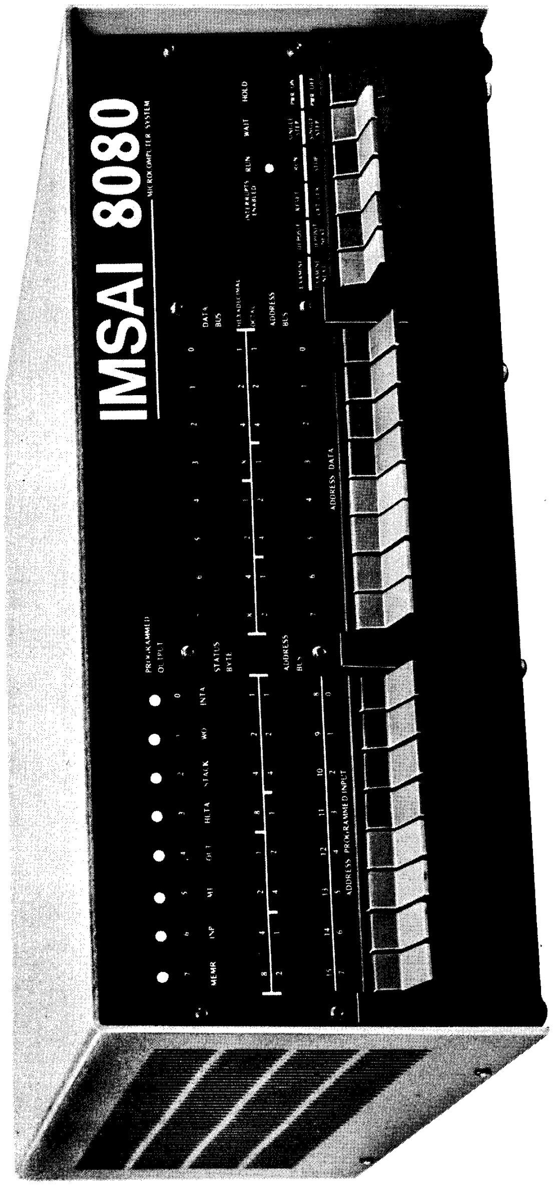
andard of Excellence in Microcomputer Systems.

IMSAI

PCS 80/10 Microcomputer System

user manual

IMSAI Manufacturing Corporation
San Leandro, CA


IMSAI 8080
MICROCOMPUTER SYSTEM
PROGRAMMED OUTPUT
7 6 5 4 3 2 1 0
MEMR INP M1 OUT HLTA STACK WO INTA
STATUS BYTE
DATA BUS
7 6 5 4 3 2 1 0
HEXADECIMAL
OCTAL
ADDRESS BUS
15 14 13 12 11 10 9 8
7 6 5 4 3 2 1 0
ADDRESS PROGRAMMED INPUT
ADDRESS DATA
INTERRUPTS ENABLED
RUN
WAIT
HOLD
EXAMINE
EXAMINE NEXT
DEPOSIT
DEPOSIT NEXT
RESET
EXT CLR
RUN
STOP
SINGLE STEP
SINGLE STEP
PWR ON
PWR OFF

CAUTION

FAILURE TO OBSERVE THESE IMPORTANT PRECAUTIONS WILL VOID WARRANTY

1. Read all material before beginning construction.
2. Use ONLY electronic quality rosin core solder.
3. Use extreme care with static-sensitive chips to prevent static discharge damage. (These chips are inserted in black conductive foam material in your kit.)
4. Do NOT plug or unplug boards while power is on.
5. Do NOT apply power to any board or circuit before checking each component and each trace.
6. Do NOT insert chips in socket before all soldering on the board is completed.
7. Do NOT use nonstandard parts such as fuses of a higher current rating.
8. Do NOT leave out any construction step.
9. Use only specified AC power.
10. Prevent flat cable end from touching areas of the system that may be carrying current.
11. Clean unit with soap and water or isopropyl alcohol only to prevent damage to plastic components.
12. Some repair operations are quite demanding. Do not attempt repairs beyond your level of skill to prevent damage to the board or the components.
13. Use ONLY a 25 watt electronic soldering iron for assembly of your IMSAI kit.
14. Do NOT perform any solder work on a board while power is applied.
15. Do NOT plug or unplug a chip from a socket while power is applied.
16. Check power supply voltages BEFORE inserting any boards into chassis.
17. For all assembled units, read USER GUIDE section for jumpering instructions.
18. To register your kit for warranty protection, fill out warranty cards and mail to IMSAI. Kits without warranty cards on file are NOT covered by warranty.

IMSAI MICROCOMPUTER SYSTEM USER MANUAL

IMSAI Manufacturing Corporation
14860 Wicks Boulevard
San Leandro, California 94577
Made in the U. S. A.

September, 1977

IMSAI 8080 MICROCOMPUTER SYSTEM

USER MANUAL

ERRATA INFORMATION

Errata information will be found immediately preceeding the section to which the information applies and should be used for clarification and/or correction of the section indicated.

CAUTION: FAILURE TO OBSERVE PERTINENT INFORMATION WHICH IS INCLUDED WILL VOID WARRANTY.

CUSTOMER SERVICE

REPLACEMENT PARTS

If you need a replacement part, use only standard parts from commercial sources. Use of surplus or second-run parts will void warranty. If you have trouble locating a part, write IMSAI and include:

- Part number and description as shown in the parts list.
- Serial number of cabinet or board name and revision number.
- Date of purchase.
- Nature of defect.

Note: Parts damaged through carelessness or misuse will not be replaced under warranty.

TECHNICAL CONSULTATION

Need help with your kit or system?

We encourage you to call or write IMSAI for assistance with any technical problems (except program debugging and "customizing" of hardware for special application, which we will not handle).

The effectiveness of our technical assistance depends on the information you furnish. Be sure to include:

- Serial number of cabinet and/or board name and revision number.
- Date of purchase.
- Exact description of problem.
- Everything you have done in attempting to correct the problem.
- All switch positions, connections to other equipment, system configuration, operation procedure, voltage readings and any other information that you think might be useful.

Note: Telephone traffic is lightest at midweek. . . please be sure your manual and all notes are on hand at time of call.

REPAIR SERVICE

Service facilities are available for both warranty and non-warranty repair work. If this service is desired, send IMSAI:

- Name and address.
- Date of purchase.
- Copies of all correspondence and notes relevant to the problem.
- A complete description of the problem.
- Authorization to return your kit C.O.D. for service (IF ANY) and shipping charges.
- The equipment to be repaired should be sent to IMSAI well packed.
- The original packing slip number.

TABLE OF CONTENTS

IMSAI 8080 SYSTEM

GENERAL ASSEMBLY AND TEST INSTRUCTIONS

IMSAI 8080 System
User Manual
Table of Contents

INTRODUCTION

The documentation for the IMSAI 8080 Microcomputer System consists of several books. This volume, the IMSAI 8080 Microcomputer System User Manual, contains a detailed description of the features and configuration of the computer as a complete system. It also describes the printed circuit boards that form the system building blocks. The chapter format begins with a functional description of the system or board, including brief notes about all the features. The actual operation of the system or board is then described in a theory of operations section. The physical and electronic arrangement of the system or board are next shown with a photograph and a schematic. Assembly of the board from a kit is described by assembly drawings or photos, a complete parts list, and assembly instructions in each chapter. Finally, the information that tells the user how to use the design features of the board to implement various functional options is contained in a user guide section for each board.

Operation of the computer as a system is documented in the IMSAI 8080 Microcomputer System User Manual in the chapter General Assembly and Test Instructions and also in the chapter on the CP-A (Front Panel Control Board). The software supplied with the basic unit, consisting of a resident monitor, assembler and text editor is described in the last chapter of the IMSAI 8080 Microcomputer System User Manual. This chapter includes both a description of the software and a complete object listing.

Supporting documentation is provided by a copy of the Intel 8080 Microcomputer System Users Manual, supplied in every system kit to give IMSAI users the primary source of detailed information about the function and instruction set of the logic implemented by the intergrated circuit chip set used in the IMSAI 8080. A handy reference card, the Intel 8080 Assembly Language Reference Card, is supplied to summarize much useful information from this manual.

To assist users in gaining a full perspective on the design and use of microprocessor-based computer systems, IMSAI includes a basic text, An Introduction to Microcomputers.

IMSAI is currently working intensively on both additional hardware (more peripheral systems, the Shared Memory Facility, etc.) and system software. Full documentation of these additions will be made available to IMSAI owners as it is produced.

IMSAI 8080 MICROCOMPUTER SYSTEM FEATURES

The IMSAI 8080 is a high quality microprocessor based computer system offering outstanding capability and flexibility at low cost.

Designed to facilitate simple modular expansion, the system has both the power and the versatility to handle a broad range of data processing needs.

The IMSAI 8080 is currently supported by a broad range of peripheral devices and interfaces, and comes with a basic resident monitor, assembler, and text editor, free of charge. A broad range of high level system software is now under development, and will be available soon in both source and object form to registered IMSAI 8080 owners.

The IMSAI 8080 is available in kit or assembled form. While primarily designed as a commercial computer, the unit is configured to facilitate construction by any careful assembler. High grade industrial quality design and components are used in both kits and assembled units.

Complete documentation is provided with each system, including:

IMSAI 8080 Microcomputer System User Manual (this book).

Intel 8080 Microcomputer Systems User's Manual, completely describing the integrated circuits used, and the instruction set.

An Introduction to Microcomputers, a fundamental textbook on the use of microcomputer systems.

a 90 day warranty on the system is provided. Full factory service is available at a cost commensurate with the work required.

SYSTEM FUNCTIONAL DESCRIPTION

The IMSAI 8080 Microcomputer System is a full-scale general purpose digital computer. Although small in size and low in cost, the system is exceedingly versatile and capable of data processing in the complete spectrum of practical applications.

The IMSAI 8080 used an 8080A microprocessor LSI chip to perform the central processing function. The instruction set provided by the 8080A is described fully in Chapter 4 of 8080 Microcomputer System User's Manual, provided as part of the IMSAI 8080 documentation package.

The IMSAI 8080 system is capable of unlimited expansion, due to the bus structure and IMSAI's exclusive shared memory facility, which permits parallel processing. The computing power that can be made available with the IMSAI 8080 system building blocks exceeds that of any currently available minicomputer.

The operation of the IMSAI 8080 is described in the manual chapters titled "General Assembly and Test Instructions" and in the CP-A Front Panel Control Board chapter. Input/output features are described in the I/O board chapters including SIO (Serial Input/Output board), PIO (Parallel Input/Output board) and UCRI (Cassette Recorder Interface board).

IMSAI 8080 SOFTWARE FEATURES

Basic system software (resident monitor, text editor and assembler) distributed in object form, with listing, and free of charge.

Future software releases are:

4K BASIC - upward compatible to DEC standard SUPER-BASIC

8K BASIC - Upward compatible to DEC standard SUPER-BASIC

12K BASIC - DEC standard SUPER-BASIC compatible

Floppy Disk Operating System

Linkage Editor
Macro-Assembler with relocatable code generation

12K FORTRAN compiler

IMSAI 8080 HARDWARE FEATURES

Flat cable interconnection used throughout.
Absolute minimum point-to-point wiring.
Front panel has programmed output port with LED indicators.
Front panel has large easy-to-use paddle handle switches.
Front panel legends are produced photographically and mounted behind acrylic panel for protection.
Front panel has filler to increase contrast of LED indicators.
Long-life LEDs used throughout.
Front panel circuit designed so that one-shot timing links are non-critical.
No point-to-point wiring to connect or disconnect front panel to or from system.
Attractive custom designed cabinet and panel.
Rackmount cabinet available as special option.
Cabinet designed to facilitate customizing front panel.
Sturdy card cage construction.
Room for 22 cards.
Power supply subchassis with high-current transformer and computer-grade electrolytic capacitors.
Heavy duty power supply supplies power sufficient for a full complement of cards (28 amps, up to 500 watts).
Straight-through back plane wiring. No special purpose slots.
Front panel plugs into any slot to operate machine.
Double-sided printed circuit boards with plated-through holes and solder mask.
All board contact fingers are gold-plated over nickel.
PC board material is glass-fiber-reinforced epoxy laminate.
On-board power regulation. Power is regulated by integrated circuit regulators with thermal current limits.
Tantalum board decoupling capacitors. Ample .luf disk ceramic power decoupling capacitors.
Designed with latest LSI and MSI components. Package count minimized.
Heavy current tri-state bus drivers used throughout the system.
System designed from initial concept for multi-processor, shared memory options.

MICROCOMPUTER SYSTEM SPECIFICATIONS

Processor: 8080A microcomputer chip

Directly Accessible Memory: 65,536 words
Word Size: 1 byte (8 bits)
Register Instruction Cycle Time: 2 microseconds
Basic Machine Cycle Time: 0.5 microseconds
Directly Accessible Input and Output Ports: 256
Machine Instruction Set Size: 78 basic instructions (181 instructions with variants)
Nested Subroutine Call Capability: Limited only by memory size
Interrupt Capability: 8 hardware levels
Registers: 6, plus stack pointer, program counter and accumulator

Memory Type: Semiconductor (1024xl format chips)

Cabinet: Custom aluminum case with acrylic front panel

Dimensions: 19½" x 17" x 7"
Weight: 40 pounds
Front Panel Switch Type: Paddle
Color: IBM blue and grey

Power: 28 amp unregulated power supply with onboard regulators

Power Requirement: Under 50 watts for basic system
Maximum Power Capability: Up to 500 watts
Power Type: 115 VAC, 60 hz. single phase

Connections: Mounting space for 10 EIA-type 25-pin connectors on the back panel. Opening and cable clamp provided for flat cables to exit from the cabinet. 3M flat cable system used throughout.

IMSAI 8080 System
General Assembly and
Test Instructions

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°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

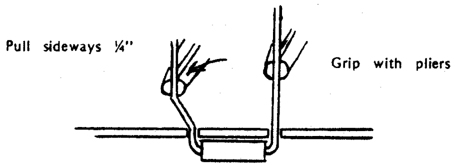
The 7805 regulators for the +5 volts are supplied with a heat sink and mounting hardware. The three leads must be bent down at the proper lengths to match the solder pads, and this should be done with the needle-nose pliers. The lead should come straight out and bend sharply down, rather than slope gradually 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.

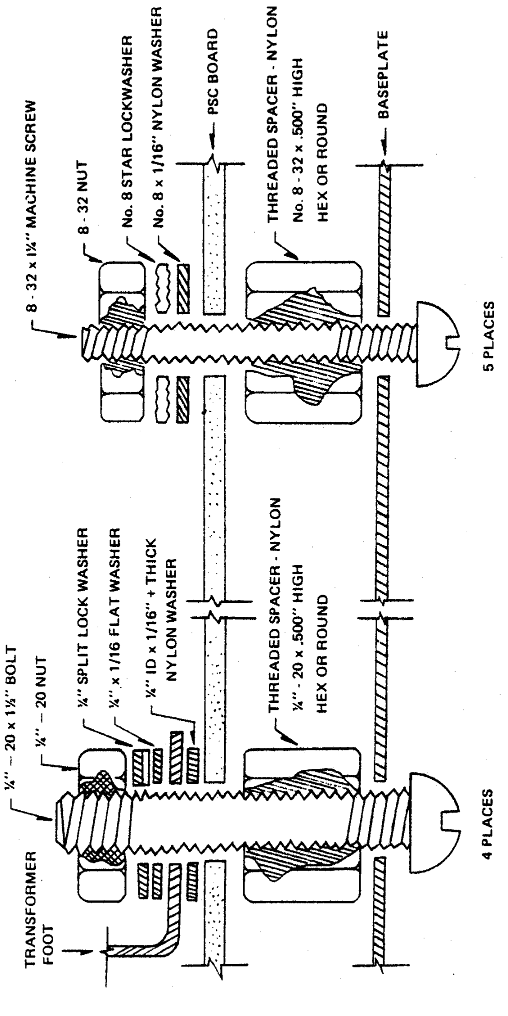
CONNECTION BETWEEN POWER SUPPLY AND MOTHER BOARD

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.

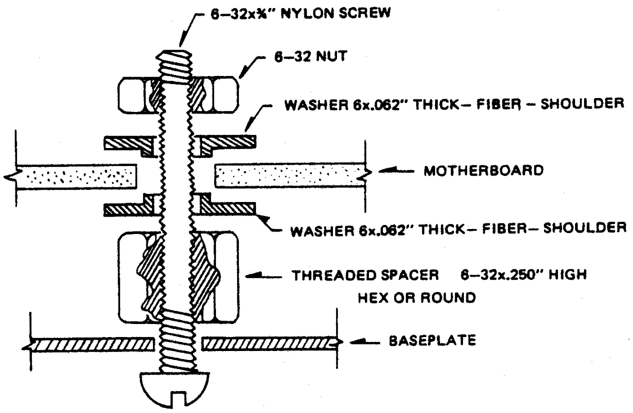
c) 2 or 3 #14 or #12 gauge wire from +8 volt plane to +8 volt bus on Mother board.



PS-C BOARD MOUNTING SYSTEMS

MAINFRAME ASSEMBLY
FIGURE 1

MOTHERBOARD MOUNTING SYSTEM


6–32x¾" NYLON SCREW
6–32 NUT
WASHER 6x.062" THICK– FIBER – SHOULDER
MOTHERBOARD
WASHER 6x.062" THICK– FIBER– SHOULDER
THREADED SPACER 6–32x.250" HIGH
HEX OR ROUND
BASEPLATE

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 knowledgeable 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, Ml, 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 lights. 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 PROGRAMMED OUTPUT port seem to indicate the opposite of what might have been expected when a bit was read in as a 1 and output 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 microprocessors. 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 changed, 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.

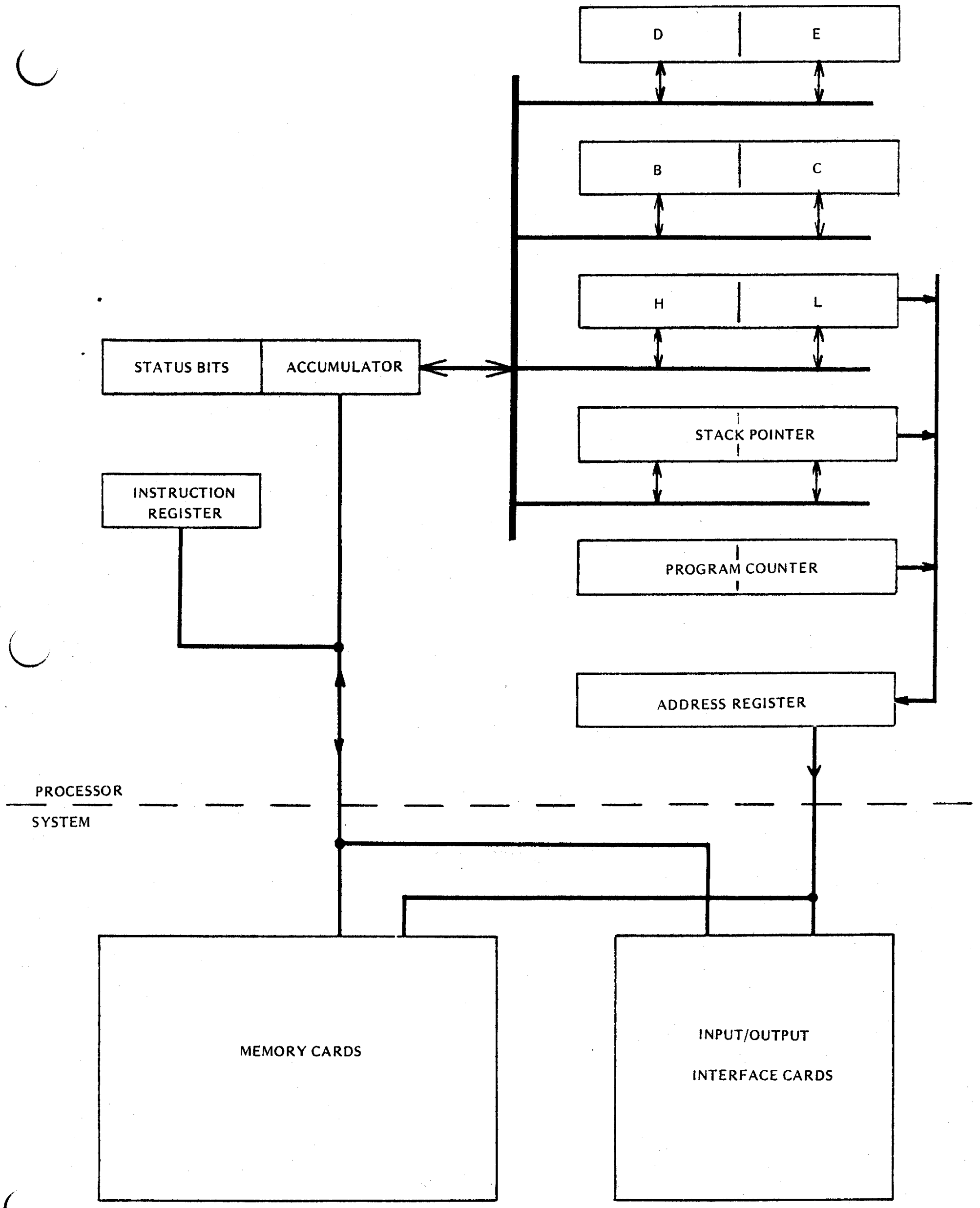
The STATUS BITS are 5 bits that are set to 1 or 0, according to the results of the last data operation performed in the ACCUMULATOR. One of the STATUS BITS or condition flags is the Z bit (zero bit) which is turned on when the last operation in the ACCUMULATOR resulted in the ACCUMULATOR being left all 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

