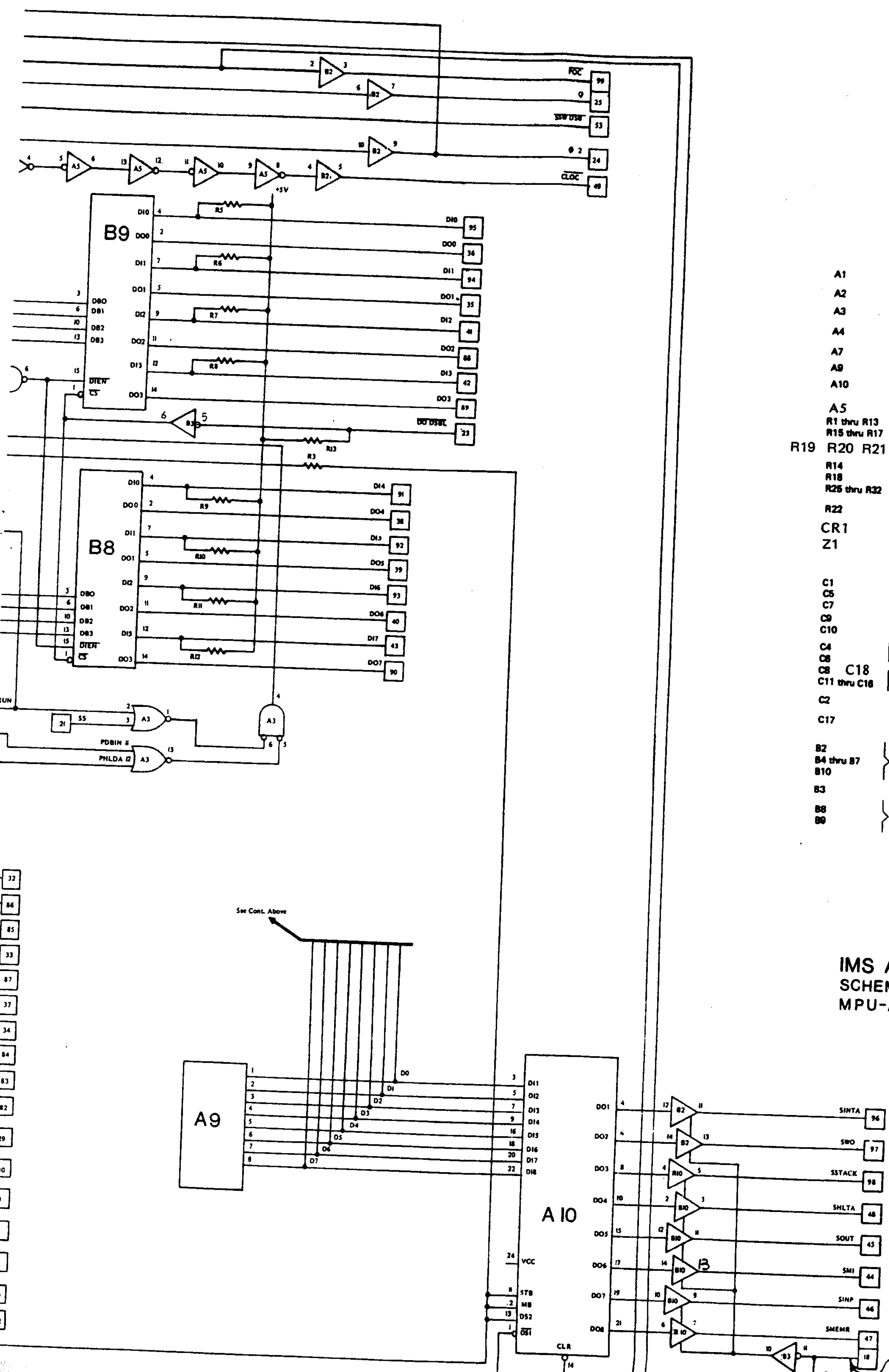
IMS Associates Inc.
SCHEMATIC DIAGRAM
CPA REV 4 2/76

2/27/76

CPA-R


A1
A7
A4
R15
R20
+5V
C18
78L12
Z1
7805
PRESET
PRDY
XRDY
PINT
PHOLD
RESIN
RDYIN
STSTB
RESET
SYNC
READY
TANK
XTAL
INT
HOLD
GND
PSYNC
PWR
PDBIN
PINTE
PHLDA
PWAIT
CCDSBL
ADC