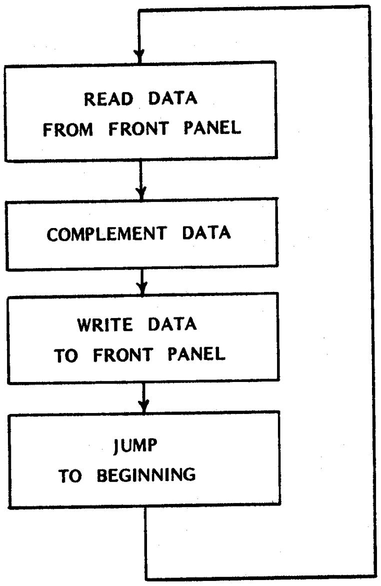

READ DATA
FROM FRONT PANEL
COMPLEMENT DATA
WRITE DATA
TO FRONT PANEL
JUMP
TO BEGINNING

FIG. 2

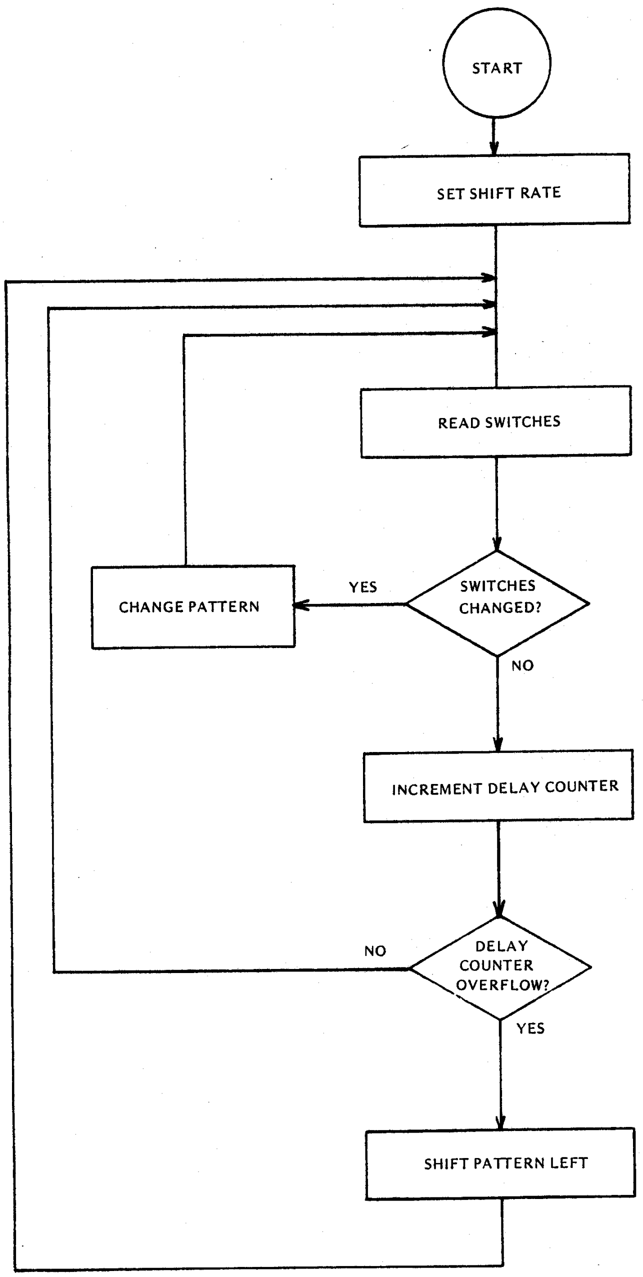


FIG. 3

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		HEX			
ADD.	INST.	ADD.	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 I'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 I's in positions which were changed, with all 0's elsewhere.

B= DISPLAY BYTE STORAGE

C=SWITCH DEBOUNCE INDICATOR
I=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 Al 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 under the base plate and the front frame fits on top 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 (see page 1 - 9).

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 under 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.

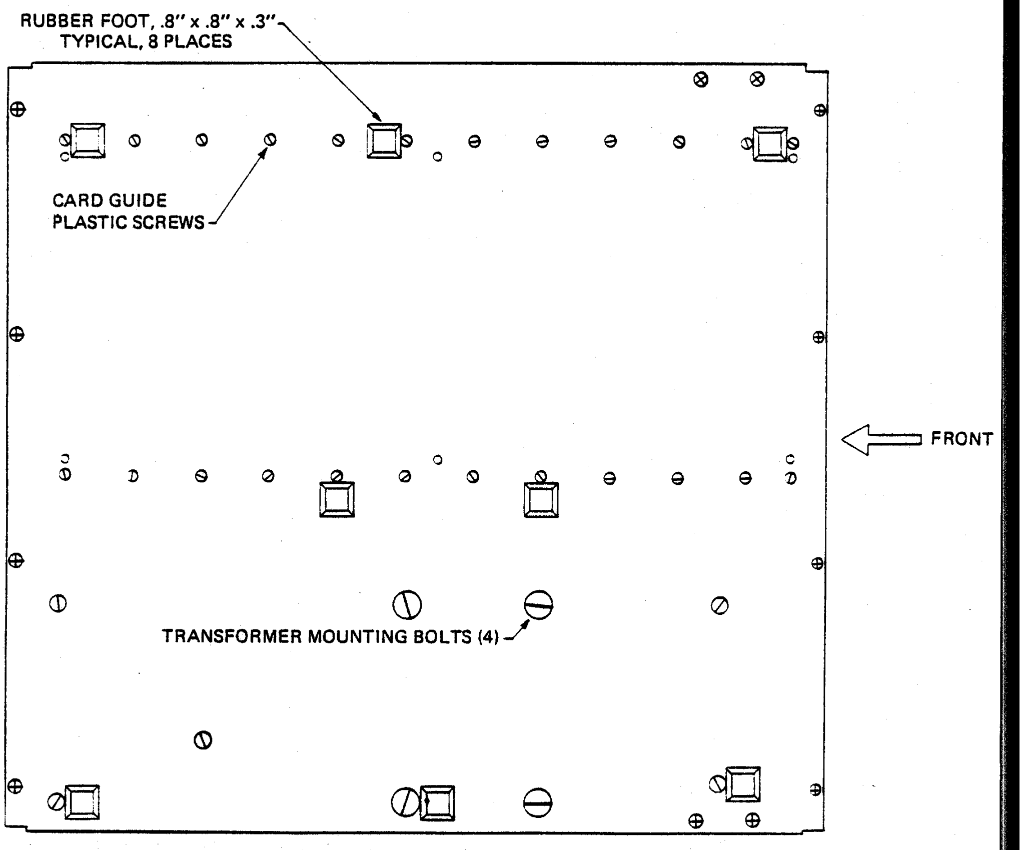
NOTE: for installations without slides where easy removal of the computer is desired, the side plates can be mounted directly in the rack and the computer can be slid on its rubber feet on the bottom flanges of the side plates. The rear of the side plates in this case should be fastened securely to the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

PARTS LIST

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION
Screw	20-3302001	21	6-32x5/16" Phillips Pan Head Machine
Screw	20-3502001	6	6-32x½" Phillips Pan Head Machine
Screw	20-3702001	4	6-32x3/4" Phillips Pan Head Machine
Nut	21-3120001	4	6-32 Hex
Lockwasher	21-3350001	31	#6 Internal Star
Feet	28-0400001	8	Rubber Feet
Guard	34-0200001	1	Plastic Fan Guard
Label	93-0000002	1	8080 "Danger" Sticker
Frame	93-3010001	1	Front Frame Rev. D
Frame	93-3010002	2	Card Frame Rev. C
Frame	93-3010003	1	Back Frame Rev. C
Clamp	93-3010013	1	Cable Clamp Rev. A
Base Plate	93-4010004	1	8080 Chassis Base Plate Rev. C



RUBBER FOOT PLACEMENT
8080 CHASSIS
BOTTOM VIEW

MOTHER BOARD

FUNCTIONAL DESCRIPTION

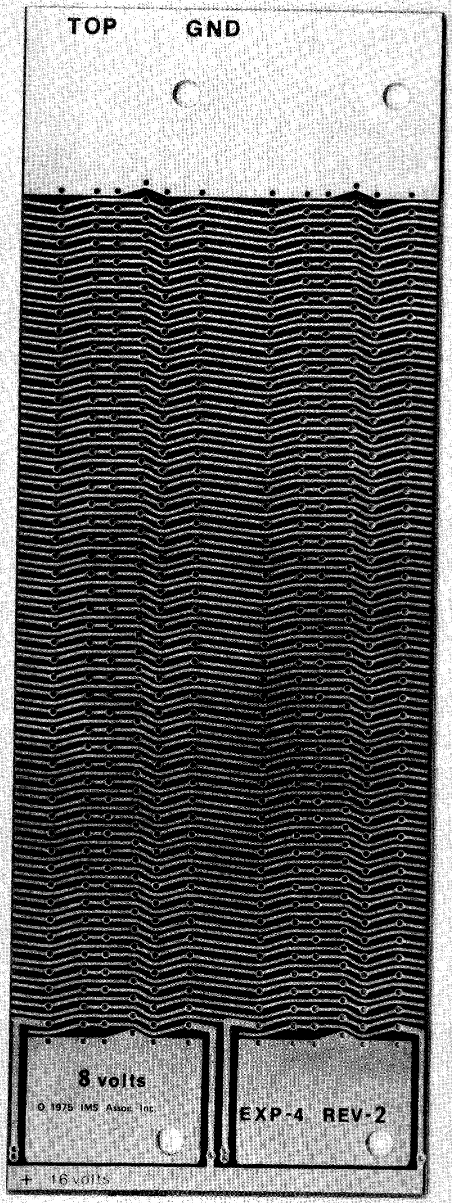
The IMSAI 8080 system Mother boards are available in three different length sections varying from a minimum of 4 printed circuit card connector positions. The basic system includes a Mother board with six connector positions on it. One is used for the front panel and the other five are available for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

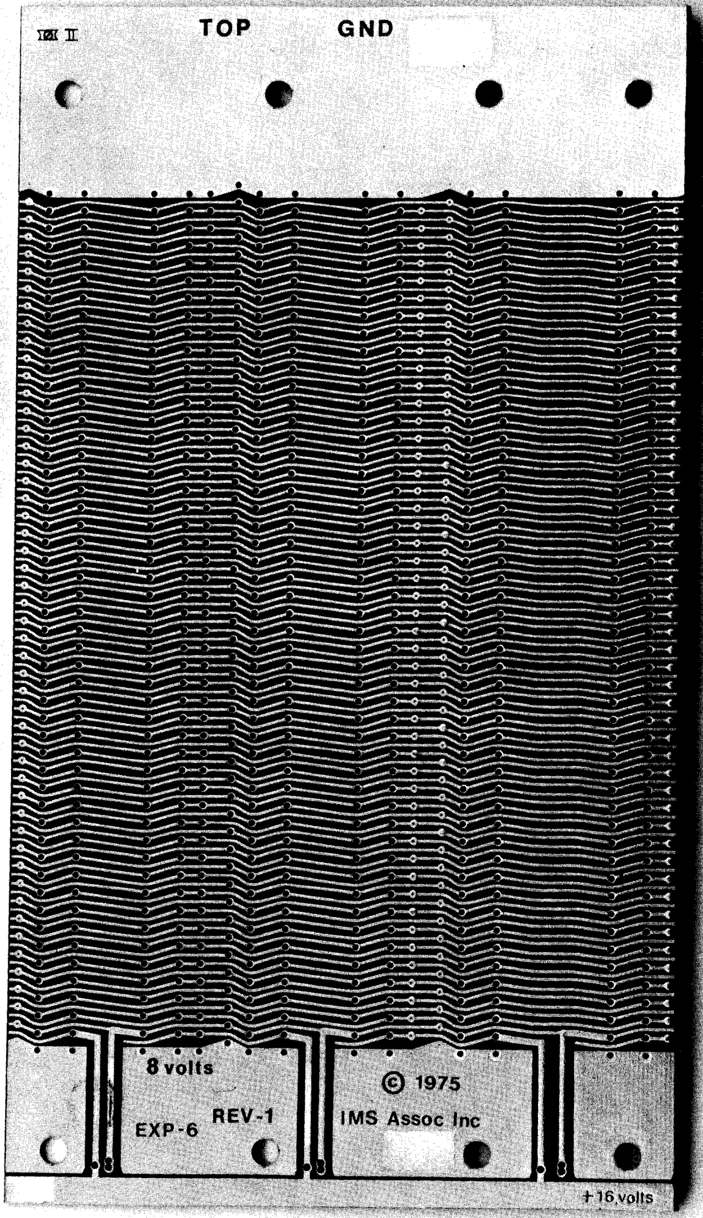
The card-to-card spacing on the Mother board is 3/4 inch except for the front position which is reserved for the front panel board or the parallel I/O board for the dedicated processor to accommodate mounting the card in the special front position in the cabinet.

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.

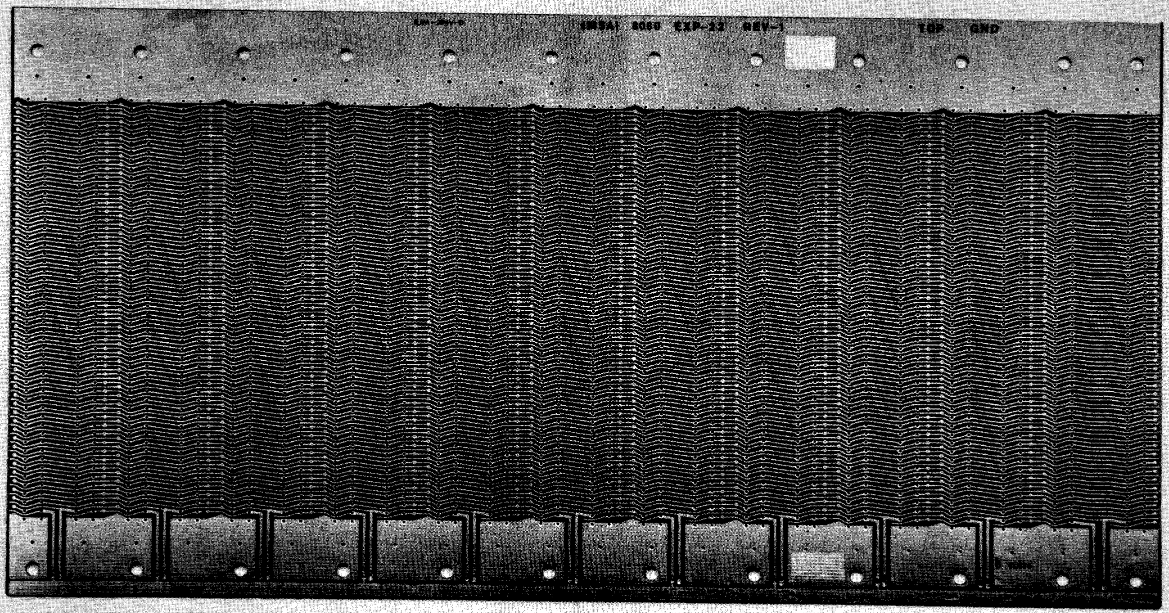
The Mother board is 1/16 inch printed circuit board with double-sided plated-through holes. Each of the connector pins is connected by traces on both sides of the board. Heavy power traces are provided to handle the very large currents involved in a fully-loaded back plane. The two connectors supplied 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.

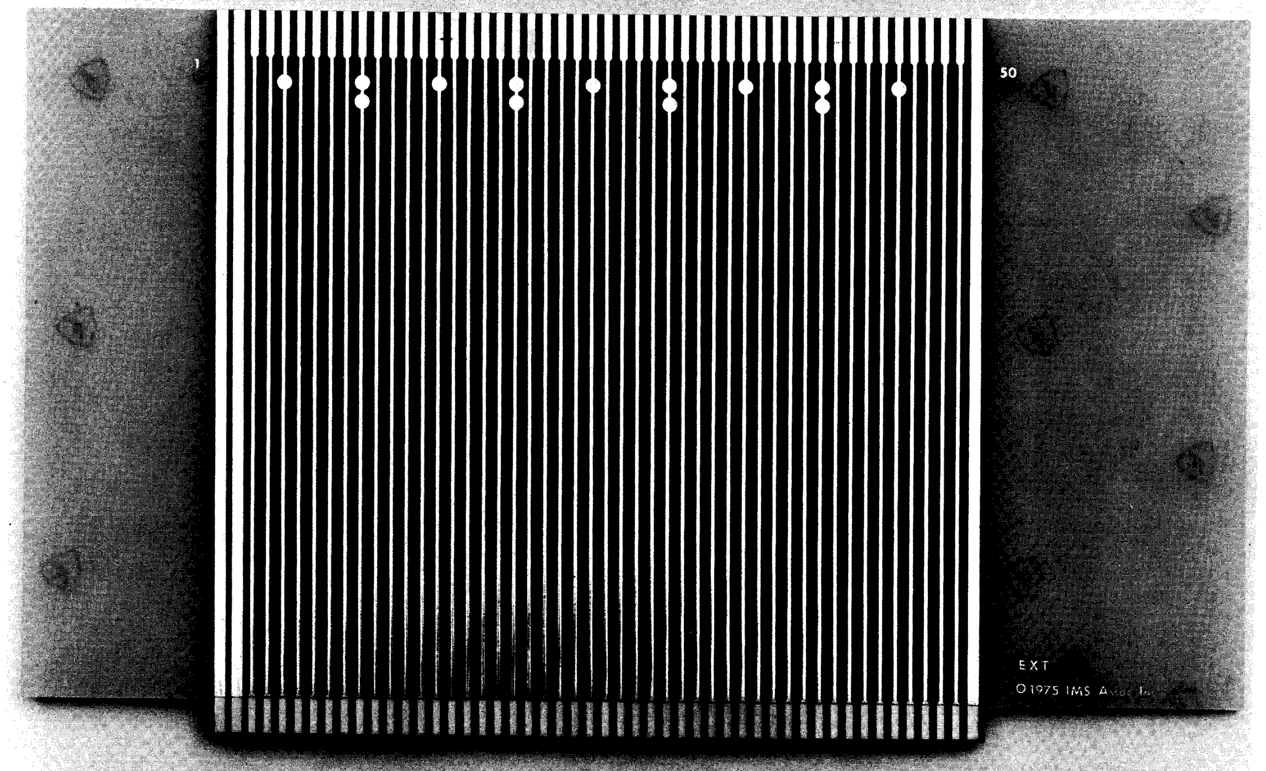

TOP
GND
8 volts
© 1975 IMS Assoc. Inc.
EXP-4 REV-2
+ 16 volts



EXP-6



EXP-22 Rev. 1



EXT (EXTENDER BOARD)

Mother Board
Parts List

EXP-4

	IMSAI		
ITEM	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

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

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	$\overline{\text{VI Ø}}$	54	$\overline{\text{EXT CLR}}$
5	$\overline{\text{VI 1}}$	55	*
6	$\overline{\text{VI 2}}$	56	
7	$\overline{\text{VI 3}}$	57	
8	$\overline{\text{VI 4}}$	58	
9	$\overline{\text{VI 5}}$	59	
10	$\overline{\text{VI 6}}$	60	
11	$\overline{\text{VI 7}}$	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	Ø2	74	$\overline{\text{PHOLD}}$
25	Ø1	75	$\overline{\text{PRESET}}$
26	PHLDA	76	PSYNC
27	PWAIT	77	$\overline{\text{PWR}}$
28	PINTE	78	PDBIN
29	A 5	79	A Ø
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 Ø	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 Ø
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

IMSAI 8080 SYSTEM BUS STRUCTURE

The IMSAI 8080 system bus structure consists of 100 lines. These are arranged 50 on each side of the plug-in boards, with pins 1 through 50 on the component side and pins 51 through 100 on the back side. As the board is viewed right-side up (components up, 100 pin connector towards you) pin #1 is on the left end on the top and pin 51 is on the back side directly opposite pin #1.

Conventions:

SYMBOLS: "P" prefix indicates a processor command or control signal

"S" prefix indicates a processor status signal

LOADING: All inputs to a card should be loaded with a maximum of 1 TTL low power load

LEVELS: All bus signals except the power supply are TTL. All Data and Address lines are positive TRUE (ground = 0 bit)

BUS DEFINITION

Front Side			
No.	SYMBOL	NAME	FUNCTION
1	+8V	+8 volts	Unregulated input to 5v regulators
2	+16V	+16 volts	Positive unregulated voltage
3	XRDY	External Ready	Used by Front Panel: Pulling this line low will cause the processor to enter a WAIT state and allows the status of the normal Ready Line (PRDY) to be examined.
4	$\overline{\text{V10}}$	Vectored Interrupt Line #0	

BUS DEFINITION

Front Side

No.	SYMBOL	NAME	FUNCTION
5	$\overline{\text{VI1}}$	Vectored Interrupt Line # 1	
6	$\overline{\text{VI2}}$	Vectored Interrupt Line #2	
7	$\overline{\text{VI3}}$	Vectored Interrupt Line #3	
8	$\overline{\text{VI4}}$	Vectored Interrupt Line #4	
9	$\overline{\text{VI5}}$	Vectored Interrupt Line #5	
10	$\overline{\text{VI6}}$	Vectored Interrupt Line #6	
11	$\overline{\text{VI7}}$	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 $\overline{\text{PWR}}$ 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	$\overline{\text{POC}}$	Power-On Clear	
100	GND	GROUND	

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.

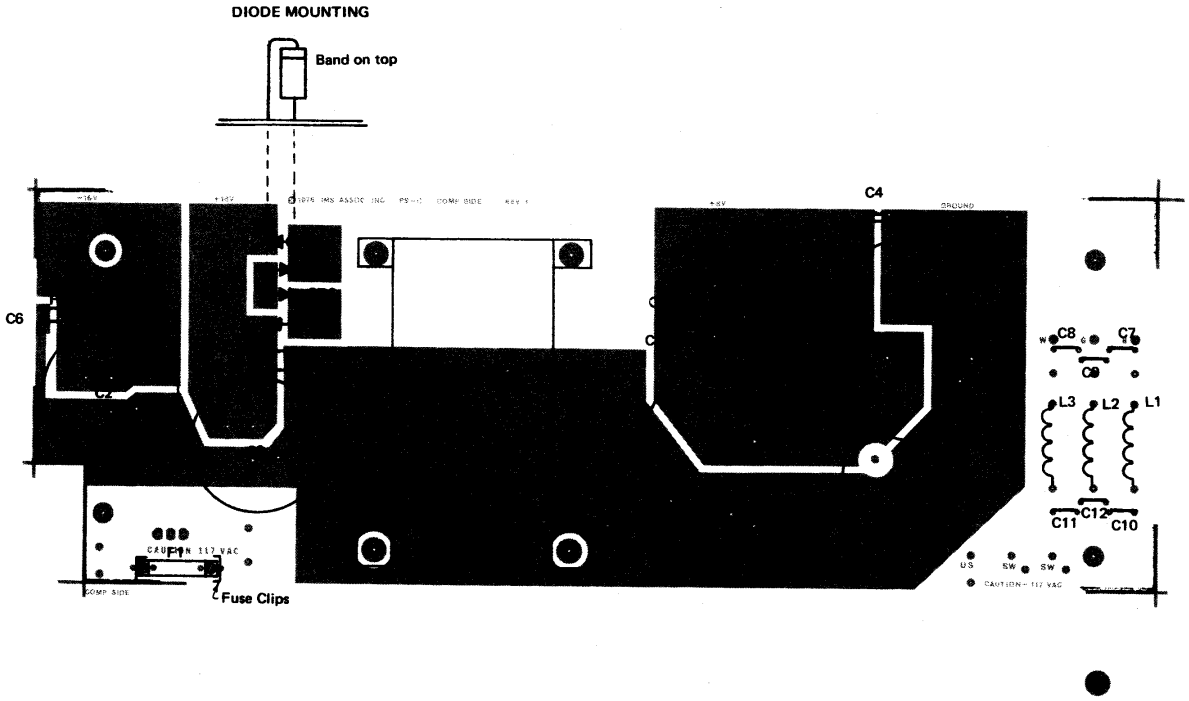
Line Filter: The line filter is a triple PI L-C filter designed to remove high frequency noise present on the AC line. This filter attenuates line noise above 1MHz in frequency.

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.

0	ORIGINAL	5/76
1	MODIFIED	1/77



ASSEMBLY DIAGRAM
PSC-U REV. 1 1/77

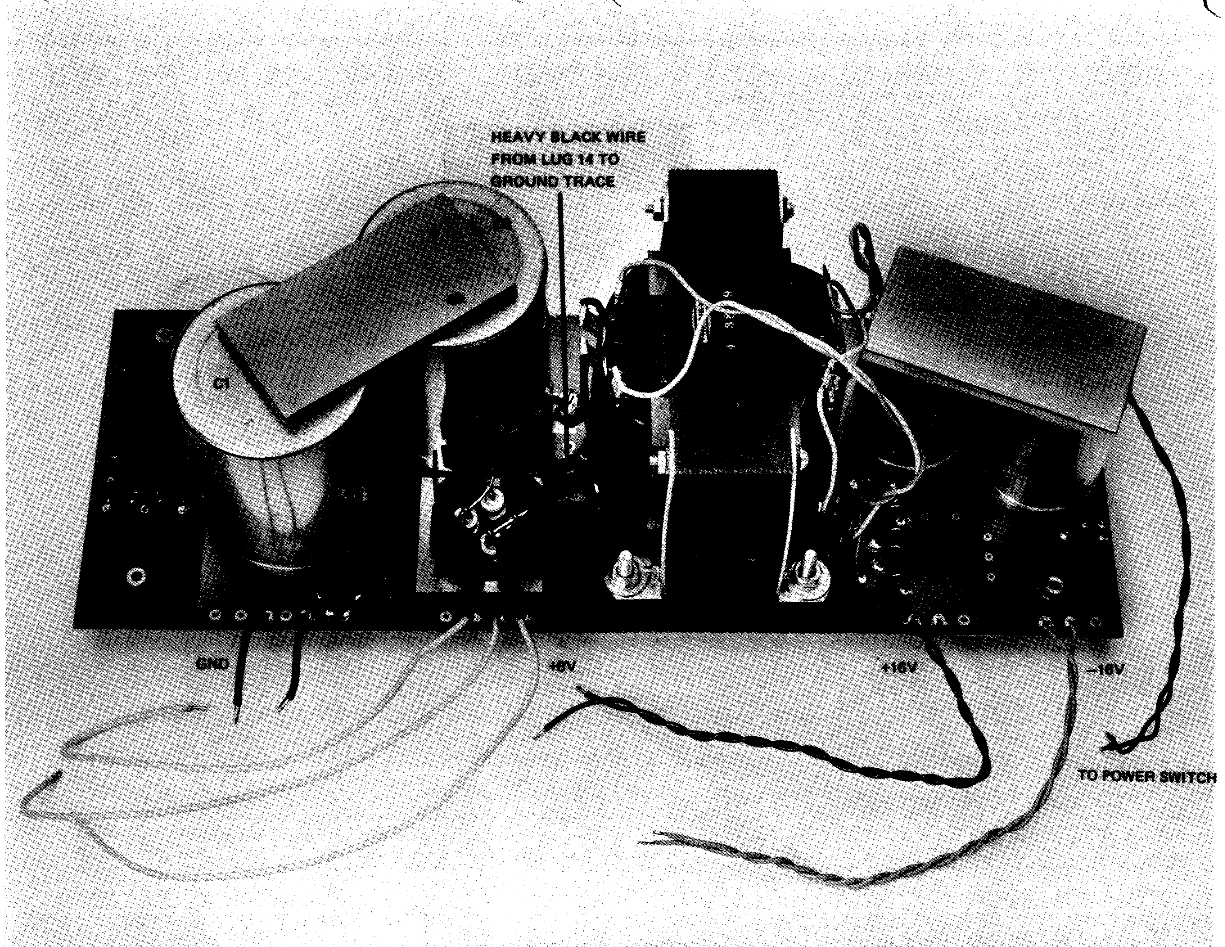


FIG. 1 PS-28U OVERALL

FIG. 2 PRIMARY ING

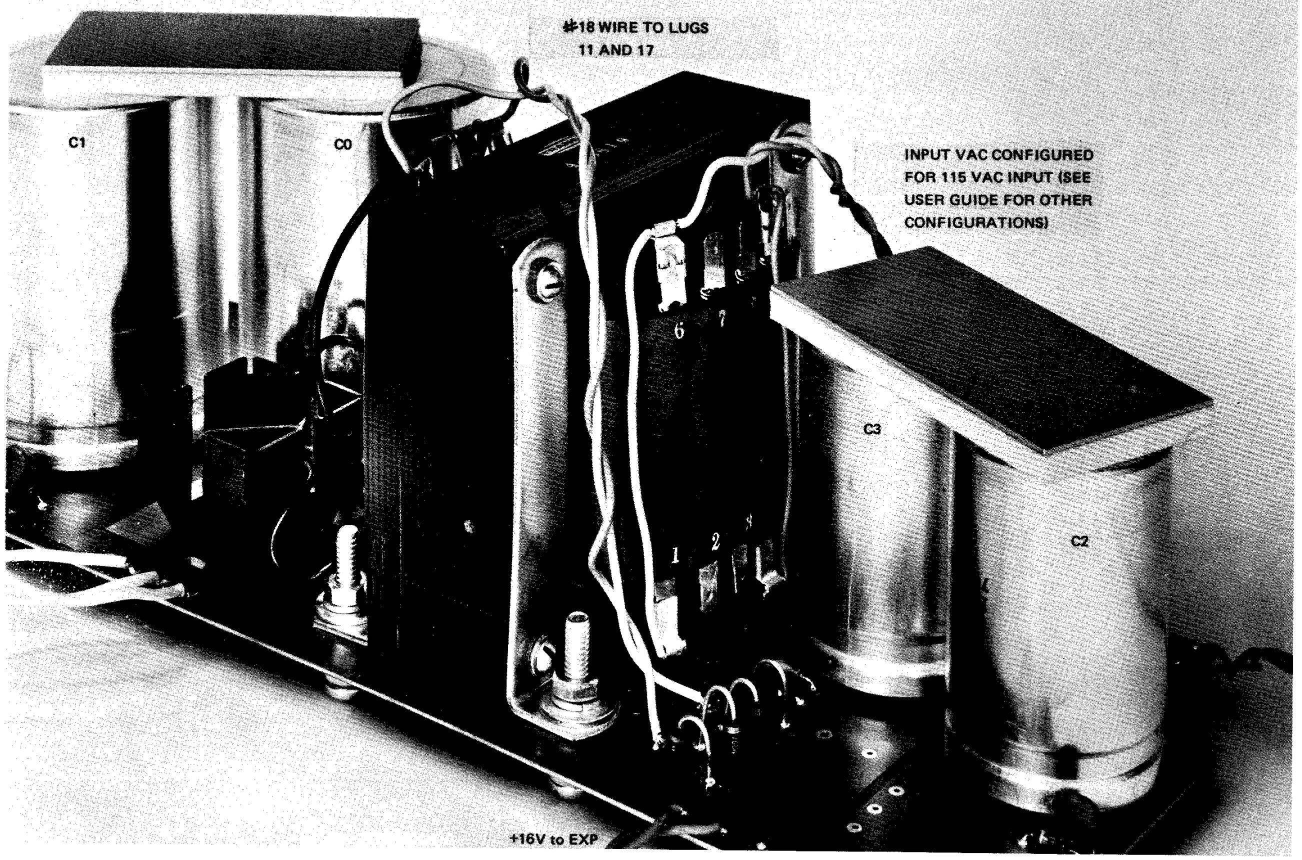
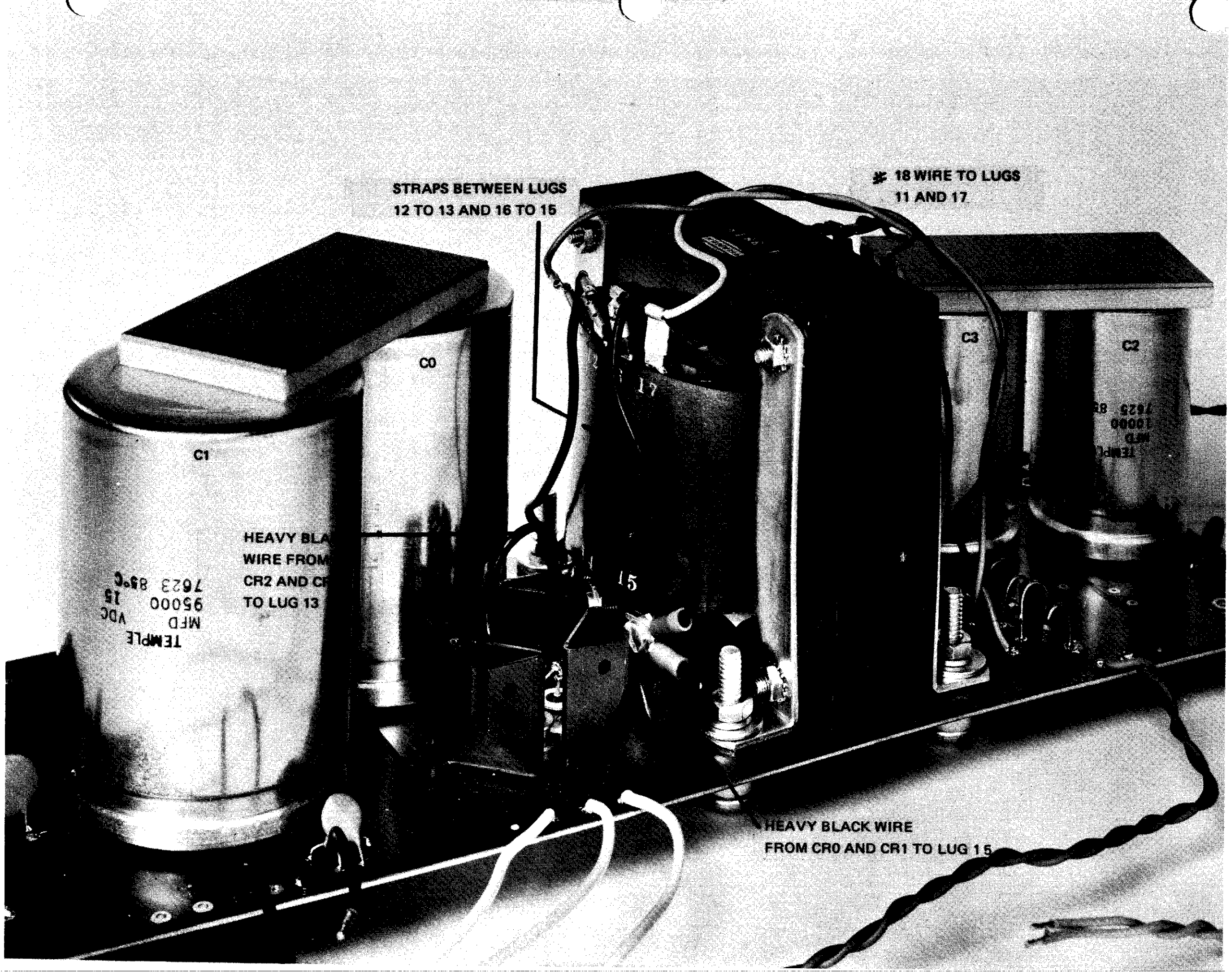


FIG. 3 SECONDARY WIRING



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

PS-28U
Parts List

	IMSAI		
ITEM	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 11 AC Phase 1 to 16V Rect
Pin 15 AC Phase 2 to 8V Rect	Pin 17 AC Phase 2 to 16V Rect
Pin 14 Ground	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.

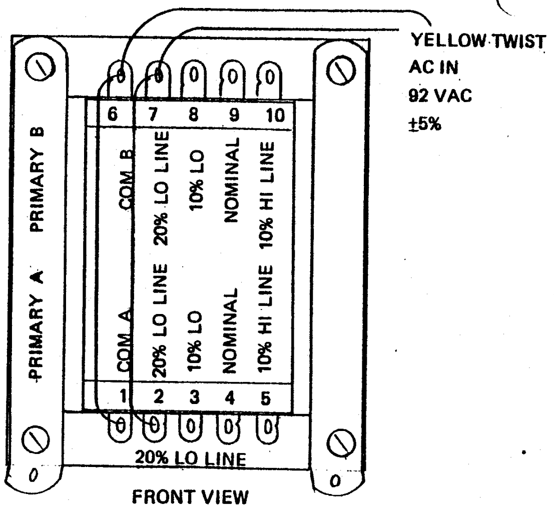
( ) 21. In a similar manner, mount C3 with the negative terminal bolted to the ground plane and positive terminal bolted to the +16 volt plane.

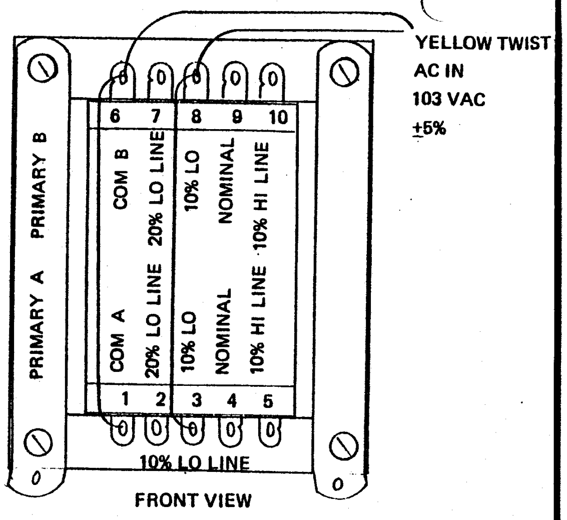
( ) 22. To install capacitor C2, bolt the positive terminal to the DC ground plane and the negative terminal to the negative (-16 volt) plane.

FAN INSTALLATION (OPTIONAL)

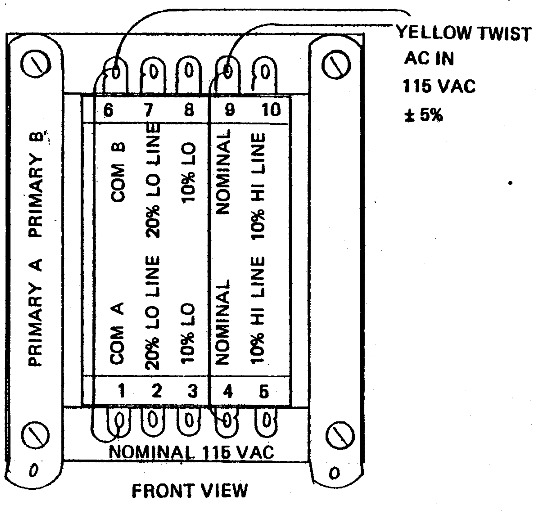
( ) 23. Attach the fan leads to lugs #6 and #9 or #1 and #4 to supply 115 VAC to the fan. This wiring is standard for all input AC wiring configurations.

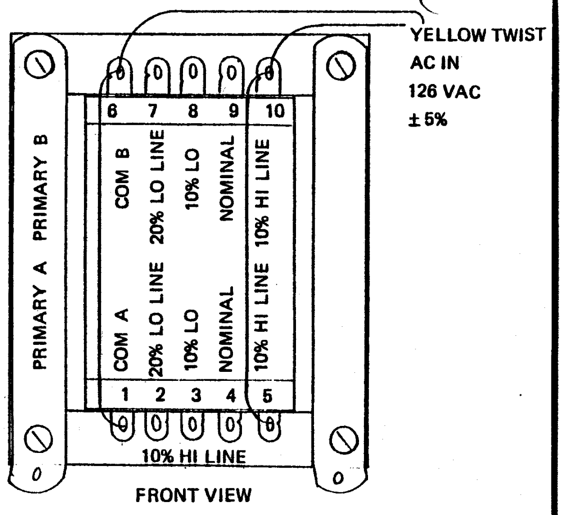
SEE MAINFRAME ASSEMBLY SECTION TO INSTALL POWER SUPPLY IN CHASSIS AND CONNECT TO MOTHER BOARD.



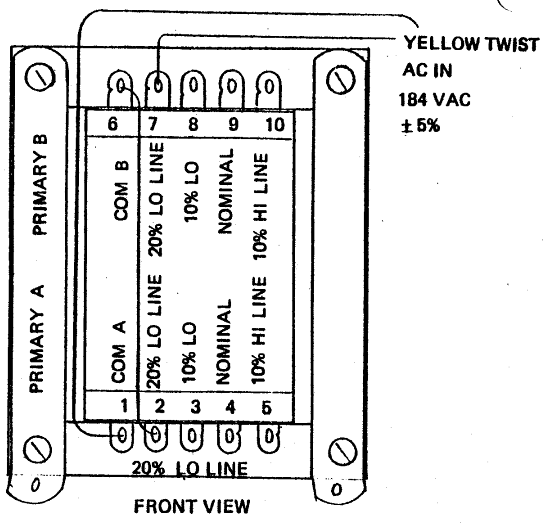


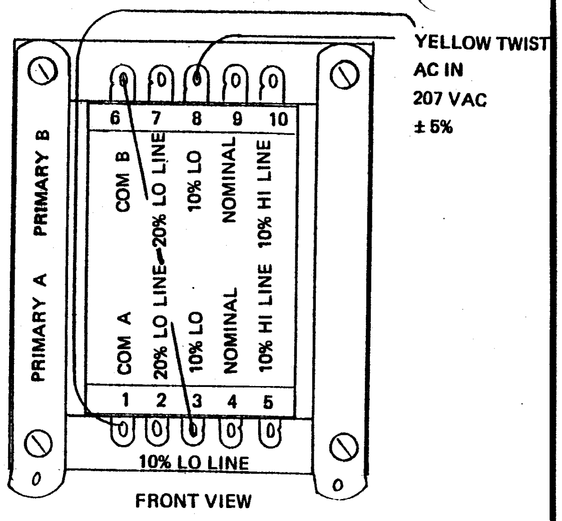
WIRING CHART: 92 – 103 VAC 60 Hz IN
Use next lowest line input taps when operating full chassis or on 50 Hz. See User Guide for more information. Use 5A fuse.