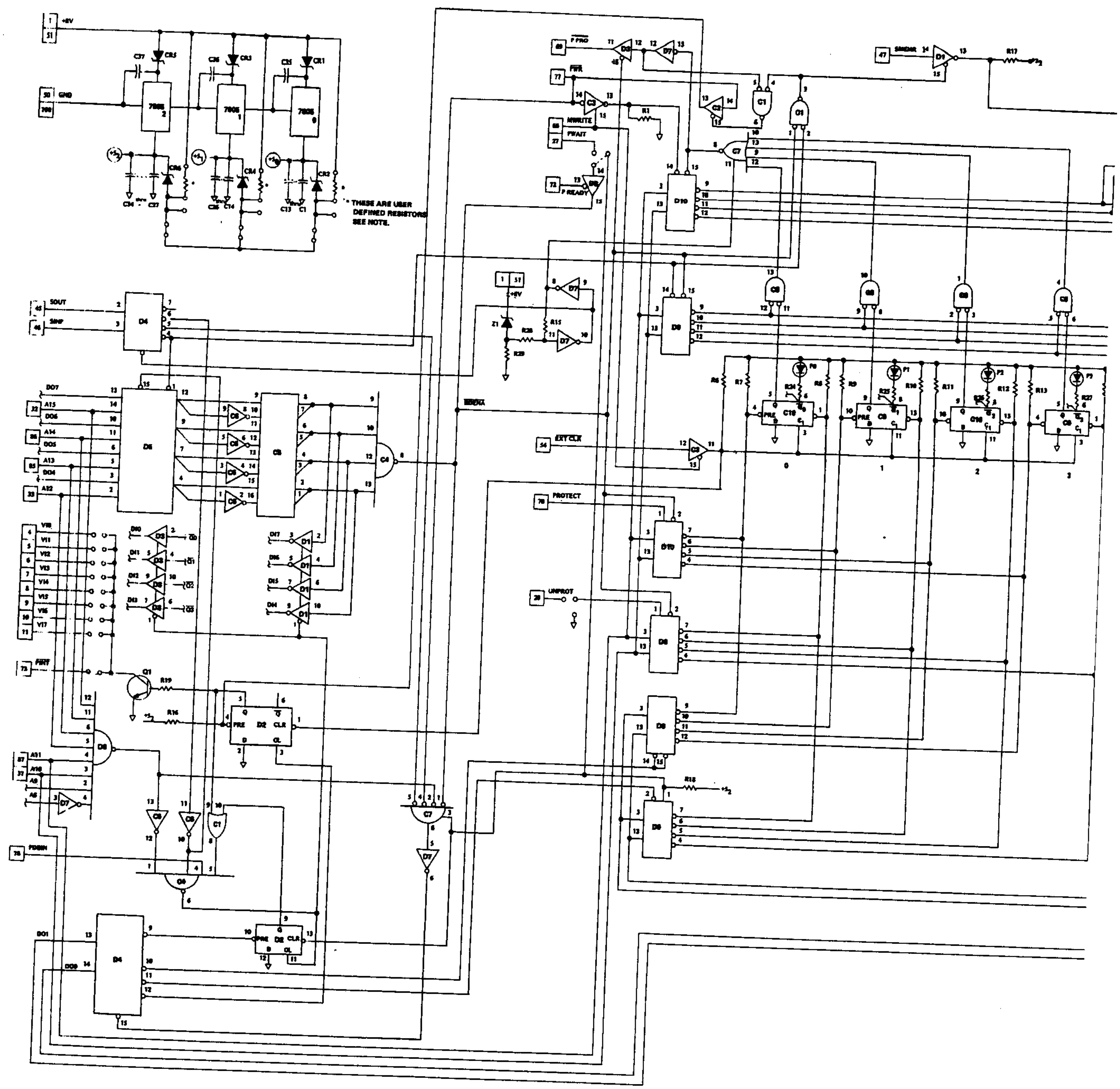
MPU-L



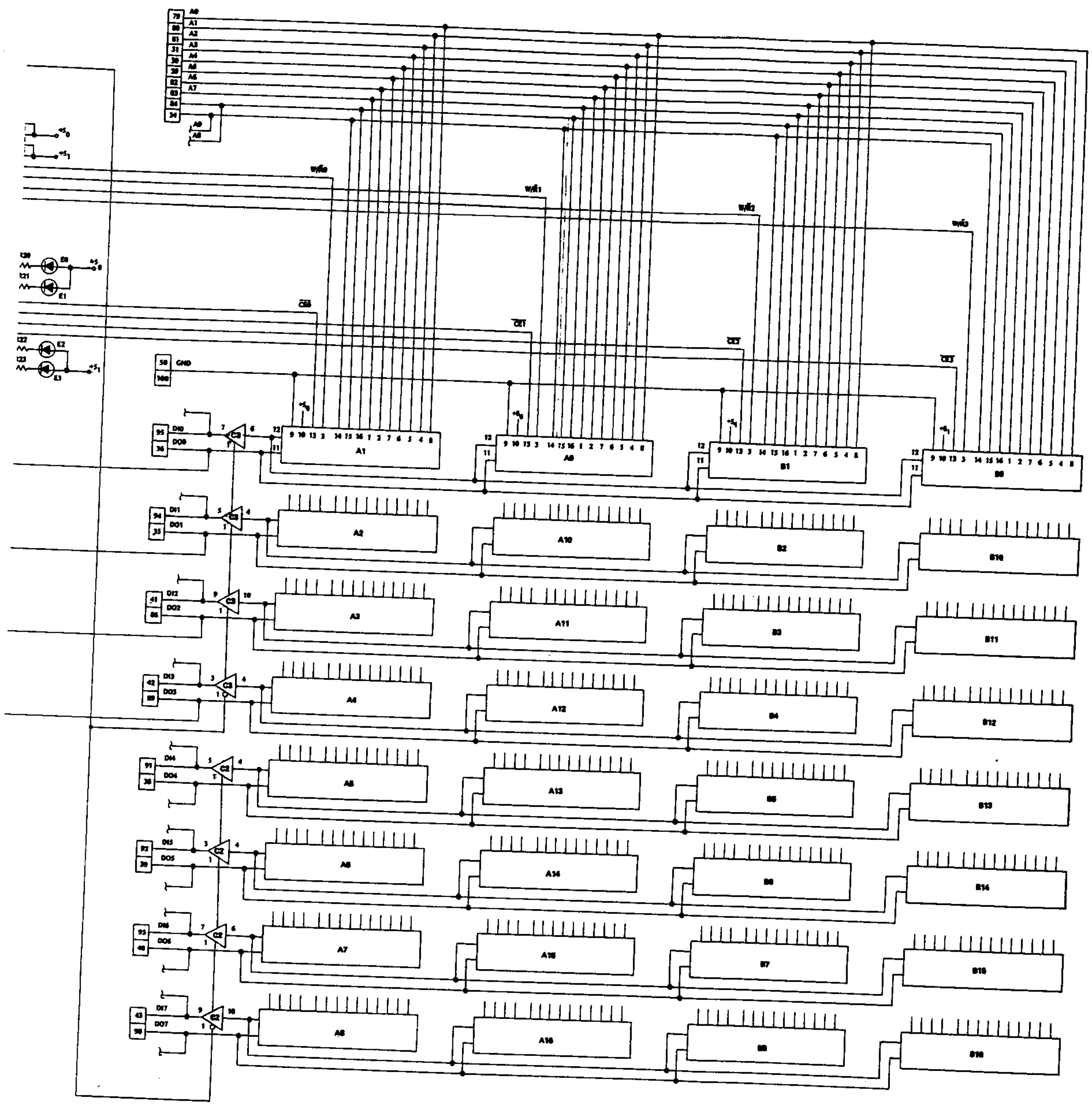
A1	8224
A2	74LS00
A3	74LS02
A4	7474
A7	8080A
A9	Data Bus Socket
A10	8212
A5	74ls04
R1 thru R13, R15 thru R17, R19 R20 R21	1K ¼W
R14, R18, R25 thru R32	4.7K ¼W
R22	470 ¼W
CR1	1N914
Z1	1N751
C1, C5, C7, C9, C10	33mF
C4, C6, C8 C18, C11 thru C16	.1mF
C2	39pF
C17	56pF
B2, B4 thru B7, B10	8T97
B3	74LS04
B8, B9	8216