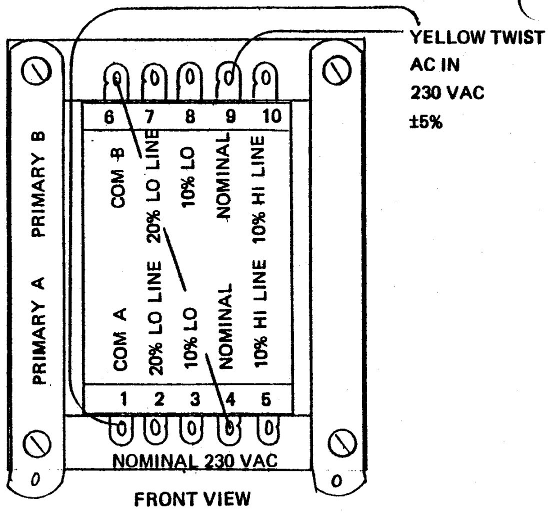


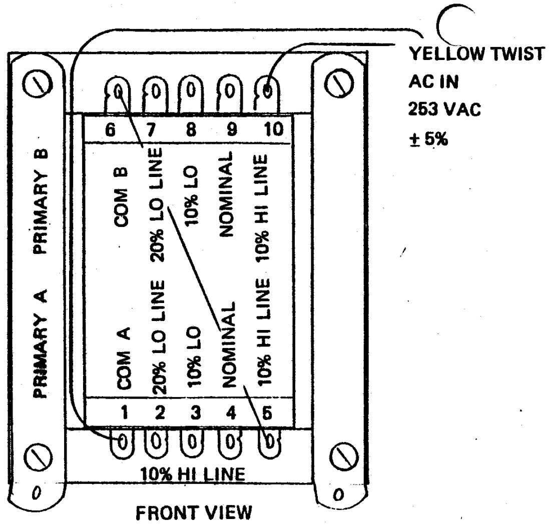
WIRING CHART: 115 – 126 VAC 60 Hz IN
Use next lowest line input taps when operating full chassis or on 50 Hz. See User Guide for more information. Use 5A fuse.



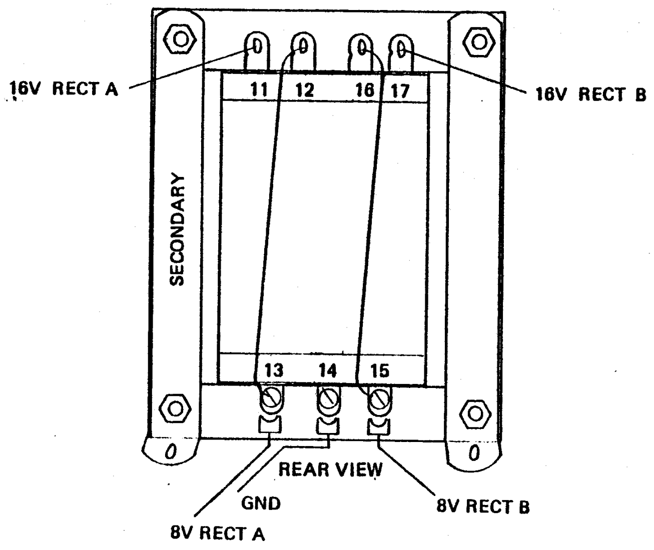


WIRING CHART: 184 – 207 VAC 60 Hz IN
Use next lowest line input taps when operating full chassis or on 50 Hz. See User Guide for more information. Use 2½A fuse.





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

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.

IMPORTANT

When using the RAM-16, 32 or 65 boards in systems containing an IMSAI CPA board of Rev. 4 or earlier, make the changes described ECN 77-0039 which may be found in the Appendix of this chapter.

CP-A, Rev. 4
Errata
ECO 77-0098
9/1/77

ERRATA

Reason for Change:

These modifications should be made to prevent the spurious triggering of one-shots on the CP-A while in RUN mode, causing unpredictable program execution.

Nature of Change:

This modification will disable one-shots (3 74123 Dual Monostable Multivibrators) during RUN.

Instructions:

On the component side:

1) Cut the trace between U23, Pin 11 and R60.

On the solder side:

2) Cut the trace between U17, Pin 2 and feed through.

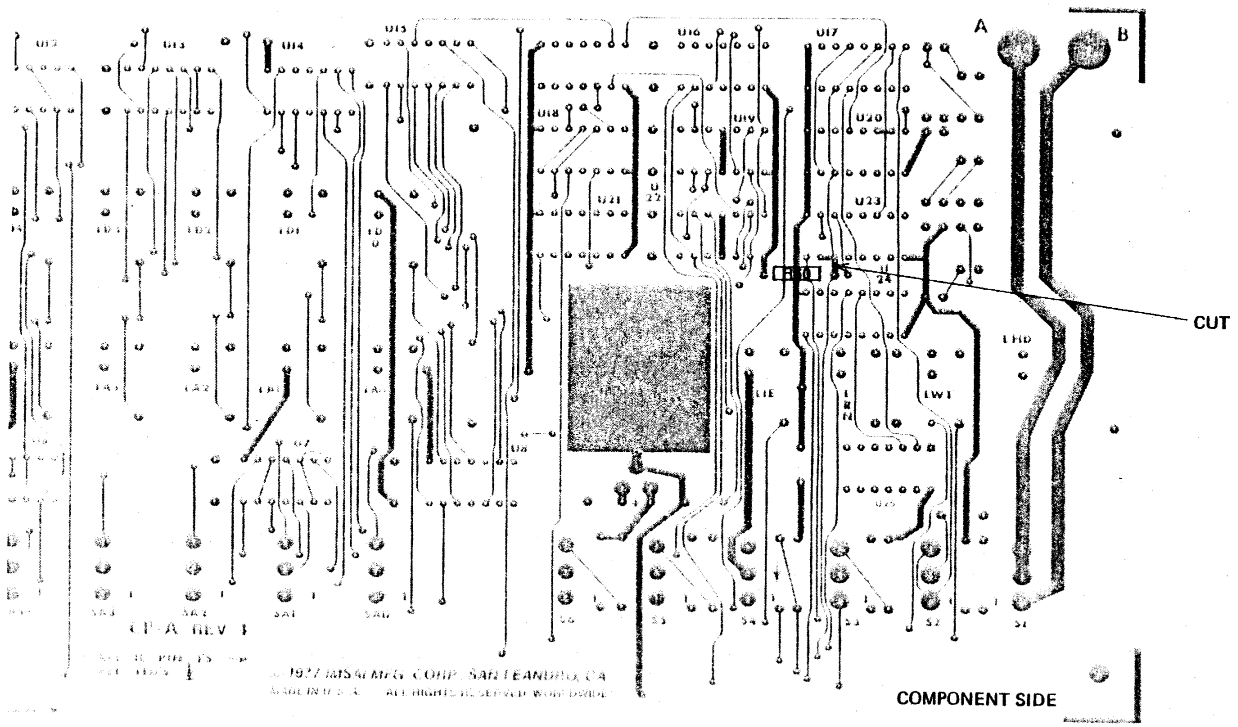
3) Cut the trace between U20, Pin 11 and U19, Pin 8.

4) Connect U20, Pin 11 to U22, Pin 6.

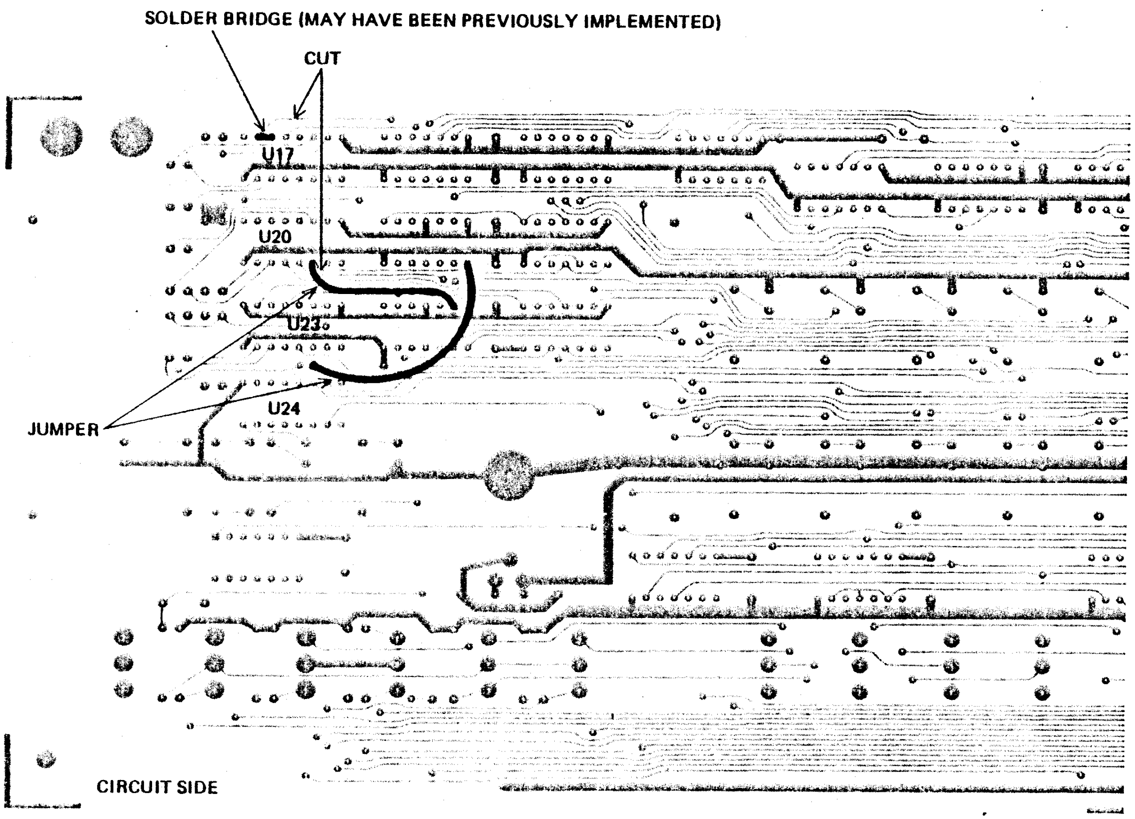
5) Connect U19, Pin 8 to load side of R60.

6) Connect U17, Pin 2 to U17, Pin 3.

Refer to the following diagrams for clarification.



IMSAI SYSTEM
CP–A REV. 4
COMPONENT SIDE
ECO 77–0098



IMSAI SYSTEM
CP–A REV. 4
CIRCUIT SIDE
ECO 77--0098

March 5,1977

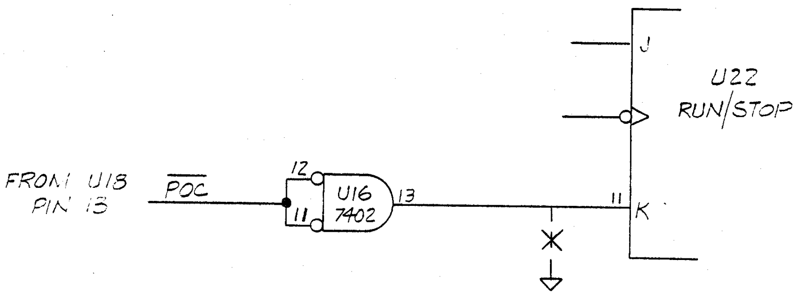
CPA REV 4 MODIFICATION

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.



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)

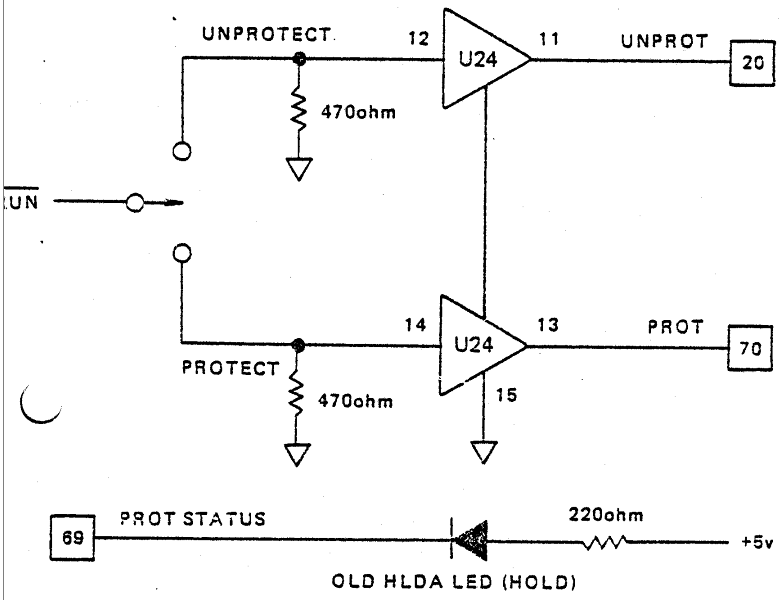
From U24 pin 11 to the pad connected to pin 20

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

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

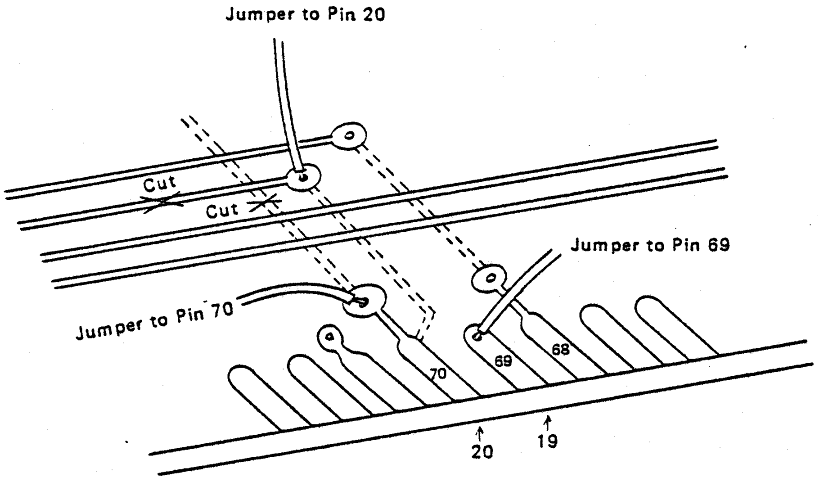
Re-assemble CP-A to chassis

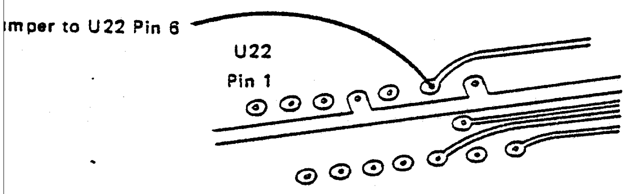
The right-hand switch now serves to change the protect status of the currently addressed block of memory when the machine is not in run mode. The LED which previously showed HOLD status now is lit when the currently addressed block of memory is protected.


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

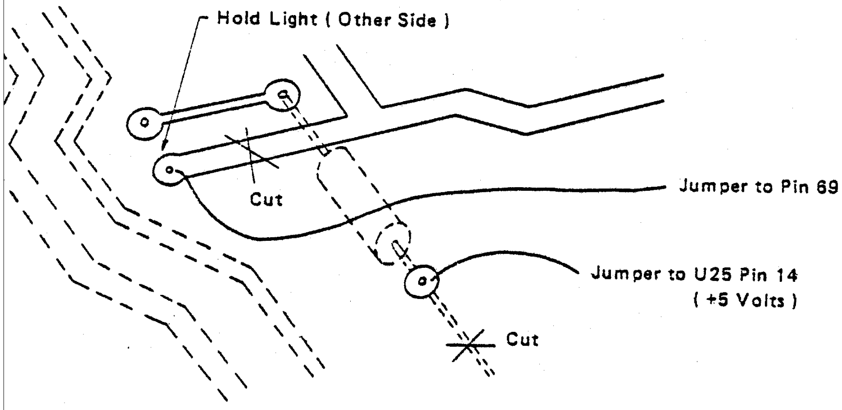
SOLDER SIDE

(51—•)

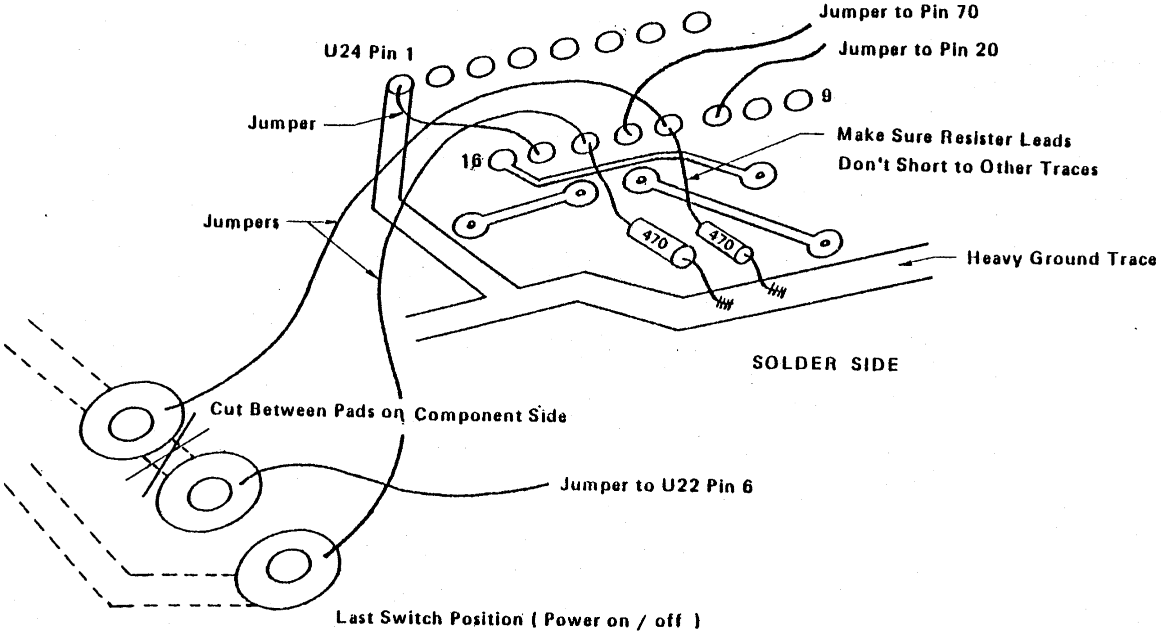


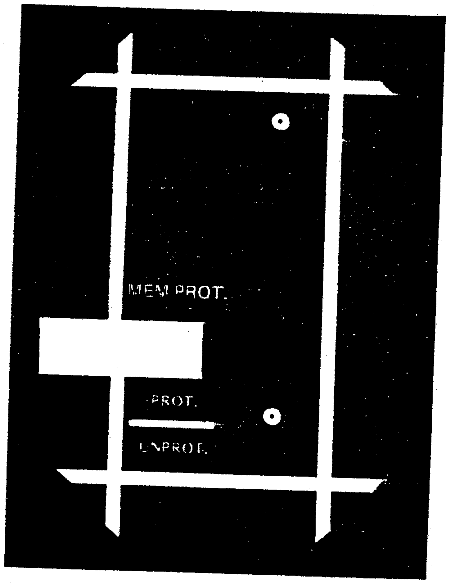


SOLDER SIDE



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

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.

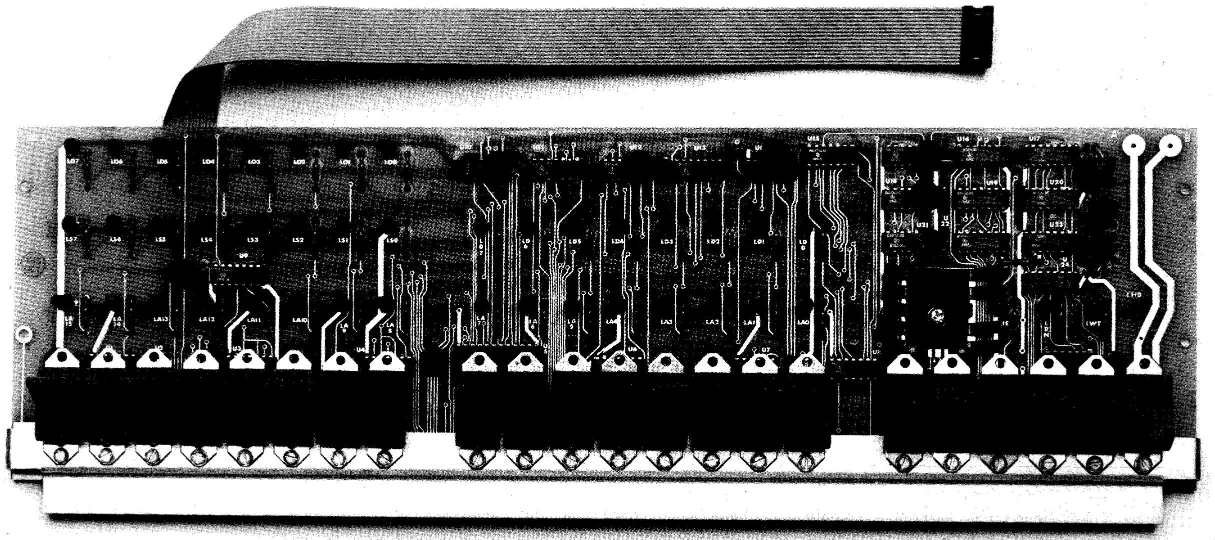
The same address decode signal is ANDed with the STATUS OUT line to enable the 8212 8 bit latch which drives the PROGRAMMED OUTPUT indicators. The information on the bi-directional data bus is then latched onto the output port at the time of the processor write strobe.