IMS ASSOCIATES INC
SCHEMATIC DIAGRAM
MPU-A REV 4 2/76
5/18/76

MPU-R


* THESE ARE USER DEFINED RESISTORS SEE NOTE.
SOUT
SINP
PROTECT
UNPROT
EXT CLK
PWAIT
P READY
MWRITE
PDBIN

RAM-L

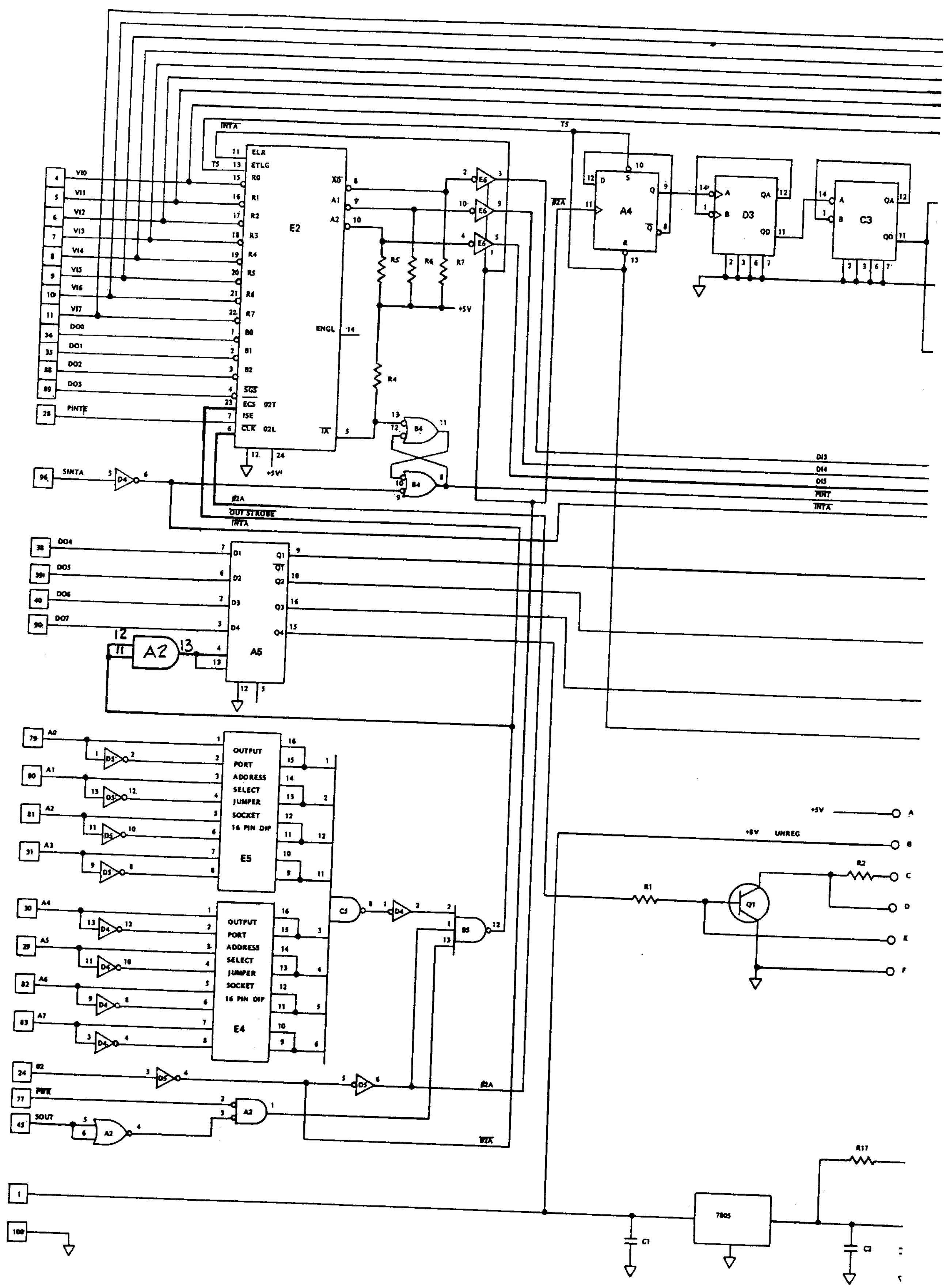


A1 thru A16	2102
B1 thru B16	2102
C1	7432
C2 C3	8T97
C4	74LS20
C5	16 PIN JUMPER SOCKET
C6	7404
C7	7426
C8	7402
C9 C10	74LS74
D1	8T96
D2	74LS74
D3	8T97
D4	74LS139
D5	74LS157
D6	7430
D7	7404
D8 D9 D10	74LS158
C1 thru C34	.1 uF
C35 thru C37	33 uF
CR1 thru CR6	1N4002
Z1	1N751A
Q1	2N3904
R1 thru R18	1K ¼ WATT
R19 thru R28	220 ¼ WATT
R29	100 ¼ WATT
E0 thru E3	GREEN LED
F0 thru F3	RED LED

NOTE *:
THESE ARE USER DEFINED RESISTORS. REFER TO USERS GUIDE FOR EXPLANATION.

IMS ASSOCIATES INC.
SCHEMATIC DIAGRAM
RAM 4A-4 REV. 2 6/76

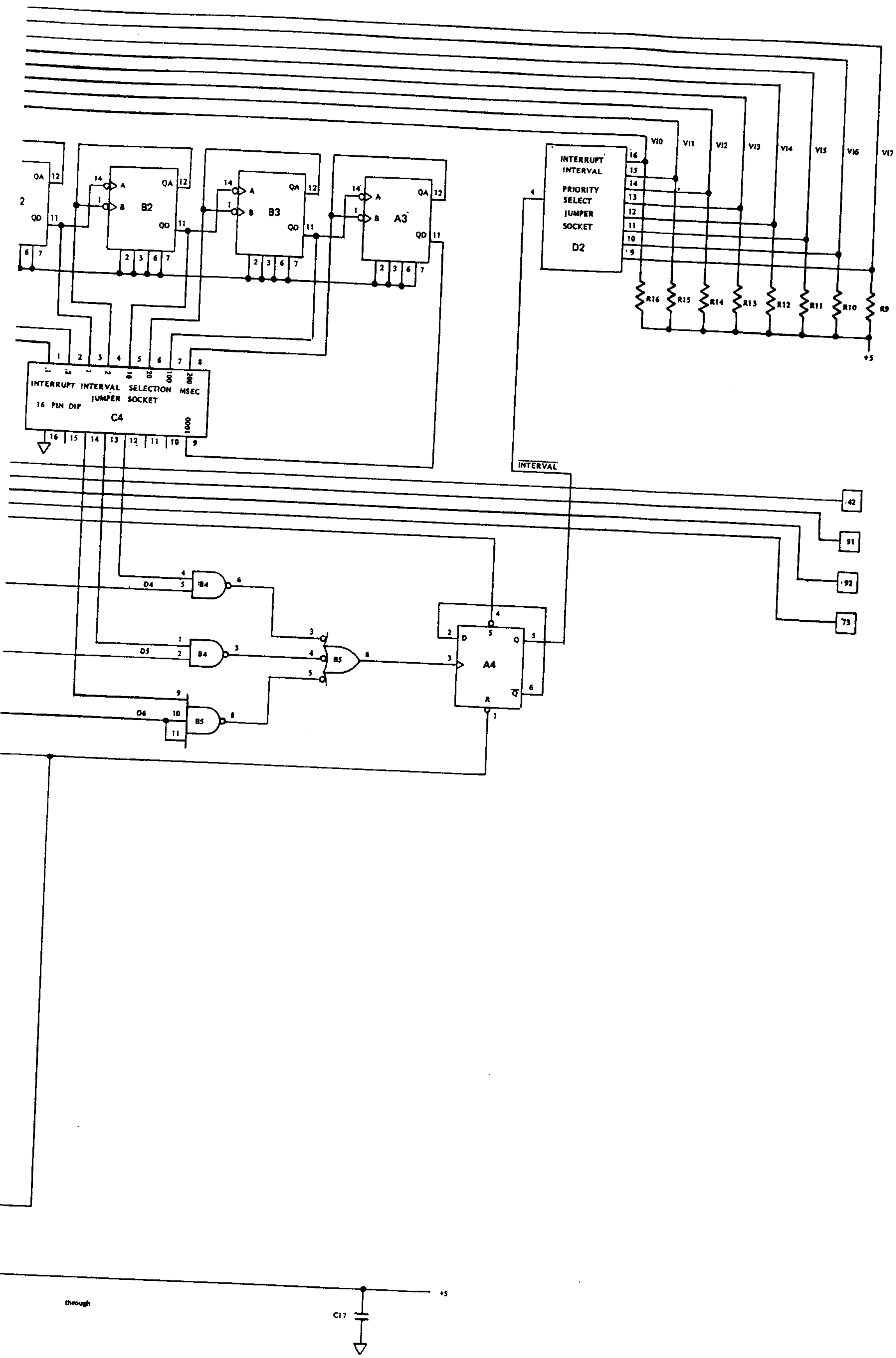
RAM-R



EASI FILL CORP. HANGER EFH135

PIC-3

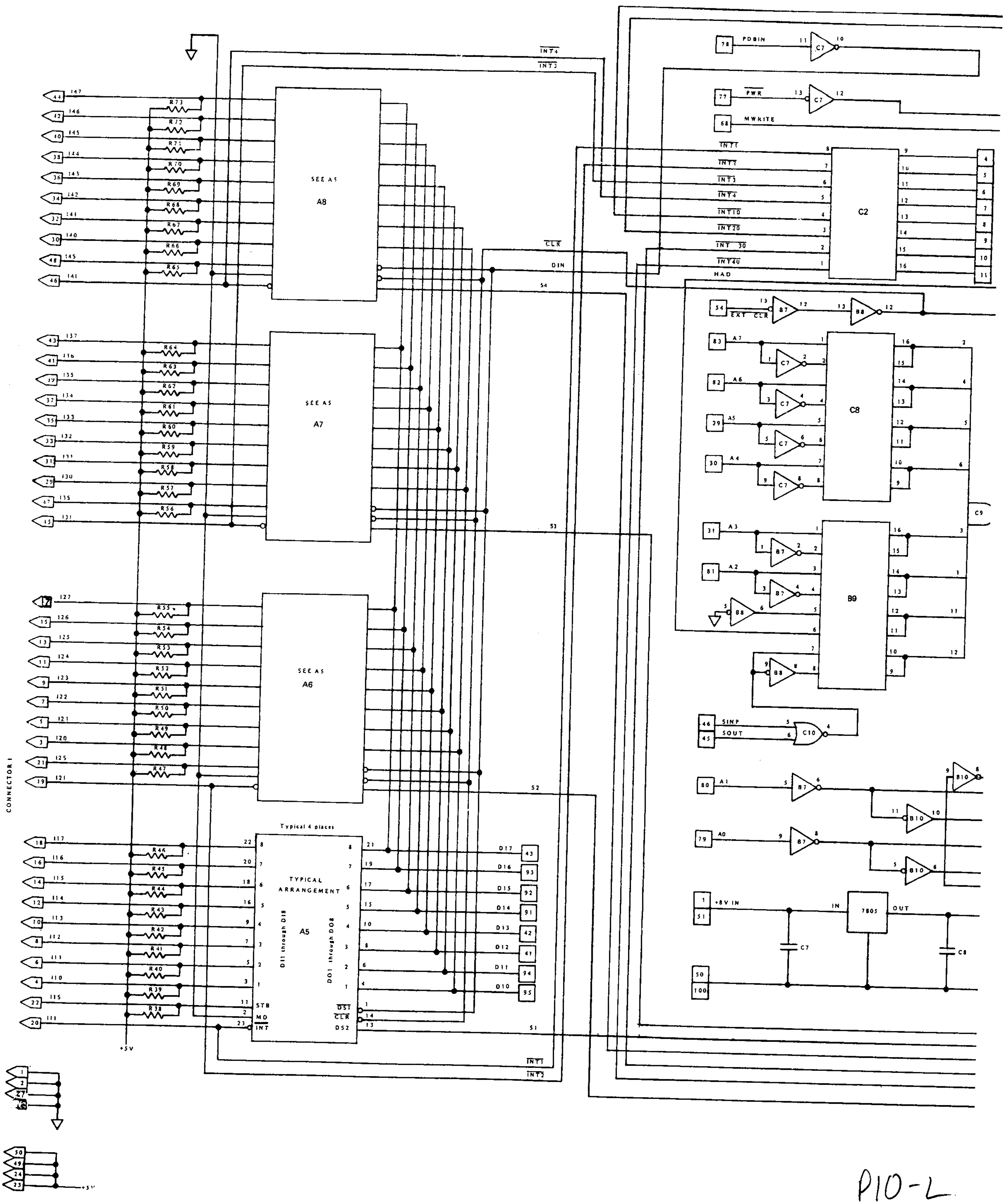
PIC-L



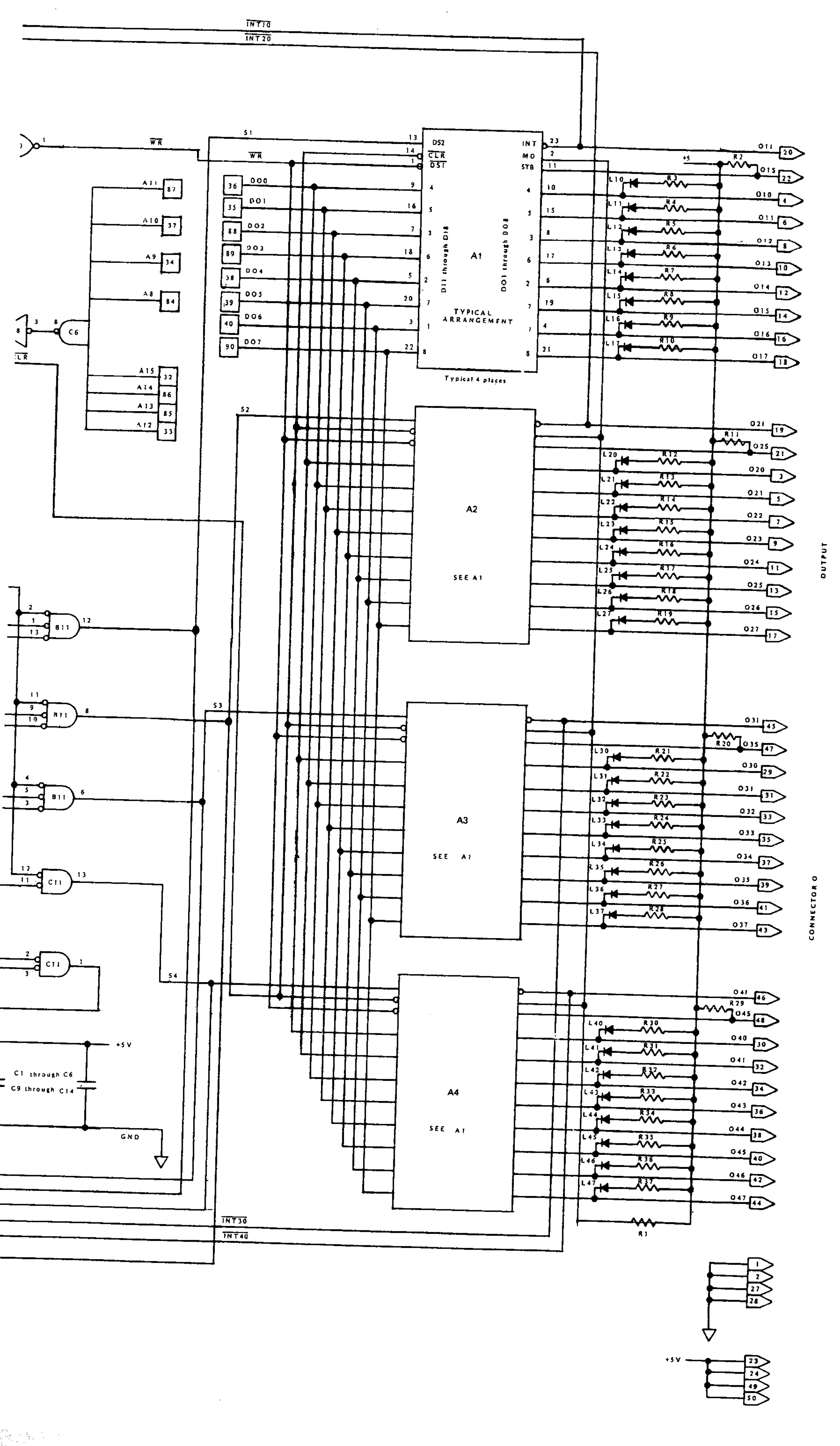
A2	74LS02
A3	7490
A4	7474
A5	74LS75
B2, B3	7490
B4	7400
B5	7410
C2, C3	7490
C4	JUMPER SOCKET
C5	74LS30
D2	JUMPER SOCKET
D3	7490
D4, D5	74LS04
E2	8214
E4, E5	JUMPER SOCKET
E6	8T98
C1, C2	33mf
C3 thru C17	.1mF
Q1	2N3904
R1, R3 thru R16	1K¼W
R2	220 ¼W