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.

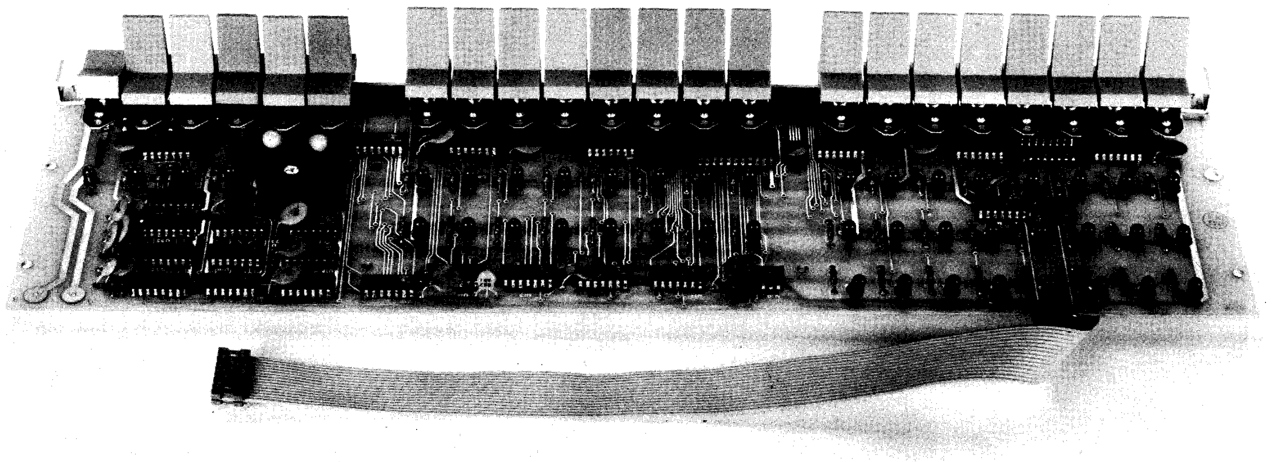
The STATUS WORD DISABLE line, Pin 53 in the backplane, is also run by the signal which puts the high order address switches onto the data bus for the port FF read instruction so that the bi-directional data bus is not being driven by the bi-directional drivers on the MPU board at the same time that the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

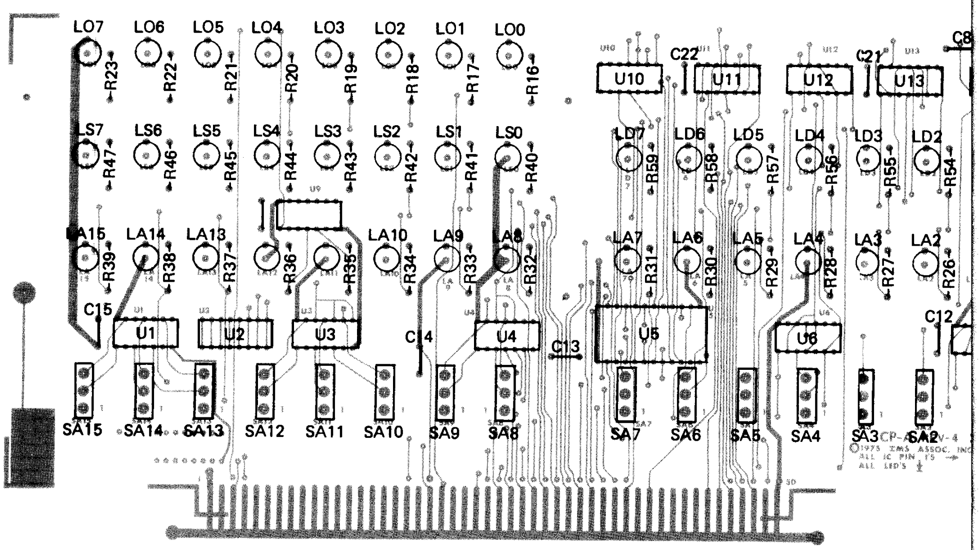
When the RESET switch is thrown to EXTERNAL CLEAR, the switch directly grounds the EXTERNAL SWITCH line on the backplane. There is a diode between the RESET line and the EXTERNAL CLEAR line so that during a reset operation an EXTERNAL CLEAR is also generated.



CP-A REV. 4

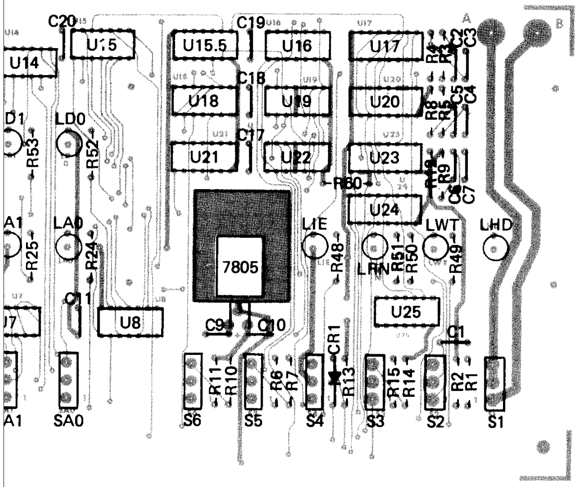


CP-A REV. 4


LO7
LO6
LO5
LO4
LO3
LO2
LO1
LO0
R23
R22
R21
R20
R19
R18
R17
R16
U10
C22
U11
U12
C21
U13
C8
LS7
LS6
LS5
LS4
LS3
LS2
LS1
LS0
R47
R46
R45
R44
R43
R42
R41
R40
LD7
LD6
LD5
LD4
LD3
LD2
R59
R58
R57
R56
R55
R54
LA15
LA14
LA13
LA10
LA9
LA8
LA7
LA6
LA5
LA4
LA3
LA2
R39
R38
R37
R36
R35
R34
R33
R32
R31
R30
R29
R28
R27
R26
C15
U1
U2
U3
C14
U4
C13
U5
U6
C12
SA15
SA14
SA13
SA12
SA11
SA10
SA9
SA8
SA7
SA6
SA5
SA4
SA3
SA2
CP-A REV-4
©1975 IMS ASSOC. INC
ALL IC PIN 15 →
ALL LED'S ⏚

REVISIONS

LTR	DESCRIPTION	DATE	APPROVED
0	ORIGINAL	12/75	
1	ECN76–0051 & ECO77–0039	11/76 & 4/77	
2	ECO77–0098	8/77	



U1 U3 U4 U6 U7	7405	LO0 THRU LO7 LA0 THRU LA15 LD0 THRU LD7 LS0 THRU LS7 LIE LHD LRN LWT	LED'S
U2	DATA HEADER SOCKET		
U5	8212		
U8 U15	74LS04		
U9	74LS30		
U10	7404		
U11	74LS10		
U12	7410		
U13 U16	7402	R1	270K ¼W
U14 U25	7400		
U15.5	7427		
U17 U20 U23	74123	R2 R6 R7 R10 R11 R13 THRU R15 R60	1K ¼W
U18 U19 U22	74107		
U21	7430	R3 R4 R5 R8 R9 R12	47K ¼W
U24	8T97		
C1 C2 C5 C6 C7 C11 THRU C22	.1 uF	R16 THRU R59	22 ¼W
C3 C8	.001 uF		
C4	.01 uF		
C9 C10	33 uF		
CR1	1N914		

TOLERANCES UNLESS OTHERWISE SPECIFIED FRACTIONS ± DEC. ± ANGLES ±		© 1977 IMSAI MFG. CORP., SAN LEANDRO, CA. ALL RIGHTS RESERVED WORLDWIDE MADE IN U.S.A.	
APPROVALS	DATE	IMSAI 8080 SYSTEM CPA REV. 4 ASSEMBLY DIAGRAM	
DRAWN			
CHECKED PRU	8/24/77	SCALE — SIZE B	DRAWING NO.
		DO NOT SCALE DRAWING	SHEET

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

CP-A, Rev. 4
Parts List

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/yellow) 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 .luf 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.

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.

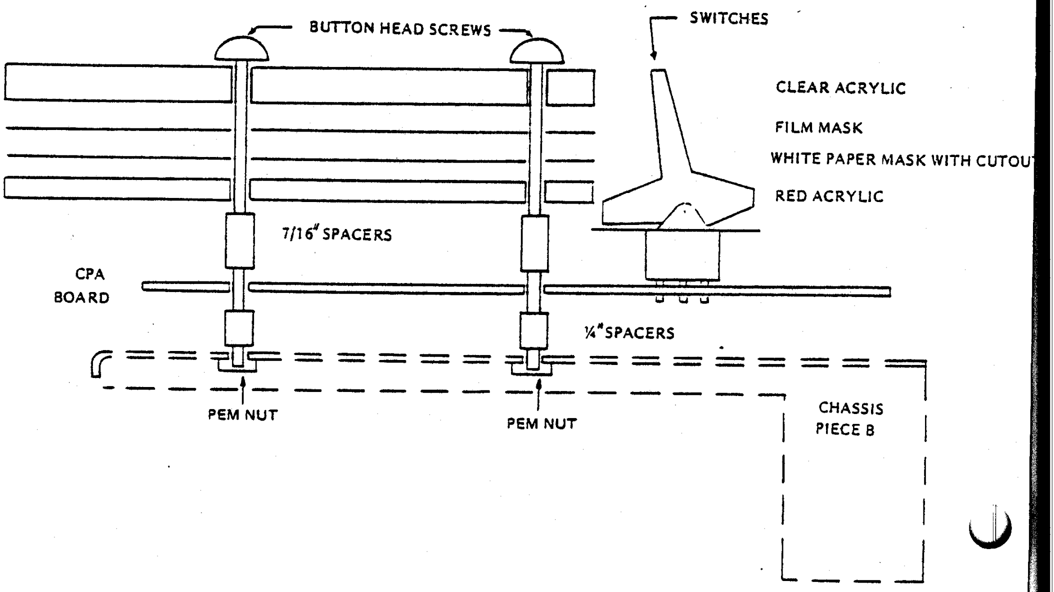
Eight ¼ inch spacers can then be slipped over the screws behind the CP-A board and eight #6 nuts and lockwashers can be put on to hold the sandwich together.

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.

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.



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 bv 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 data 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.

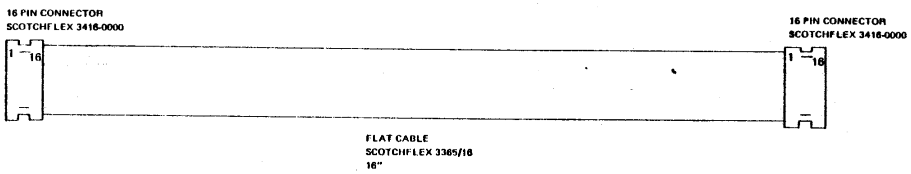
The 8-BIT PROGRAMMED OUTPUT INDICATOR lights can be controlled by the program through the use of the output instruction to port location hex FF or octal 377. When 0 bits are output into this port, the indicator lights will be turned on and when 1 bits are output into this port, indicator lights will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

REVISIONS			
LTR	DESCRIPTION	DATE	APPROVED
0	ORIGINAL	8/76	



© 1976
IMS ASSOCIATES INC.
SAN LEANDRO, CA.

TOLERANCES UNLESS OTHERWISE SPECIFIED FRACTIONS ± DEC. ± ANGLES ±		© 1977 IMSAI MFG. CORP. SAN LEANDRO, CA. ALL RIGHTS RESERVED WORLDWIDE MADE IN U.S.A.
APPROVALS	DATE	IMSAI 8080 SYSTEM CPA CABLE K ASSEMBLY 91-0400001
DRAWN YP	8/76	
CHECKED A.I.H.	8/11/76	SCALE ——— SIZE B DRAWING NO. 91-2040001
		DO NOT SCALE DRAWING SHEET 1 OF 1

APPENDIX

CP-A Modifications for Dynamic RAM

ERRATA FOR CPA REV. 4 AND EARLIER
ECN 77-0039

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 Figure 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 Figure 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 Figure 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 Figure 3.

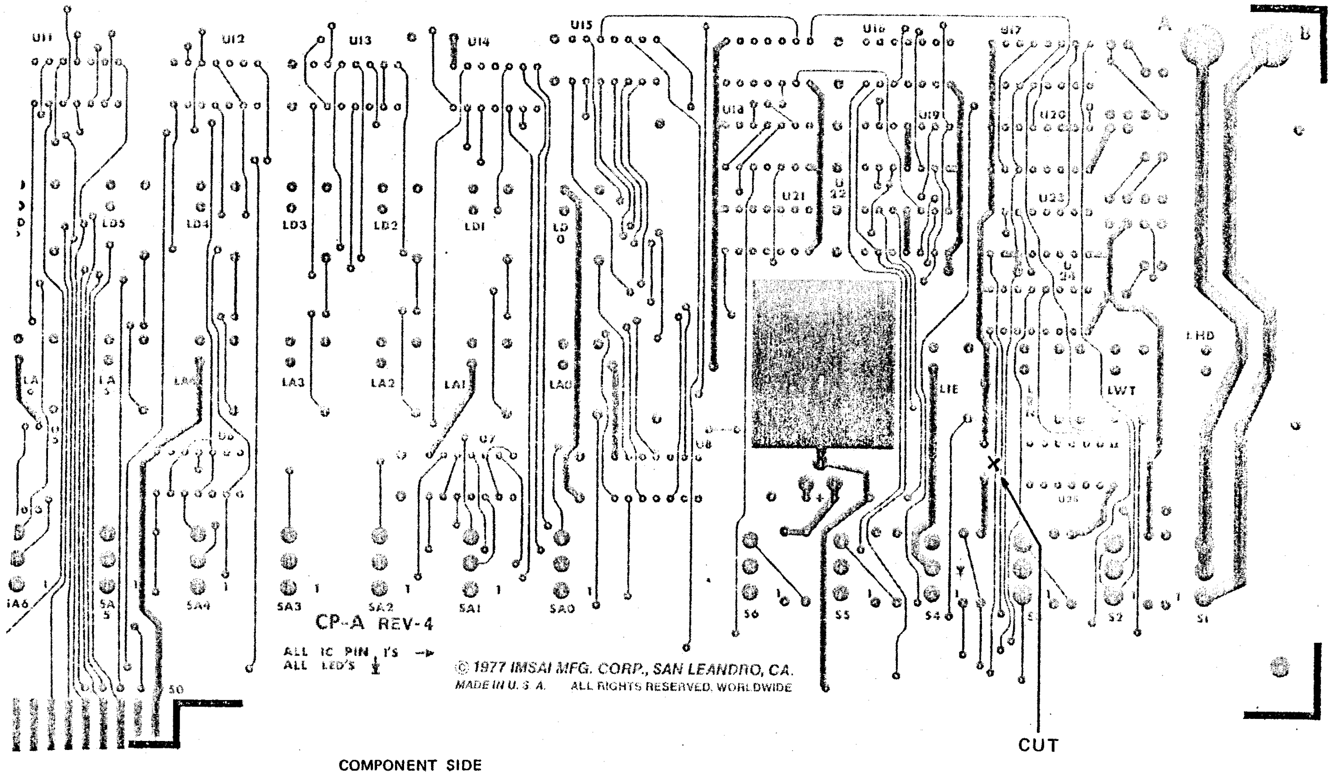


FIGURE 1
MODIFICATION TO CPA REV. 4 & EARLIER
FOR DYNAMIC RAM OPERATION

ECN - 77 - 0039

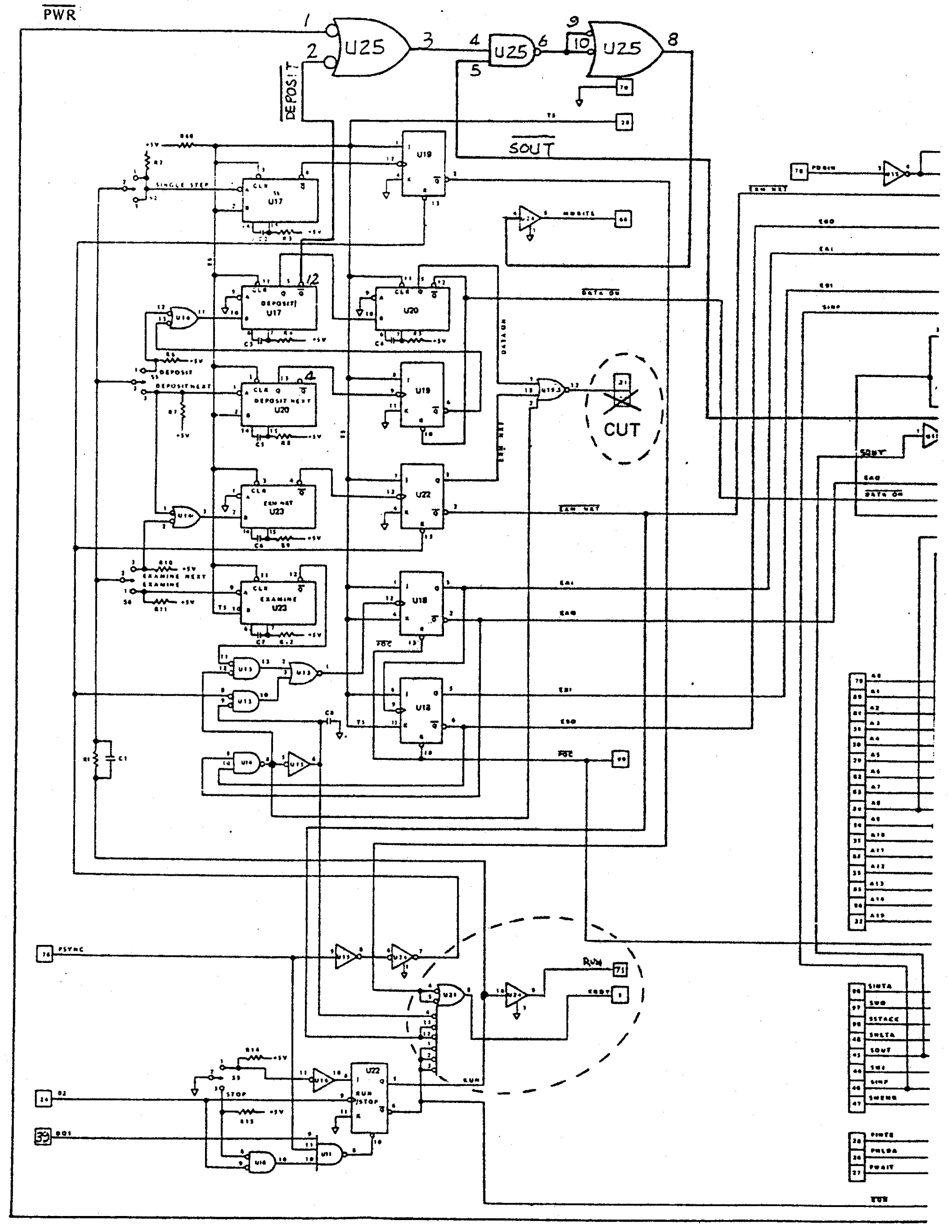
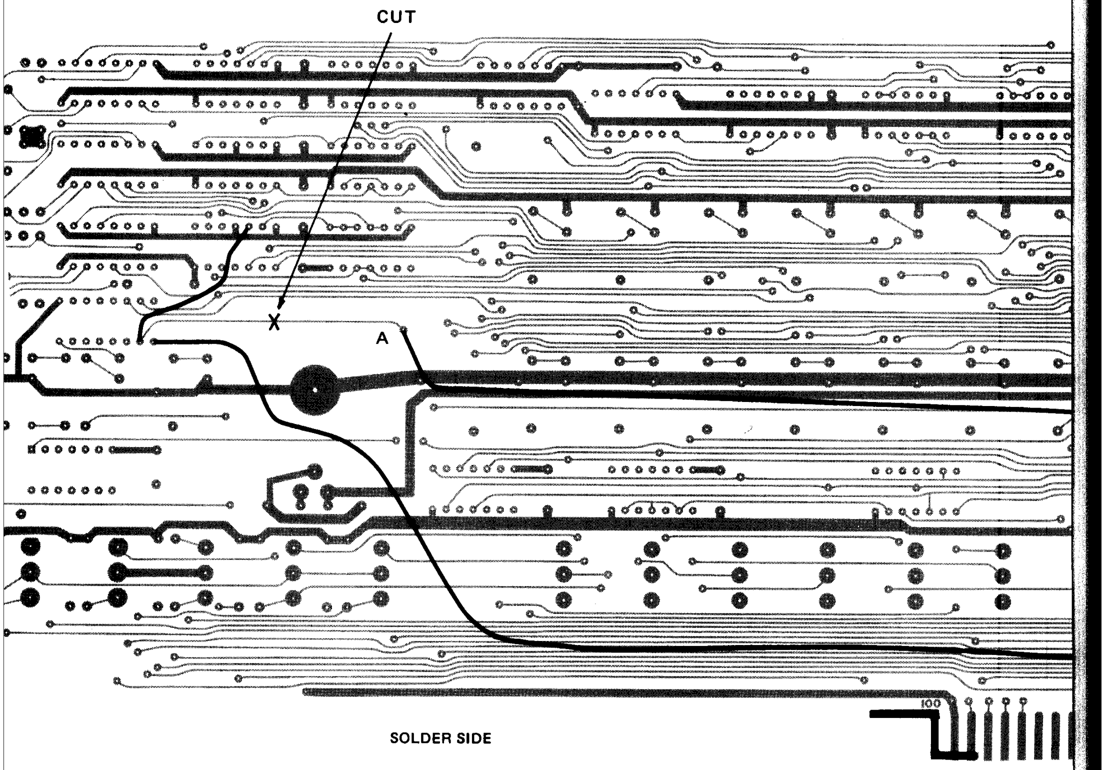