PIC-R

IMS Associates Inc.
SCHEMATIC DIAGRAM
PIC 8 REV 3 2/76
2/27/76


SEE A5
A8
SEE A5
A7
SEE A5
A6
Typical 4 places
TYPICAL ARRANGEMENT
A5
DI1 through DI8
DO1 through DO8
CONNECTOR 1
C2
C8
B9
C9
7805
PDBIN
PWR
MWRITE
EXT CLR
SINP
SOUT
+8V IN
IN
OUT
STB
MD
INT
DS1
CLR
DS2
DIN
+5V

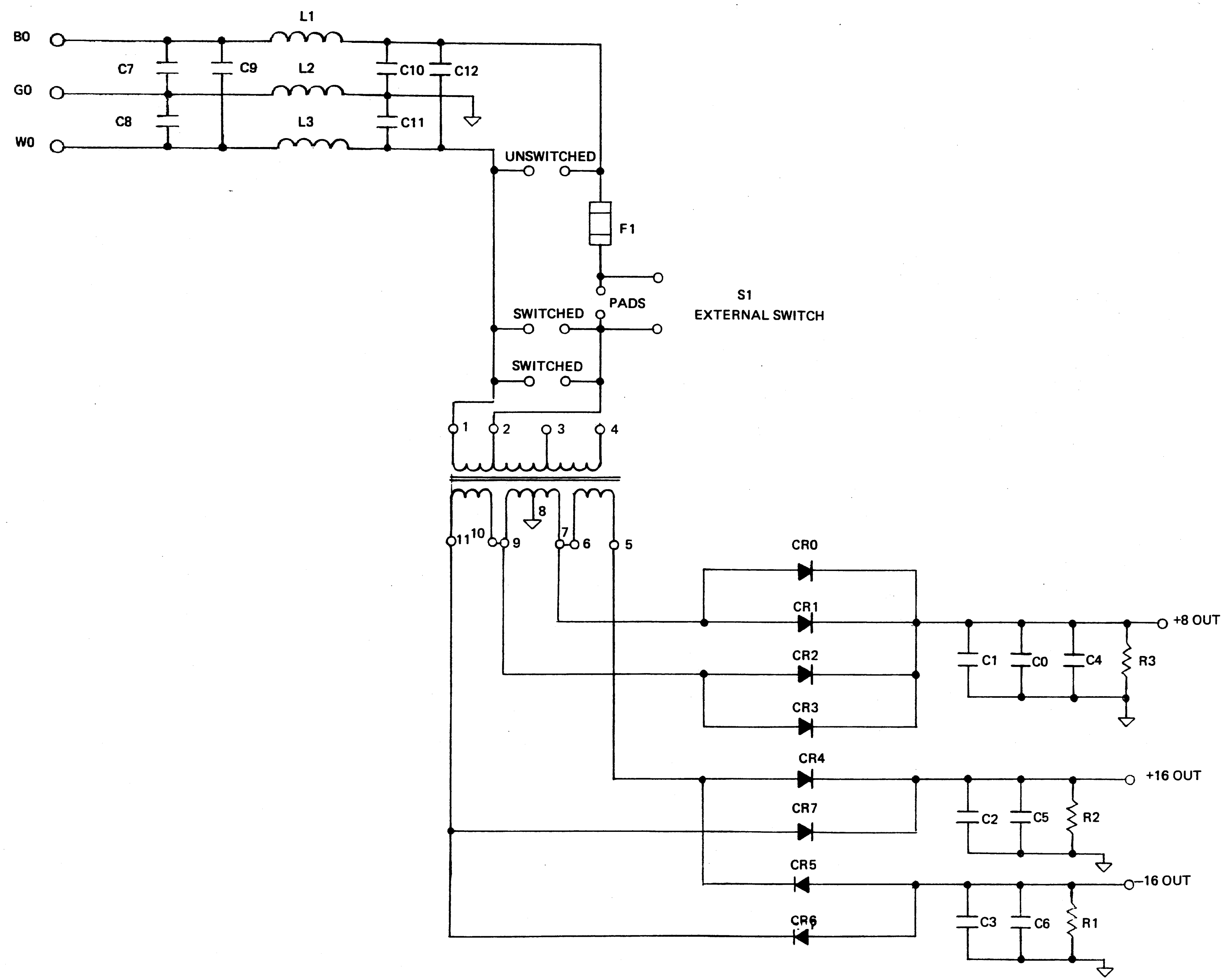
P10-L



A1 thru A8	8212
C7, B7	74LS04
B8, B10	7404
B9	JUMPER SOCKET
B11	7427
C2	JUMPER SOCKET
C6, C9	74LS30
C8	JUMPER SOCKET
C10	74LS02
C11	7402
C1 thru C6, C9 thru C13	.1mF
C7, C8	33mF
L10 thru L17, L20 thru L27, L30 thru L37, L40 thru L47	LED'S
R1, R2, R11, R20, R29, R38 thru R73	1K¼W
R3 thru R10, R12 thru R19, R21 thru R28, R30 thru R37	220 ¼W