FIGURE 3

MODIFICATION TO CPA REV. 4 & EARLIER

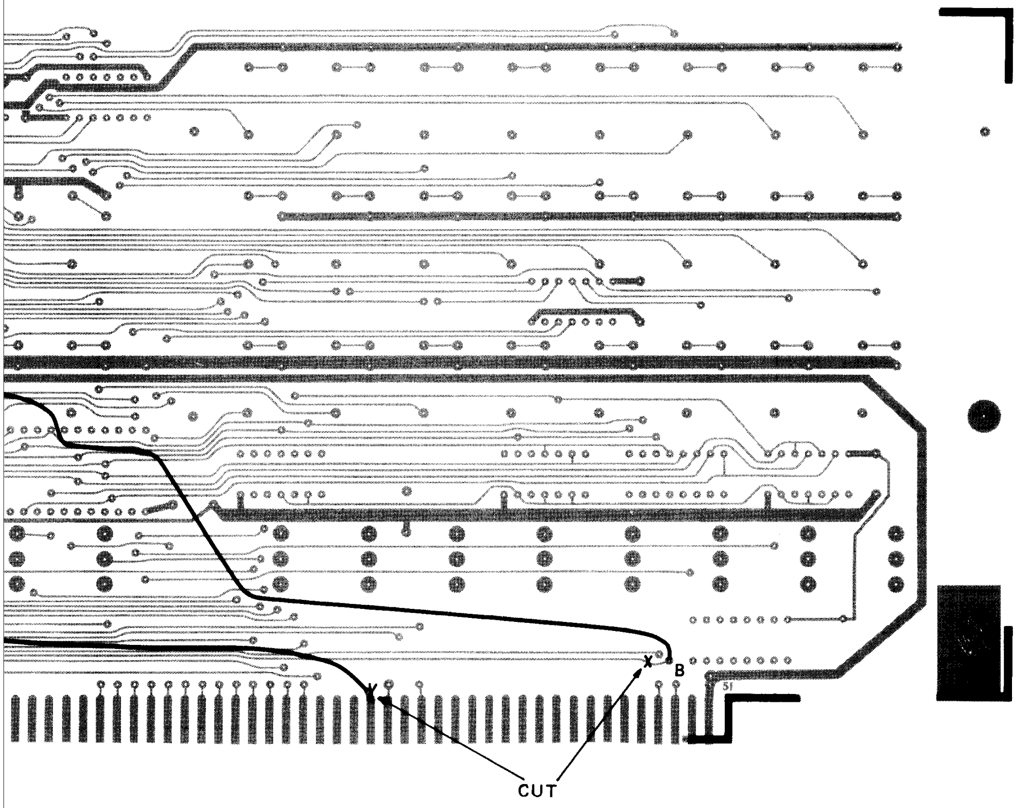
ECN 77–0039



SOLDER SIDE

REVISIONS

LTR	DESCRIPTION	DATE	APPROVED
0	ECN 77-0039	4/25/77	PRU



TOLERANCES UNLESS OTHERWISE SPECIFIED FRACTIONS ± DEC. ± ANGLES ±		IMSAI MFG. CORP. SAN LEANDRO, CA.
APPROVALS	DATE	FIGURE 2 MODIFICATION TO CPA REV. 4 & EARLIER FOR DYNAMIC RAM OPERATION ECN - 77 - 0039
DRAWN	4/22/77	
CHECKED		SCALE — SIZE B DRAWING NO.
		DO NOT SCALE DRAWING SHEET —

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.

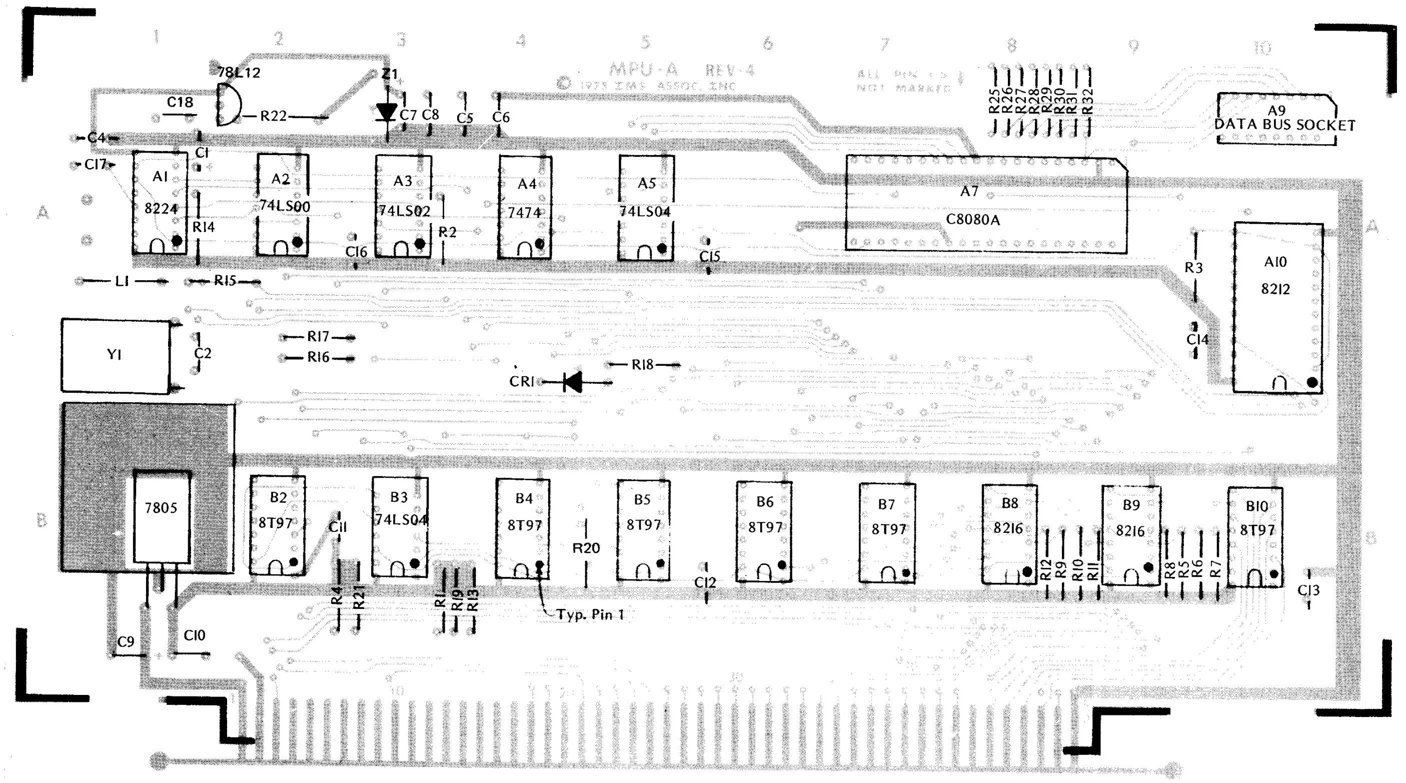
The 8224 clock chip and crystal oscillator provide the two-phase non-overlapping 2 megacycle system clock for the 8080A. These clocks are also driven onto the back plane through 8T97 tri-state buffered drivers.

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.

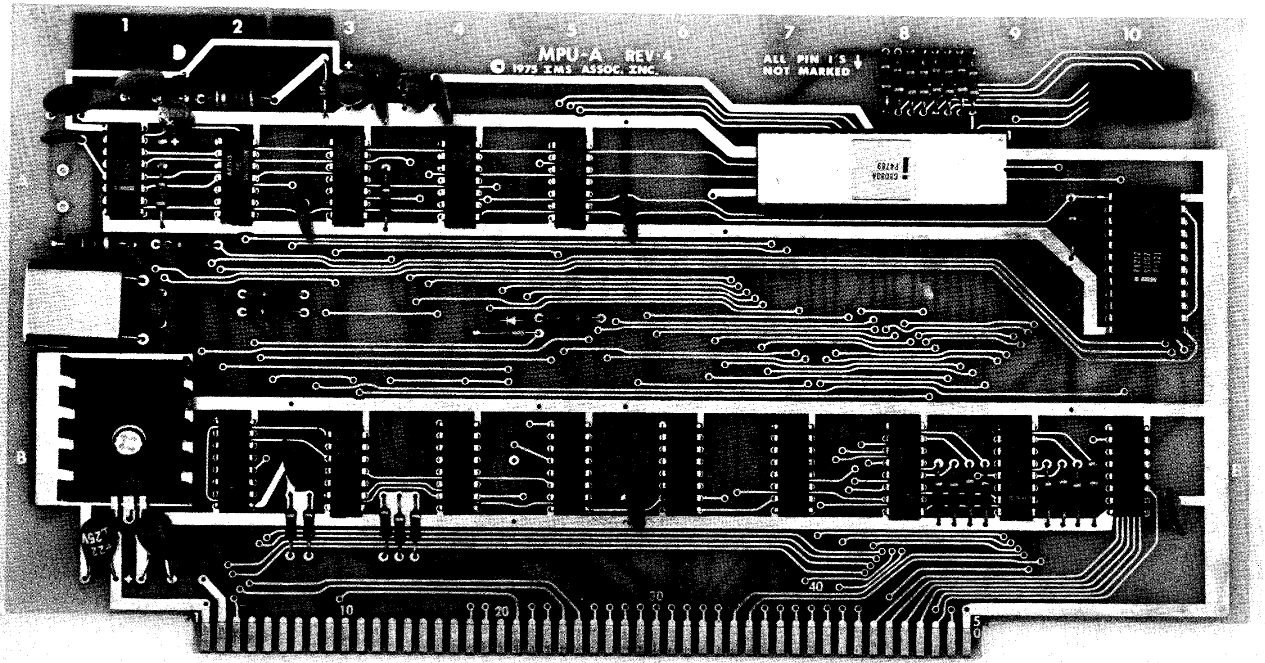
The six processor status signals (SYNC WRITE, STROBE DATA BIT IN, INTERRUPT ENABLED, HOLD ACKNOWLEDGED, and WAIT ACKNOWLEDGE) are all driven onto the back plane through 8T97 tri-state buffered drivers. These drivers may be disabled by the CONTROL DISABLE line, pin 19 on the back plane.

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.



IMS Associates Inc.
ASSEMBLY DIAGRAM
MPU-A REV 4 2/76

5/18/76


1
2
3
4
5
6
7
8
9
10
MPU-A REV 4
© 1975 IMS ASSOC. INC.
ALL PIN 1'S ↓
NOT MARKED
A
A
B
B
10
20
30
40
50

MPU-A REV 4

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION
Solder	15-0000001	5'	
Heat Sink	16-0100002	1	Thermalloy Heat Sink, 6106B-14
Screw	20-3302001	1	6-32x5/16" Phillips Pan Head Machine
Nut	21-3120001	1	6-32 Hex, CAD
Lockwasher	21-3350001	1	#6 Internal Star Lockwasher
Socket	23-0800001	1	16 Pin, Solder Tail, IC Socket
Socket	23-0800004	1	40 Pin, Solder Tail, IC Socket
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)
Capacitor	32-0239010	1	39pF Disk Ceramic
Capacitor	32-2010010	9	.1uF Disk Ceramic
Capacitor	32-2233070	5	33uF, 25V Tantalum
Diode	35-1000005	1	1N751A Diode
Diode	35-1000006	1	1N914 Silicon
Crystal	35-5000003	1	18.00 MHz, Series Resonant, HC-18 Case, Cut AT, Fundamental Mode, .02% Tolerance
8T97	36-0089701	6	Hex Tri-State Driver
74LS00	36-0740002	1	Quad 2 Input NAND (Low Power Schottky)
74LS02	36-0740202	1	Quad 2 Input NOR (LPS)
74LS04	36-0740402	2	Hex Inverter (LPS)
7474	36-0747401	1	Dual D Flip Flop
7805	36-0780501	1	5V Positive Voltage Regulator

ITEM	IMSAI PART #	QUANTITY	DESCRIPTION
78L12	36-0781202	1	12V Regulator
8080A	36-0808001	1	Microprocessor
8212	36-0821201	1	Input/Output Port
8216	36-0821601	2	Bi-Directional Bus Driver
8224	36-0822401	1	Clock Generator and Driver
Chapter	81-0000031	1	MPU-A (for separate orders only)
PC Board	92-0000011	1	MPU-A Rev. 4

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 the 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 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 39 pF disk capacitor (C2) as shown on the Assembly Diagram.

20) Insert and solder each of the nine .1 uF disk capacitors at locations C6, C8, C11 through C16, and C18.

21) Insert and solder each of the five 33 uF tantalum capacitors at locations 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

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

23) 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.

24) 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

25) If possible before plugging in the 8080A chip, the board should be inserted in a chassis, the power turned on, and 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!)

Finally, insert the 8080A microprocessor chip in the 40 pin IC socket 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 Intel 8080 Micro Computer Systems User's Manual.

Errata
2/4/77

RAM 4A-4

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

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.

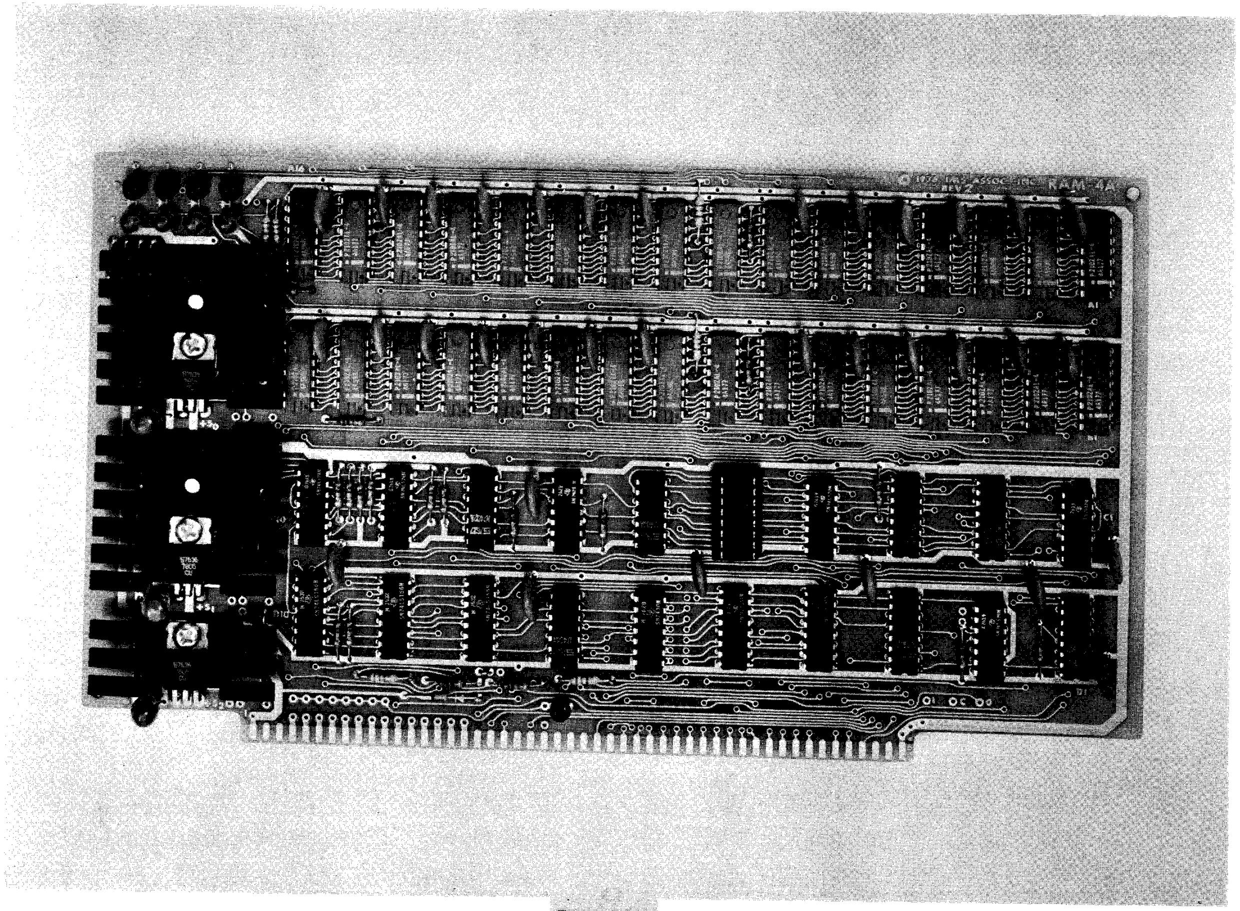
If a 1024 word block of memory is selected by its chip enable being decoded by the 74LS156 at location D8, and its respective write protect flip/flop at locations C10 or C9 are not in the protect state, then the section or the 74LS02 at location C8 associated with this block will have a high output. 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 All 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 All and A10.

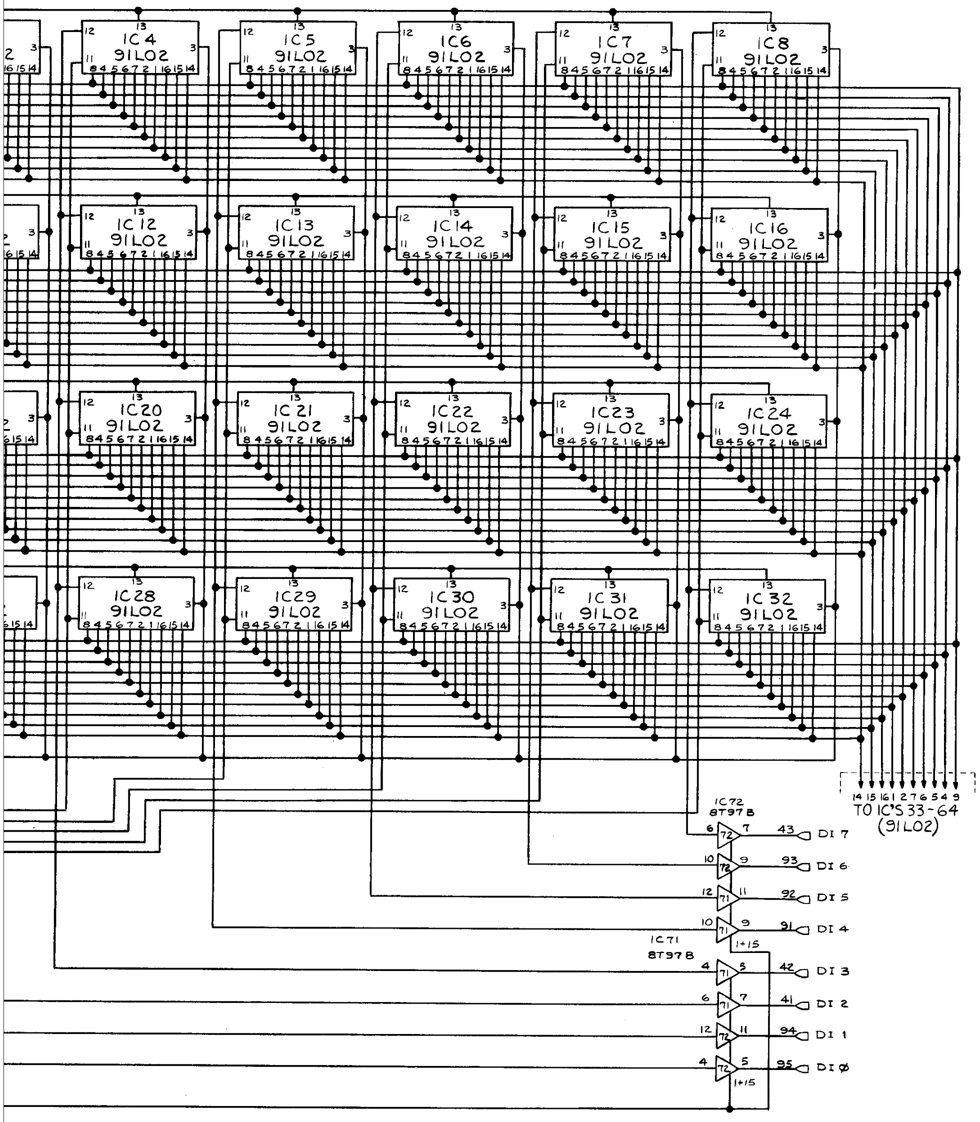
The other set of decoders are both sections of the 74LS156 at location D9. These are utilized when the four write protect flip/flops are going to have their status changed by programmed commands in the software. The command used is an output command, one of 256 available. The board is created to use 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