IMS Associates Inc.
SCHEMATIC DIAGRAM
PIO REV. 2 2/76

PIO-R



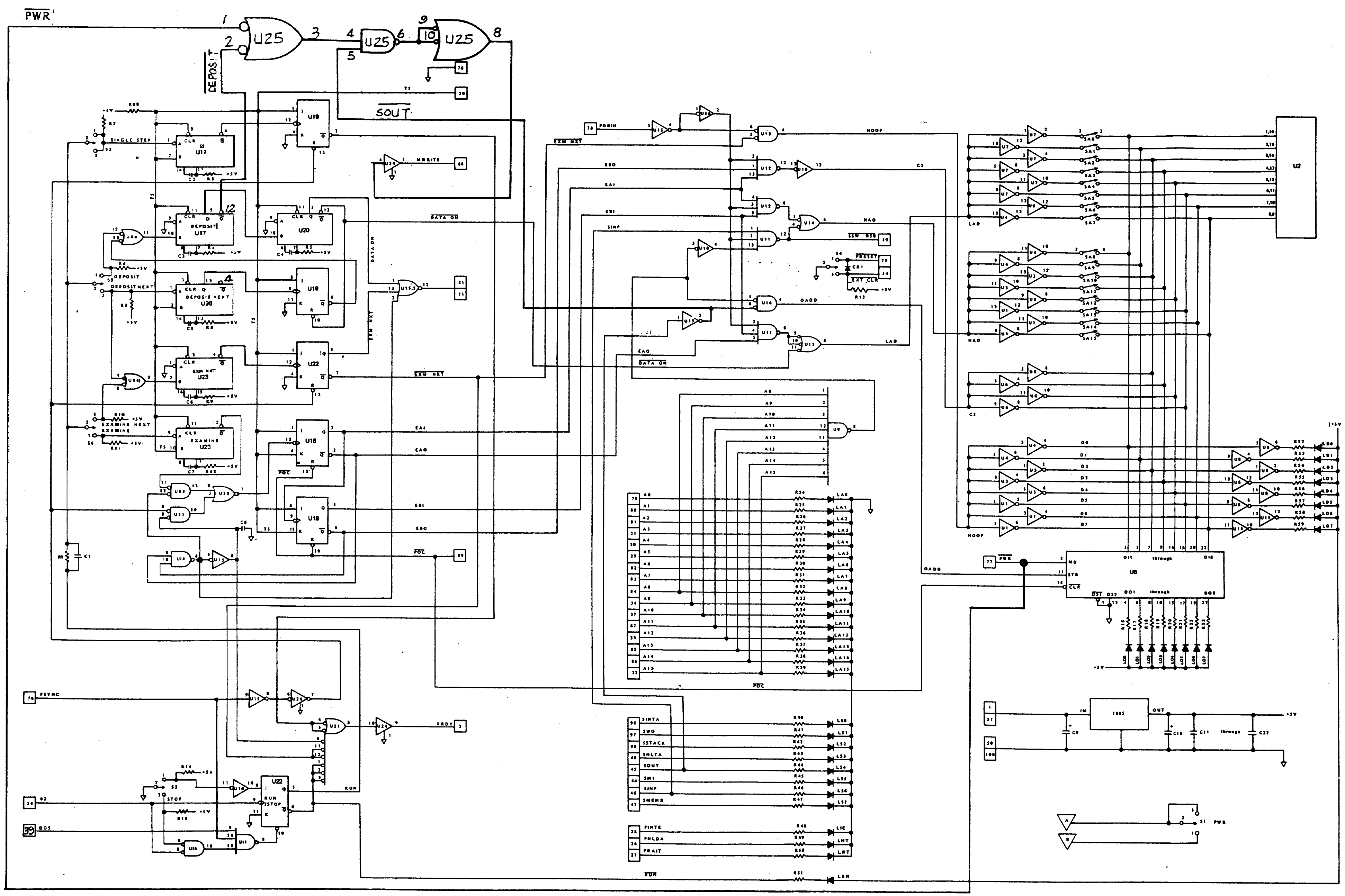
C0, C1	95K uF
C2, C3	10K uF
C4, C5, C6	.1 uF
C7 thru C12	.04 uF 500V
CR0 thru CR3	MR1121
CR4 thru CR7	MR501
F1	5A
L1 thru L3	8uH
R1, R2	1K ½W
R3	470 ½W
S1	NOT SUPPLIED

IMS ASSOCIATES INC.
SCHEMATIC DIAGRAM
PS-C REV. 1 5/76

PSC

CASI FILE CORP.

CPA

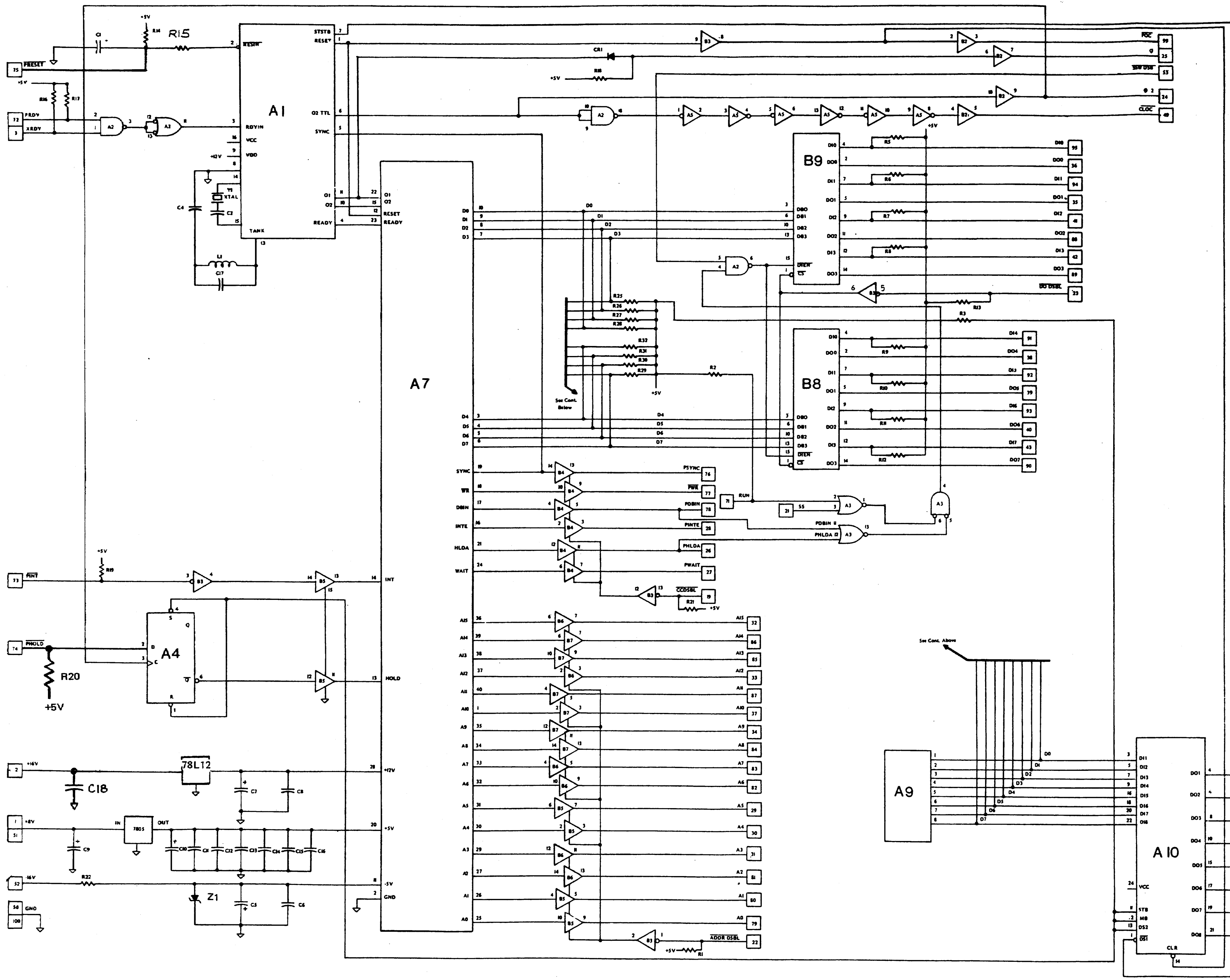


U1, U3, U4, U6, U7	7406
U2	DATA HEADER SOCKET
U5	8212
U8, U15	74LS04
U9	74LS30
U10	7404
U11	74LS10
U12	7410
U13, U16	7402
U14, U25	7400
U15.5	7427
U17, U20, U23	74123
U18, U19, U22	74107
U21	7430
U24	8T97
C1, C2, C5, C6, C7, C11 thru C22	.1mF
C3, C8	.001mF
C4	.01mF
C9, C10	33mF
CR1	1N914
LO0 thru LO7, LA0 thru LA15, LD0 thru LD7, LS0 thru LS7, LIE, LHT, LRN, LWT	LED'S
R1	270K¼W
R2, R6, R7, R10, R11, R13 thru R15, R60	1K¼W
R3, R4, R5, R8, R9, R12	47K¼W
R16 thru R59	220 ¼W

IMS Associates Inc.
SCHEMATIC DIAGRAM
CPA REV 4 2/76

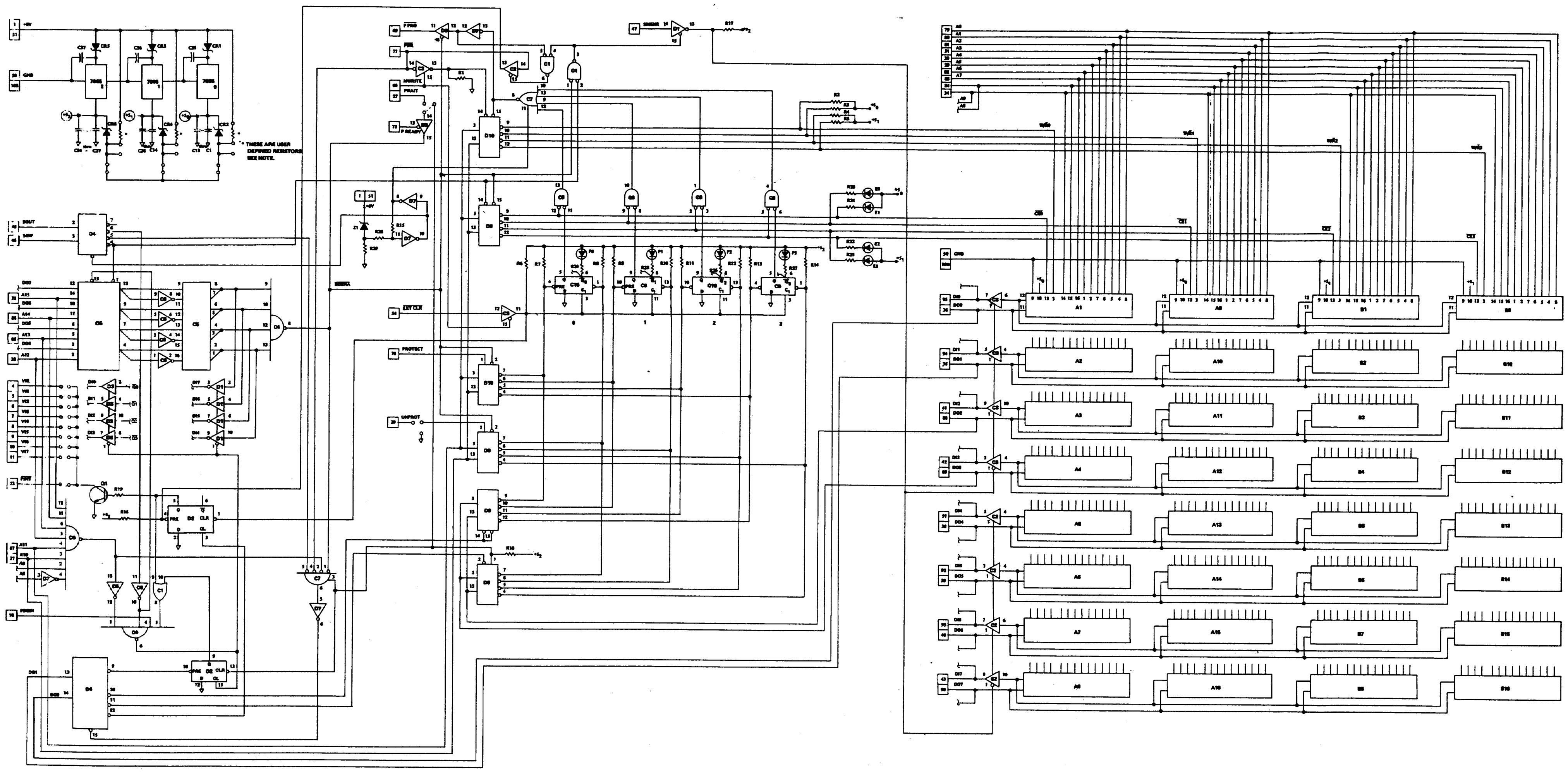
2/27/76

CPA



A1	8224
A2	74LS00
A3	74LS02
A4	7474
A7	8080A
A9	Data Bus Socket
A10	8212
A5	74ls04
R1 thru R13, R15 thru R17, R19 R20 R21	1K ¼W
R14, R18, R25 thru R32	4.7K ¼W
R22	470 ¼W
CR1	1N914
Z1	1N751
C1, C5, C7, C9, C10	33mF
C4, C6, C8, C18, C11 thru C16	.1mF
C2	39pF
C17	56pF
B2, B4 thru B7, B10	8T97
B3	74LS04
B8, B9	8216

IMS ASSOCIATES
SCHEMATIC DIAG
MPU-A REV 4
5/18/76



A1 thru A16	2102
B1 thru B16	2102
C1	7432
C2, C3	8T97
C4	74LS20
C5	16 PIN JUMPER SOCKET
C6	7404
C7	7425
C8	7402
C9, C10	74LS74
D1	8T98
D2	74LS74
D3	8T97
D4	74LS138
D5	74LS157
D6	7430
D7	7404
D8, D9, D10	74LS189
C1 thru C34	.1 uF
C35 thru C37	33 uF
CR1 thru CR6	1N4002
Z1	1N751A
Q1	2N3904
R1 thru R18	1K ¼ WATT
R19 thru R28	220 ¼ WATT
R29	100 ¼ WATT
E0 thru E3	GREEN LED
F0 thru F3	RED LED