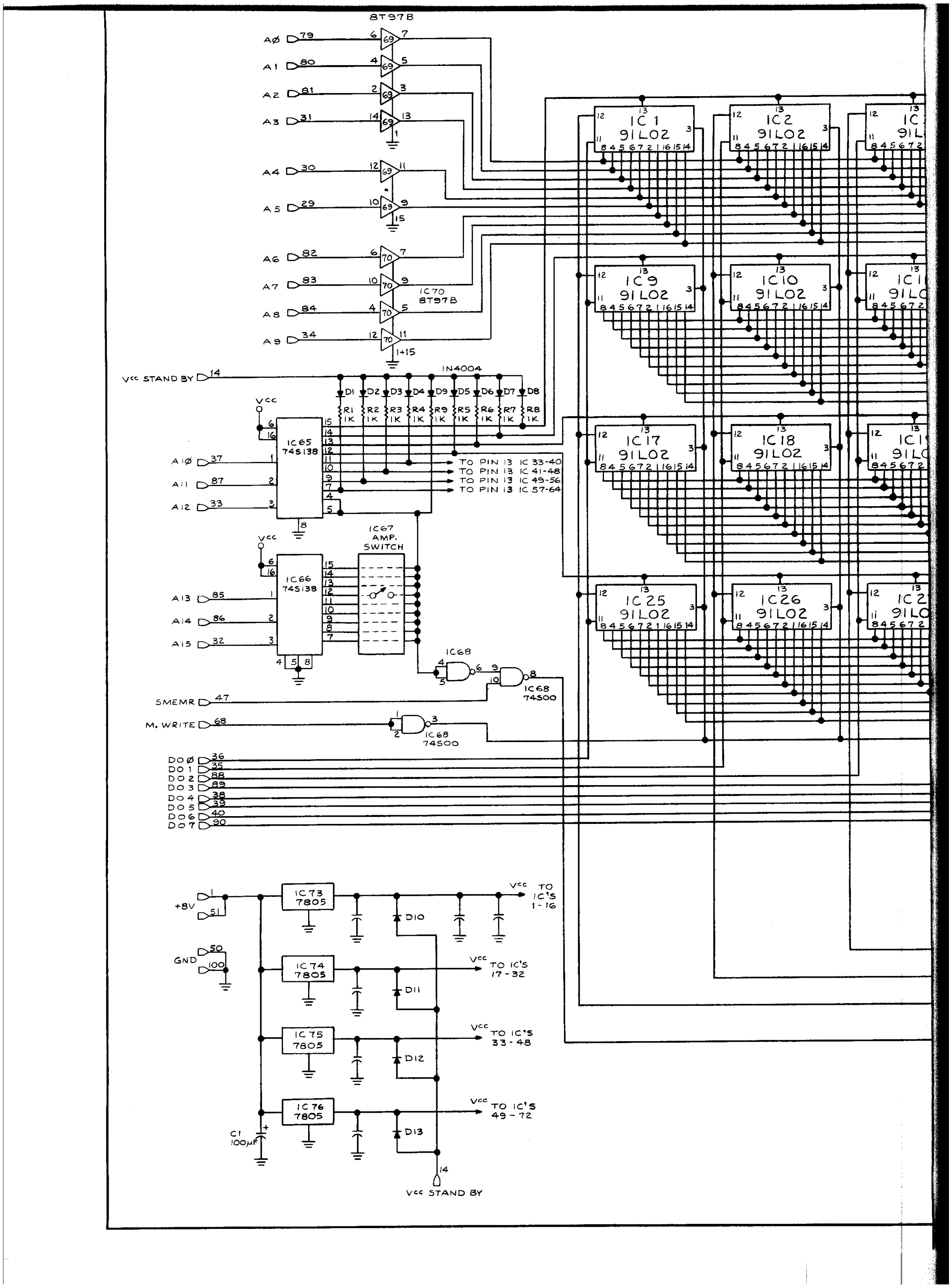


SEALS ELECTRONICS

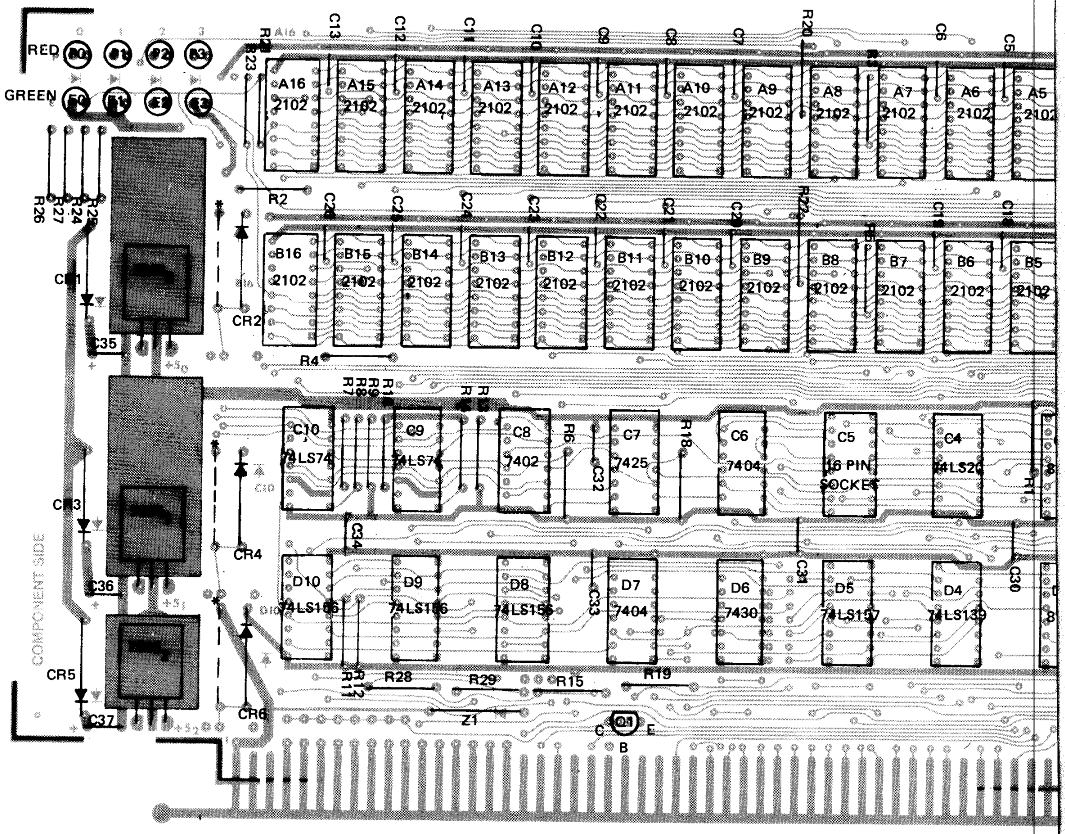
BOX 11651, KNOXVILLE, TN. 37919

TITLE 8192 STATIC MEMORY BD
8KSC (A)(8)

ENGINEER B. SEALS DATE	ENG. APPROVAL Bruce Seals DATE	PART NO. 8KSC
DRAFTSMAN C COFFEY DATE 4-26-76	DRAFTING APPROVAL DATE	DRAWING NO 8KSC-000
SCALE	RELEASED BY Bruce Seals DATE	



8T97B
AØ 79
A1 80
A2 81
A3 31
A4 30
A5 29
A6 82
A7 83
A8 84
A9 34
69
70
IC70
8T97B
1+15
15
IC 1
91L02
IC2
91L02
IC9
91L02
IC10
91L02
IC17
91L02
IC18
91L02
IC25
91L02
IC26
91L02
8 4 5 6 7 2 1 16 15 14
Vcc STAND BY 14
IN4004
D1 D2 D3 D4 D9 D5 D6 D7 D8
R1 1K
R2 1K
R3 1K
R4 1K
R9 1K
R5 1K
R6 1K
R7 1K
R8 1K
Vcc
IC65
74S138
A1Ø 37
A11 87
A12 33
TO PIN 13 IC 33-40
TO PIN 13 IC 41-48
TO PIN 13 IC 49-56
TO PIN 13 IC 57-64
IC67
AMP.
SWITCH
IC66
74S138
A13 85
A14 86
A15 32
IC68
IC68
74S00
SMEMR 47
M. WRITE 68
IC68
74S00
DOØ 36
DO1 35
DO2 88
DO3 89
DO4 38
DO5 39
DO6 40
DO7 90
+8V 1
51
GND 50
100
IC73
7805
IC74
7805
IC75
7805
IC76
7805
Vcc TO IC'S 1-16
Vcc TO IC'S 17-32
Vcc TO IC'S 33-48
Vcc TO IC'S 49-72
D10
D11
D12
D13
C1
100µF
14
Vcc STAND BY

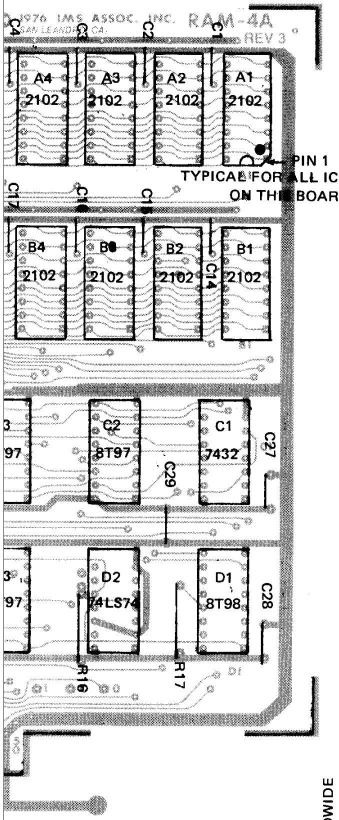


NOT SUPPLIED *

NOTE: THESE ARE USER DEFINED RESISTORS

SEE USER GUIDE SECTION FOR EXPLANATION

REVISIONS			
LTR	DESCRIPTION	DATE	APPROVED
0	REV.2	5/76	
1	MOD & UPDATE	12/76	
2	REV.3	1/77	



PIN 1
TYPICAL FOR ALL IC'S
ON THE BOARD

C5 is a 16 Pin Jumper Socket
All Capacitors are .1uF except
C35 thru C37
Diodes CR1 thru CR6 are 1N4002
Diode Z1 is 1N751A Zener
Transistor Q1 is 2N3604
LED's E0 thru E3 are green
P0 thru P3 are red
Resistors are ¼ Watt
R1 thru R18 are 1K
R19 thru R28 are 220 Ohm
R29 is 100 Ohm

TOLERANCES UNLESS OTHERWISE SPECIFIED FRACTIONS DEC. ANGLES ± ± ±		IMSAI MFG. CORP. SAN LEANDRO, CA.		
APPROVALS	DATE	RAM4A-4 REV.3 ASSEMBLY DIAGRAM		
DRAWN PRU	2/77			
CHECKED		SCALE	SIZE B	DRAWING NO. 91-2010017
		DO NOT SCALE DRAWING		SHEET

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) .luF 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

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.

6) 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 an 4 if the board is to be selecte 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.
A12	Pin 15	Pin 2	Place jumper between pins 15 and 2 if the board is to be selected when this bit is high.
	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.

If the Hardware Write Protect Feature is not used, tie the Protect Line (pin 70 on the edge connector) to ground. This will prevent noise on the Protect Line from triggering an unwanted protect state.

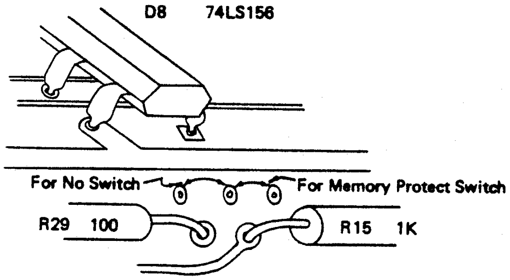


FIGURE 1

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.

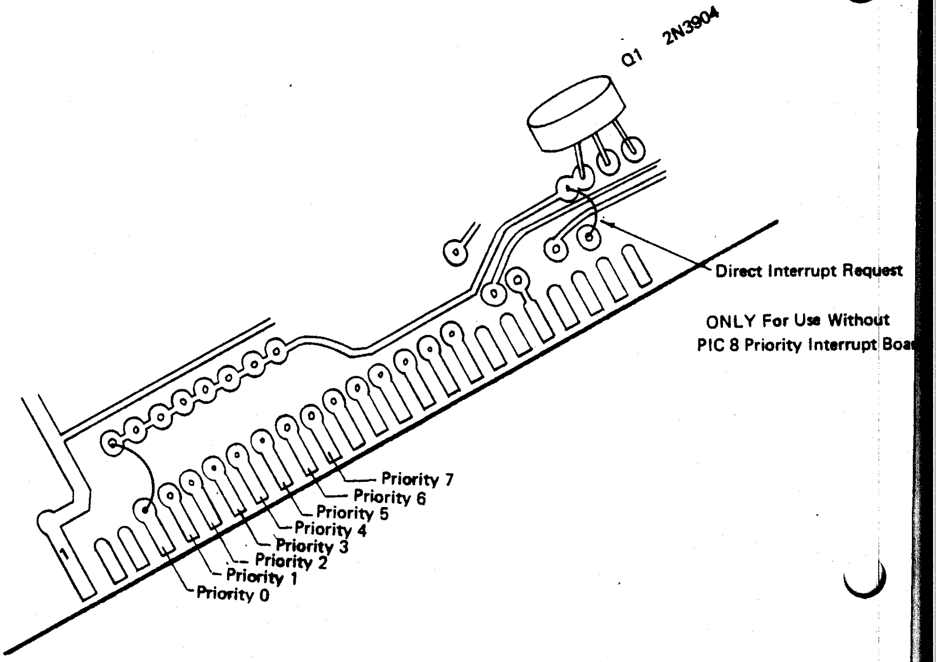

Q1 2N3904
Direct Interrupt Request
ONLY For Use Without
PIC 8 Priority Interrupt Boa
Priority 7
Priority 6
Priority 5
Priority 4
Priority 3
Priority 2
Priority 1
Priority 0
1

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).

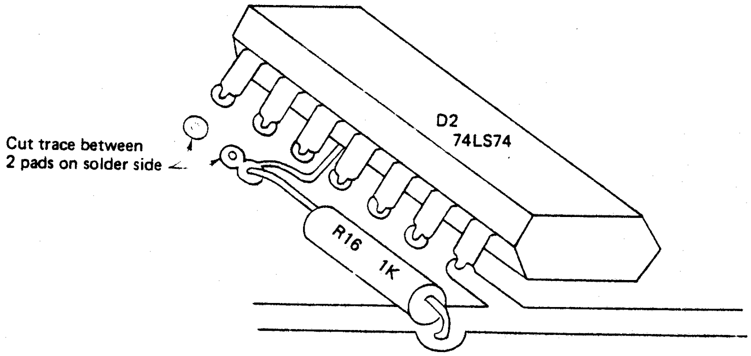


FIGURE 3

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.

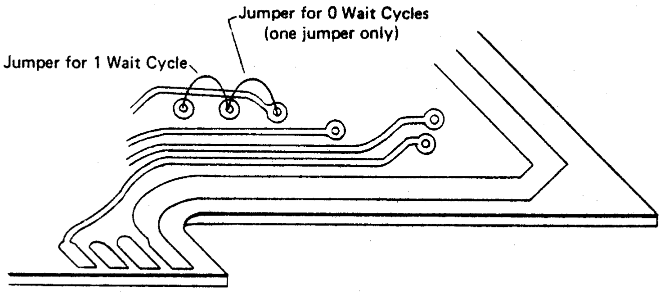


FIGURE 4

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.

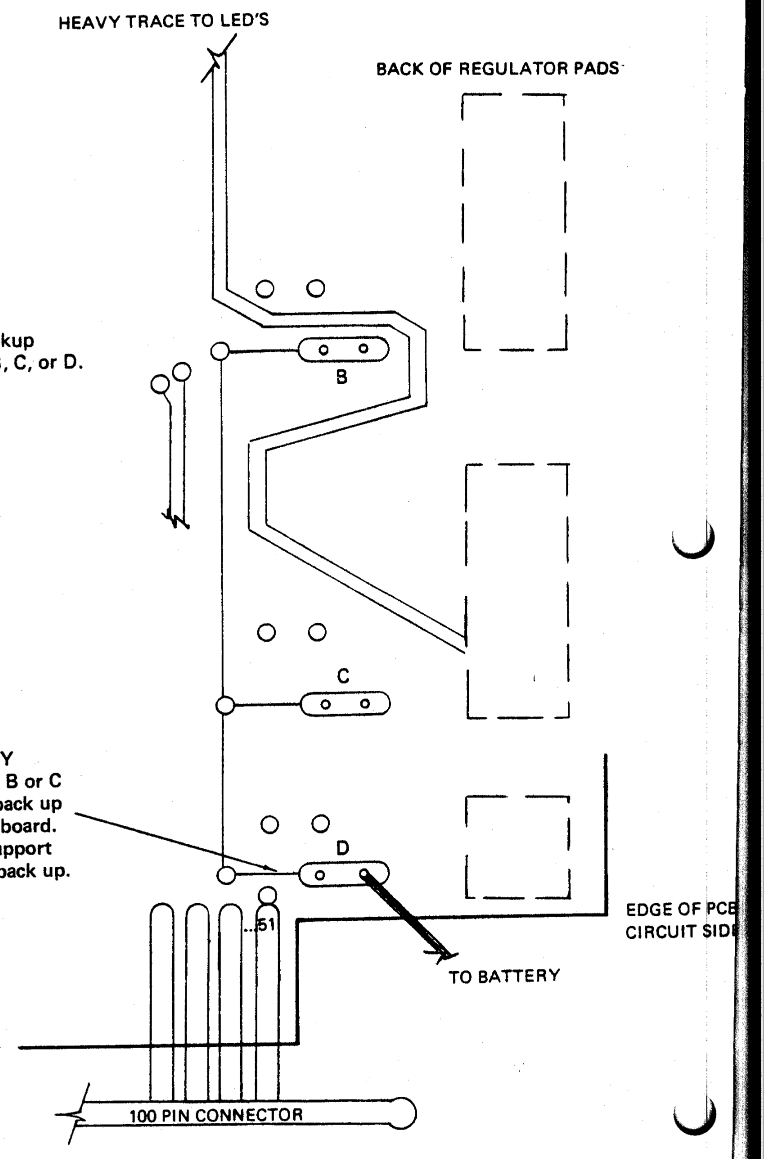
A simple test program for testing some of the special features of your new RAM 4A board follows:

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



Connections for battery hookup can be made at either Pads B, C, or D.

You should cut this trace ONLY if attaching the battery at Pads B or C and you wish your battery to back up only the Memory chips on the board. This is recommended, as the support chips do not need the battery back up.

RAM 4A
Board Tester

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		Meaning
(Hex)	(Binary)	
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 one of switches 8-15.
2. The LED's will display the high 8 bits of the address at the location that failed.
3. Change one of switches 8-15.
4. The LED's will display the low 8 bits of the address.
5. Change one of switches 8-15.
6. The LED's will display the data that the location is supposed to contain.
7. Change one of switches 8-15.
8. The LED's will display the data the location actually contains.
9. Change one 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	
F000	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: 040
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: 050
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
*
```

PROM-4

FUNCTIONAL DESCRIPTION

The IMSAI PROM-4 board provides up to 4K bytes of non-volatile read-only assembly. 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 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-decoupled on-card voltage regulators provide plug-in compatibility with either the IMSAI 8080 or the Altair computer system.

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 feature 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 8702-1702 chips.

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 connectors (dual 50 pin on .125" centers) for electrical 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 atainst short-circuits and thermal overloads.

PIO

FUNCTIONAL DESCRIPTION

The PIO 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 hand-shaking 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 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.

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.

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.

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.

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.

IMSAI 8080
Self-Contained System
Acknowledgement
Revision 4

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
Revision 4

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 - ØDCB	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

The system is available on three media, EPROM, paper tape and Tarbell Format Cassette. The paper tape version may be loaded into RAM using the IMSAI paper tape bootstrap loader, PGM-4A (supplied with the IMSAI 8080 on paper tape; also available on EPROM). The cassette version may be loaded with the tape cassette loader, PGM-5A.

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. Several lines of data may be entered before the slash is typed.

Example:

```
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 functions with 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/ 0   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.

Note that you must type a space between the word "FILE" and the slash before the file name.

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 typing CTRL-X.

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; and
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.

ASM 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.

Tabs (typed as CTRL-I) may be used.

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 tab or 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 first 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 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, that is, separates it from the line number by a single space only. To leave off the name field, the user skips two or more spaces or tabs 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-op 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; i.e., +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 ore 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

The format is

```
label ORG expression
```

where the label is optional but, if present, will be equated to the given expression. The effect of the ORG pseudo-op is to set the current location counter to the value of the given expression. When present in a program, the ORG pseudo-op overides the address given in the ASSM or ASSME command.