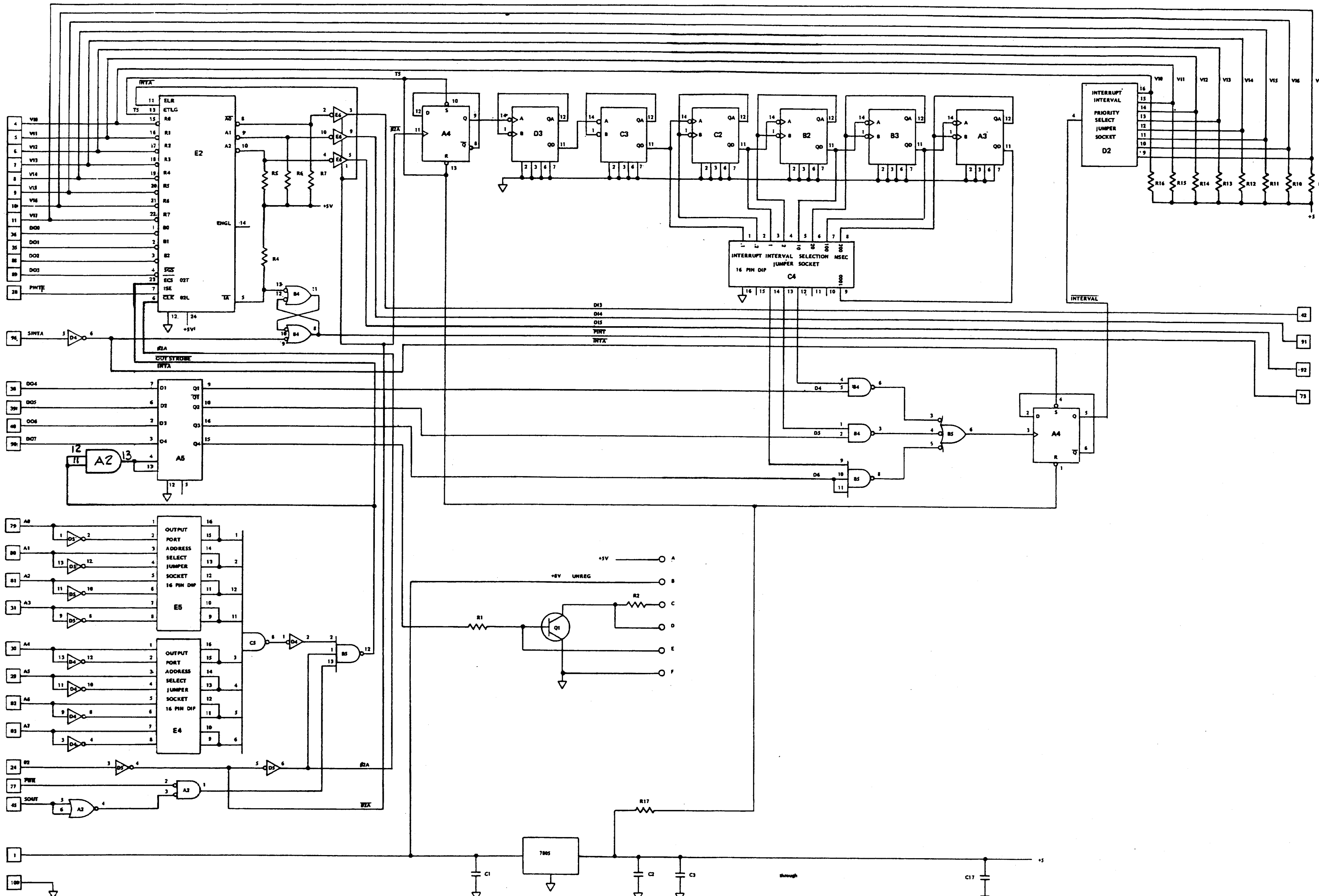
NOTE *:
THESE ARE USER DEFINED RESISTORS. REFER TO USERS GUIDE FOR EXPLANATION.

IMS ASSOCIATES INC.
SCHEMATIC DIAGRAM
RAM 4A-4 REV. 2 5/76

RAM

EASI FILL CORP. HANGER EFH115

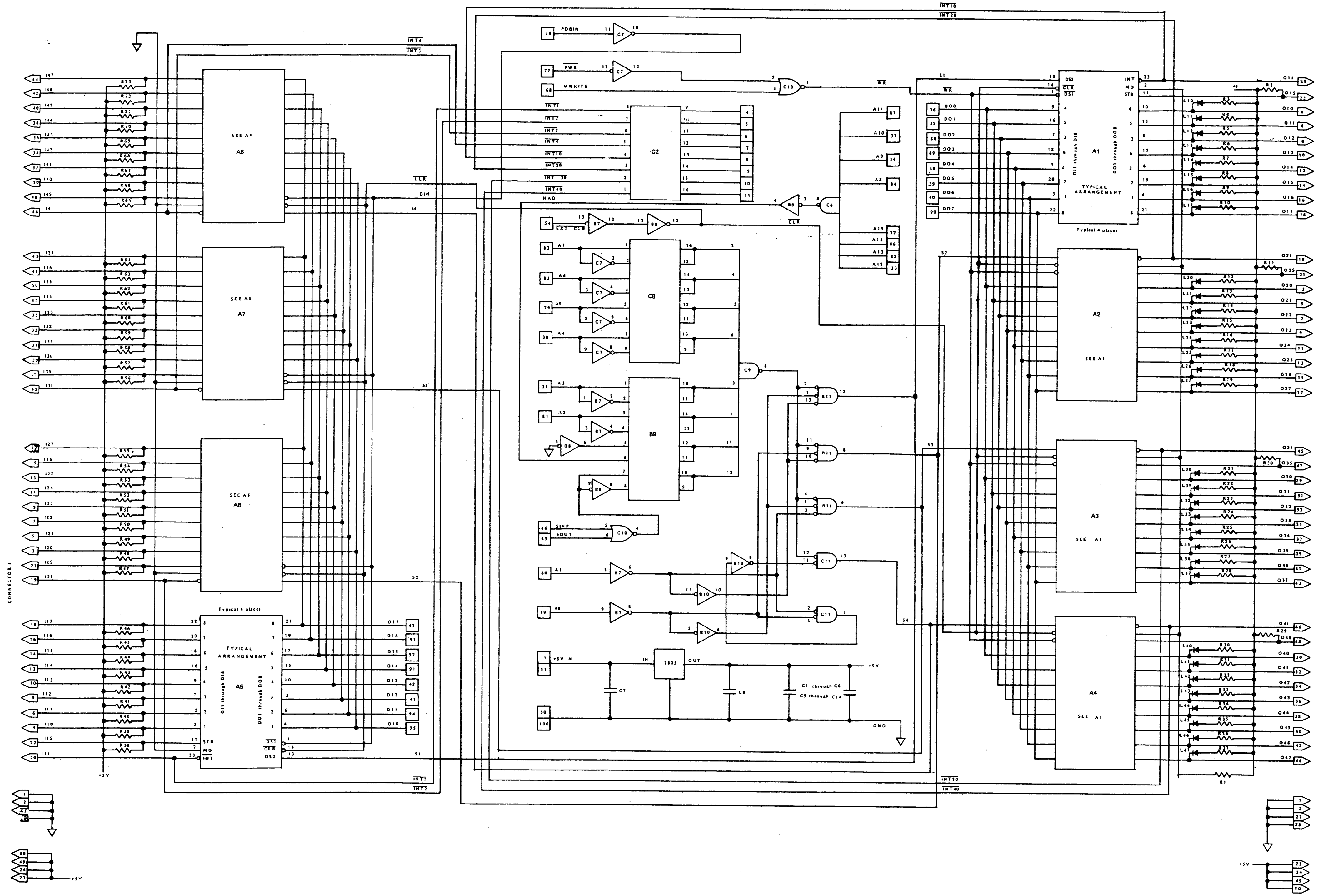
PIC-3



A2	74LS02
A3	7490
A4	7474
A5	74LS75
B2, B3	7490
B4	7400
B5	7410
C2, C3	7490
C4	JUMPER SOCKET
C5	74LS30
D2	JUMPER SOCKET
D3	7490
D4, D5	74LS04
E2	8214
E4, E5	JUMPER SOCKET
E6	8T98
C1, C2	33mf
C3 thru C17	.1mF
Q1	2N3904
R1, R3 thru R15	1K¼W
R2	220¼W

PIC

IMS Associates Inc.
SCHEMATIC DIAGRAM
PIC 8 REV 3 2/76
2/27/76



A1 thru A8	8212
C7, B7	74LS04
B8, B10	[illegible]
B9	JUMPER SOCKET
B11	7427
C2	JUMPER SOCKET
C6, C9	74LS30
C8	JUMPER SOCKET
C10	74LS02
C11	7402
C1 thru C6, C9 thru C13	.1mF
C7, C8	33mF
L10 thru L17, L20 thru L27, L30 thru L37, L40 thru L47	LED'S
R1, R2, R11, R20, R29, R38 thru R73	1K¼W
R3 thru R10, R12 thru R19, R21 thru R28, R30 thru R37	220 ¼W

IMS Associates Inc.
SCHEMATIC DIAGRAM
PIO REV. 2 2/76

©1976

PIO