END --- End of Assembly

The pweudo-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 --- Equate Symbolic Value

The format is

label EQU expression

Where label is a symbol the value of which will be determined from the expression, and the expression is an expression which when evaluated will be assigned to the symbol given in the name field.

DS --- Define Storage

The 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

The 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	Syntax Error
R	Register Error
A	Argument Error

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.

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 caries 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
```

```
                ;11 MAY 77.  JRB. REMOVE PROMPT CHARACTER TO
                ;            PERMIT READING PAPER TAPE
                ;10 NOV 77.  BRH. CONVERT FOR ASSEMBLY UNDER CP M
                ;            FIX CLEAR BREAKPOINT ROUTINE  ADD PROMPT CHAR
                ;            AND RECOGNIZE TAPS.
                ;
                ;*********************************************************************
                ;*********************************************************************
                ;*****                                                          *****
                ;*****            SELF-CONTAINED SYSTEM  VER 1 REV 4            *****
                ;*****                                                          *****
                ;*****         (C) COPYRIGHT 1976,1977 IMSAI MFG CORP           *****
                ;*****             SAN LEANDRO  CALIFORNIA                      *****
                ;*****                                                          *****
                ;*********************************************************************
                ;*********************************************************************
                ;
0000                    ORG     00H
0000 C34000             JMP     INIT    ;DEAD START
0003 C35700             JMP     INITA   ;RESTART SIO AND ENTER MONITOR
                ;
0008                    ORG     08H
0008 C3510D             JMP     BRKP    ;BREAKPOINT RESTART
                ;
0040                    ORG     40H
                ;
                ; THIS ROUTINE INITIALIZES THE FILE AREA FOR SUBSEQUENT
                ; PROCESSING
                ;
0040 212410     INIT:   LXI     H FILE0
0043 0E4E               MVI     C MAXFIL*FELEN
0045 AF                 XRA     A
0046 77         INIT2:  MOV     M A
0047 23                 INX     H
0048 0D                 DCR     C
0049 C24600             JNZ     INIT2
                ;
                ; CLEAR THE BREAKPOINT TABLE
                ;
004C 0618               MVI     B.NBP*3
004E 210C10             LXI     H BRT
0051 77         INIT3:  MOV     M A
0052 23                 INX     H
0053 05                 DCR     B
0054 C25100             JNZ     INIT3
                ;
                ; SET UP THE SIO BOARD
                ;
0057 3EAA       INITA:  MVI     A.0AAH  ;GET DUMMY MODE WORD
0059 D303               OUT     TTS     ;OUTPUT IT
005B 3E40               MVI     A 40H   ;GET RESET BIT
005D D303               OUT     TTS     ;RESET SIO BOARD
005F 3ECE               MVI     A 0CEH  ;GET REAL MODE WORD
0061 D303               OUT     TTS     ;SET THE MODE FOR REAL
0063 3E37               MVI     A 37H   ;GET THE COMMAND
0065 D303               OUT     TTS     ;OUTPUT IT
```

```
                        ;
                        ; 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
                        ;
ØØ67 31B11Ø     EOR:    LXI     SP AREA-18
ØØ6A CD3AØ1             CALL    CRLF    ;PRINT CR  LF
ØØ6D ØØ                 NOP             ;PROMPT WAS HERE IN REV 3
ØØ6E ØØ                 NOP             ;.. NOPS INSERTED
ØØ6F ØØ                 NOP             ;.  TO MATCH EXISTING
ØØ7Ø ØØ                 NOP             ;.. PROMS. REMOVE WHEN
ØØ71 ØØ                 NOP             ;.  A WHOLE NEW EPROM MASTER NEEDED
ØØ72 CD85ØØ             CALL    READ    ;READ INPUT LINE
ØØ75 23                 INX     H
ØØ76 7E                 MOV     A M     ;FETCH FIRST CHARACTER
ØØ77 FE3A               CPI     '9'+1   ;COMMAND OR LINE NUMBER?
ØØ79 DA1BØ4             JC      LINE    ;JUMP IF LINE FOR FILE
ØØ7C CDAØØ1             CALL    VALC    ;GET COMMAND VALUES
ØØ7F CD56Ø1             CALL    COMM    ;CHECK LEGAL COMMANDS
ØØ82 C367ØØ             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.
                        ;
ØØ85 21C61Ø     READ:   LXI     H IBUF  ;GET INPUT BUFFER ADDRESS
ØØ88 22741Ø             SHLD    ADDS    ;SAVE ADDRESS
ØØ8B 1EØ2               MVI     E,2     ;INITIALIZE CHARACTER COUNT
ØØ8D CDF9Ø2     NEXT:   CALL    IN8     ;READ A LINE
ØØ9Ø 78                 MOV     A B
ØØ91 FE18               CPI     24      ;CHECK FOR CTRL X
ØØ93 CA67ØØ             JZ      EOR
ØØ96 FEØD               CPI     ASCR    ;GET AN ASCII CR
ØØ98 C2B1ØØ             JNZ     DEL
ØØ9B 7D                 MOV     A L
ØØ9C FEC6               CPI     IBUF AND ØFFH   ;CHECK FOR FIRST CHAR
ØØ9E CA67ØØ             JZ      EOR
ØØA1 36ØD               MVI     M ASCR  ;PLACE CR AT END OF LINE
ØØA3 23                 INX     H
ØØA4 36Ø1               MVI     M 1     ;PLACE EOF INDICATOR IN LINE
ØØA6 23                 INX     H
ØØA7 3E19               MVI     A IBUF+83 AND ØFFH
ØØA9 CDF5ØØ             CALL    CLER    ;CLEAR REMAINING BUFFER
ØØAC 21C51Ø             LXI     H IBUF-1
ØØAF 73                 MOV     M E     ;SAVE CHARACTER COUNT
ØØBØ C9                 RET
ØØB1 FE7F       DEL     CPI     127     ;CHECK FOR DELETE CHARACTER
ØØB3 C2C6ØØ             JNZ     CHAR
ØØB6 3EC6               MVI     A IBUF AND ØFFH
```

```
00B8 BD                   CMP     L          ;IS THIS 1ST CHARACTER
00B9 CA8D00               JZ      NEXT
00BC 2B                   DCX     H          ;DECREMENT POINTER
00BD 1D                   DCR     E          ;DECREMENT COUNT
00BE 065F         BSPA:   MVI     B,5FH
00C0 CD0601               CALL    OUT8
00C3 C38D00               JMP     NEXT
00C6 FE20         CHAR:   CPI     ' '        ;CHECK FOR LEGAL CHARACTER
00C8 D2D000               JNC     CHAR1      ;JUMP IF NOT CONTROL CHAR
00CB FE09                 CPI     09H        ;IS IT A TAB?
00CD C28D00               JNZ     NEXT       ;IGNORE IF NOT
00D0 FE5B         CHAR1:  CPI     'Z'-1
00D2 D28D00               JNC     NEXT
00D5 47                   MOV     B,A
00D6 CD0601               CALL    OUT8       ;ECHO CHARACTER
00D9 70                   MOV     M,B
00DA 3E17                 MVI     A,IBUF+81 AND 0FFH
00DC BD                   CMP     L          ;CHECK FOR END OF LINE
00DD CABE00               JZ      BSPA
00E0 23                   INX     H
00E1 1C                   INR     E          ;INCREMENT CHARACTER COUNT
00E2 C38D00               JMP     NEXT
                  ;
                  ;
                  ;
                  ; THIS ROUTINE IS USED TO BLANK OUT A PORTION OF MEMORY
                  ;
00E5 BD           CLER:   CMP     L
00E6 C8                   RZ
00E7 3620                 MVI     M,' '      ;PLACE BLANK IN MEMORY
00E9 23                   INX     H
00EA C3E500               JMP     CLER
                  ;
                  ; SEE IF TTY INPUT READY AND CHECK FOR CTRL X.
                  ; RETURN WITH ZERO SET IFF CTRL-X SEEN.
                  ;
00ED DB03         INK:    IN      TTS        ;GET TTY STATUS
00EF 2F                   CMA                ;INVERT STATUS
00F0 E602                 ANI     TTYDA      ;IS DATA AVAILABLE?
00F2 C0                   RNZ                ;RETURN IF NOT
00F3 CDF900               CALL    IN8        ;GET THE CHAR
00F6 FE18                 CPI     'X'-40H    ;IS IT A CTRL-X?
00F8 C9                   RET
                  ;
                  ; THIS ROUTINE READS A BYTE OF DATA FROM THE USART
                  ;
00F9 DB03         IN8:    IN      TTS        ;READ USART STATUS
00FB E602                 ANI     TTYDA
00FD CAF900               JZ      IN8
0100 DB02                 IN      TTI        ;READ DATA
0102 E67F                 ANI     7FH        ;STRIP OFF PARITY
0104 47                   MOV     B,A
0105 C9                   RET
                  ;
                  ; THIS ROUTINE OUTPUTS A BYTE OF DATA TO THE USART
                  ;
0106 78           OUT8:   MOV     A,B        ;GET CHAR IN A
0107 D60D                 SUI     0DH        ;IS IT A CR?
```

```
Ø1Ø9 CA2CØ1              JZ      OUT81    ;RESET LINE POSITION IF SO
Ø1ØC C6Ø4                ADI     ØDH-Ø9H ;WAS IT A TAB?
Ø1ØE C221Ø1              JNZ     OUT83    ;CONTINUE IF NOT
Ø111 C5                  PUSH    B        ;SAVE TAB CHAR
Ø112 Ø62Ø                MVI     B,' '    ;OUTPUT A SPACE .
Ø114 CD28Ø1      OUT84:  CALL    OUT82
Ø117 3A1911              LDA     LPOS     ;GET CURRENT LINE POSITION
Ø11A E6Ø7                ANI     7        ;IS IT A TAB STOP?
Ø11C C214Ø1              JNZ     OUT84    ;OUTPUT ANOTHER SPACE IF NOT
Ø11F C1                  POP     B        ;RESTORE TAB CHAR
Ø12Ø C9                  RET
Ø121 78          OUT83:  MOV     A,B      ;GET CHAR IN A
Ø122 3C                  INR     A        ;IS IT A NON-PRINTING CHAR?
Ø123 FE21                CPI     ' '+1
Ø125 DA2FØ1              JC      OUT8Ø    ;DON'T BUMP LINE POSITION IF SO
Ø128 3A1911      OUT82:  LDA     LPOS     ;BUMP LINE POSITION...
Ø12B 3C                  INR     A
Ø12C 321911      OUT81:  STA     LPOS
Ø12F DBØ3        OUT8Ø:  IN      TTS      ;READ STATUS
Ø131 E6Ø1                ANI     TTYTR
Ø133 CA2FØ1              JZ      OUT8Ø
Ø136 78                  MOV     A,B
Ø137 D3Ø2                OUT     TTO      ;TRANSMIT DATA
Ø139 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
                 ;
Ø13A Ø6ØD        CRLF:   MVI     B,13     ;CR
Ø13C CDØ6Ø1              CALL    OUT8
Ø13F Ø6ØA        LF:     MVI     B,1Ø     ;LF
Ø141 CDØ6Ø1              CALL    OUT8
Ø144 Ø6FF                MVI     B,255
Ø146 CDØ6Ø1              CALL    OUT8
Ø149 C3Ø6Ø1              JMP     OUT8
                 ;
                 ; THIS ROUTINE JUMPS TO A LOCATION IN MEMORY GIVEN BY
                 ; THE INPUT COMMAND AND BEGINS EXECUTION OF PROGRAM
                 ; THERE
                 ;
Ø14C CD27Ø3      EXEC:   CALL    VCHK     ;CHECK FOR PARAMETER
Ø14F CD3AØ1              CALL    CRLF
Ø152 2A8A1Ø              LHLD    BBUF     ;FETCH ADDRESS
Ø155 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.
                 ;
Ø156 11E4Ø2      COMM:   LXI     D,CTAB   ;COMMAND TABLE ADDRESS
Ø159 3EØ4                MVI     A,4      ;LENGTH OF COMMAND
Ø15B 32951Ø              STA     NCHR     ;SAVE
Ø15E CD65Ø1              CALL    COMS     ;SEARCH TABLE
Ø161 C281Ø4              JNZ     WHAT     ;JUMP IF ILLEGAL COMMAND
Ø164 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.  THE END OF THE TABLE IS MARKED WITH A -1.
                  ; 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.
                  ;
Ø165 2A741Ø       COMS:   LHLD    ADDS    ;FETCH COMPARE ADDRESS
Ø168 1A                   LDAX    D       ;GET NEXT BYTE
Ø169 3C                   INR     A       ;IS IT -1?
Ø16A CA7EØ1               JZ      COMS1   ;ABORT IF SO
Ø16D 3A951Ø               LDA     NCHR    ;GET LENGTH OF STRING
Ø17Ø 4F                   MOV     C A
Ø171 CD8ØØ1               CALL    SEAR    ;COMPARE STRINGS
Ø174 1A                   LDAX    D       ;FETCH VALUE
Ø175 6F                   MOV     L A
Ø176 13                   INX     D
Ø177 1A                   LDAX    D       ;FETCH VALUE
Ø178 67                   MOV     H A
Ø179 C8                   RZ
Ø17A 13                   INX     D       ;SET TO NEXT STRING
Ø17B C365Ø1               JMP     COMS
Ø17E 3C           COMS1:  INR     A       ;CLEAR ZERO FLAG
Ø17F 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.
                  ;
Ø18Ø 1A           SEAR:   LDAX    D       ;FETCH CHARACTER
Ø181 BE                   CMP     M       ;COMPARE CHARACTERS
Ø182 C28CØ1               JNZ     INCA
Ø185 23                   INX     H
Ø186 13                   INX     D
Ø187 ØD                   DCR     C       ;DECREMENT CHARACTER COUNT
Ø188 C28ØØ1               JNZ     SEAR
Ø18B C9                   RET
Ø18C 13           INCA:   INX     D
Ø18D ØD                   DCR     C
Ø18E C28CØ1               JNZ     INCA
Ø191 ØC                   INR     C       ;CLEAR ZERO FLAG
Ø192 C9                   RET
                  ;
                  ; THIS ROUTINE ZEROES OUT A BUFFER IN MEMORY WHICH IS
                  ; THEN USED BY OTHER SCANNING ROUTINES
                  ;
Ø193 AF           ZBUF:   XRA     A       ;GET A ZERO
Ø194 118A1Ø               LXI     D ABUF 12 ;BUFFER ADDRESS
Ø197 Ø6ØC                 MVI     B 12    ;BUFFER LENGTH
Ø199 1B           ZBU1:   DCX     D       ;DECREMENT ADDRESS
```

```
Ø19A 12                 STAX    D       ;ZERO BUFFER
Ø19B Ø5                 DCR     B
Ø19C C299Ø1             JNZ     ZBU1
Ø19F C9                 RET
                ;
                ; THIS ROUTINE CALLS ETRA TO OBTAIN THE INPUT PARAMETER
                ; VALUES AND CALLS AN ERROR ROUTINE IF AN ERROR OCCURRED
                ; IN THAT ROUTINE
                ;
Ø1AØ CDA7Ø1     VALC:   CALL    ETRA    ;GET INPUT PARAMETERS
Ø1A3 DA81Ø4             JC      WHAT    ;JUMP IF ERROR
Ø1A6 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.
                ;
Ø1A7 21ØØØØ     ETRA:   LXI     H,Ø     ;GET A ZERO
Ø1AA 228C1Ø             SHLD    BBUF-2  ;ZERO VALUE
Ø1AD 22761Ø             SHLD    FBUF    ;SET NO FILE NAME
Ø1BØ CD93Ø1             CALL    ZBUF    ;ZERO BUFFER
Ø1B3 21C51Ø             LXI     H,IBUF-1
Ø1B6 23         VAL1:   INX     H
Ø1B7 7E                 MOV     A,M     ;FETCH INPUT CHARACTER
Ø1B8 FE2Ø               CPI     ' '     ;LOOK FOR FIRST CHARACTER
Ø1BA 3F                 CMC
Ø1BB DØ                 RNC             ;RETURN IF NO CARRY
Ø1BC C2B6Ø1             JNZ     VAL1    ;JUMP IF NO BLACK
Ø1BF 22961Ø             SHLD    PNTR    ;SAVE POINTER
Ø1C2 CD39Ø9             CALL    SBLK    ;SCAN TO FIRST PARAMETER
Ø1C5 3F                 CMC
Ø1C6 DØ                 RNC             ;RETURN IF CR
Ø1C7 FE2F               CPI     '/'
Ø1C9 C2F1Ø1             JNZ     VAL5    ;NO FILE NAME
Ø1CC 11761Ø             LXI     D,FBUF  ;NAME FOLLOWS PUT IN FBUF
Ø1CF ØEØ5               MVI     C,NMLEN
Ø1D1 23         VAL2:   INX     H
Ø1D2 7E                 MOV     A,M
Ø1D3 FE2F               CPI     '/'
Ø1D5 CAE1Ø1             JZ      VAL3
Ø1D8 ØD                 DCR     C
Ø1D9 FA81Ø4             JM      WHAT
Ø1DC 12                 STAX    D       ;STORE FILE NAME
Ø1DD 13                 INX     D
Ø1DE C3D1Ø1             JMP     VAL2
Ø1E1 3E2Ø       VAL3:   MVI     A,' '   ;GET AN ASCII SPACE
Ø1E3 ØD         VAL4:   DCR     C
Ø1E4 FAECØ1             JM      DONE
Ø1E7 12                 STAX    D       ;FILL IN WITH SPACES
Ø1E8 13                 INX     D
Ø1E9 C3E3Ø1             JMP     VAL4
Ø1EC CD45Ø9     DONE:   CALL    SBL2
Ø1EF 3F                 CMC
Ø1FØ DØ                 RNC
Ø1F1 117E1Ø     VAL5:   LXI     D,ABUF
```

```
Ø1F4 CD9AØB             CALL    ALPS    ;PLACE PARAMETER IN BUFFER
Ø1F7 78                 MOV     A B     ;GET DIGIT COUNT
Ø1F8 FEØ5               CPI     5       ;CHECK NUMBER OF DIGITS
Ø1FA 3F                 CMC
Ø1FB D8                 RC              ;RETURN IF TOO MANY DIGITS
Ø1FC Ø17E1Ø             LXI     B ABUF
Ø1FF CD48Ø2             CALL    AHEX    ;CONVERT VALUE
Ø2Ø2 D8                 RC              ;ILLEGAL CHARACTER
Ø2Ø3 228A1Ø             SHLD    BBUF    ;SAVE   IN BINARY BUFFER
Ø2Ø6 217E1Ø             LXI     H.ABUF
Ø2Ø9 CDE3Ø5             CALL    NORM    ;NORMALIZE ASCII VALUE
Ø2ØC CD39Ø9             CALL    SBLK    ;SCAN TO NEXT PARAMETER
Ø2ØF 3F                 CMC
Ø21Ø DØ                 RNC             ;RETURN IF CR
Ø211 11821Ø             LXI     D ABUF-4
Ø214 CD9AØB             CALL    ALPS    ;PLACE PARAMETER IN BUFFER
Ø217 78                 MOV     A.B     ;GET DIGIT COUNT
Ø218 FEØ5               CPI     5       ;CHECK NUMBER OF DIGITS
Ø21A 3F                 CMC
Ø21B D8                 RC              ;RETURN IF TOO MANY DIGITS
Ø21C Ø1821Ø             LXI     B ABUF-4
Ø21F CD48Ø2             CALL    AHEX    ;CONVERT VALUE
Ø222 D8                 RC              ;ILLEGAL VALUE
Ø223 228C1Ø             SHLD    BBUF-2  ;SAVE IN BINARY BUFFER
Ø226 21821Ø             LXI     H.ABUF-4
Ø229 CDE3Ø5             CALL    NORM    ;NORMALIZE ASCII VALUE
Ø22C B7                 ORA     A       ;CLEAR CARRY
Ø22D 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
                ;
Ø22E 21ØØØØ     ADEC·   LXI     H Ø     ;GEP A 16 BIT ZERO
Ø231 ØA         ADE1·   LDAX    B       ;FETCH ASCII DIGIT
Ø232 B7                 ORA     A       ;SET ZERO FLAG
Ø233 C8                 RZ              ;RETURN IFF FINISHED
Ø234 54                 MOV     D H     ;SAVE CURRENT VALUE
Ø235 5D                 MOV     E L     ;SAVE CURRENT VALUE
Ø236 29                 DAD     H       ;TIMES TWO
Ø237 29                 DAD     H       ;TIMES TWO
Ø238 19                 DAD     D       ;ADD IN ORIGINAL VALUE
Ø239 29                 DAD     H       ;TIMES TWO
Ø23A D63Ø               SUI     48      ;ASCII BIAS
Ø23C FEØA               CPI     1Ø      ;CHECK FOR LEGAL VALUE
Ø23E 3F                 CMC
Ø23F D8                 RC              ;RETURN IF ERROR
Ø24Ø 5F                 MOV     E A
Ø241 16ØØ               MVI     D Ø
Ø243 19                 DAD     D       :ADD IN NEXT DIGIT
Ø244 Ø3                 INX     B       ;INCREMENT POINTER
Ø245 C331Ø2             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.
                    ;
Ø248 21ØØØØ         AHEX    LXI     H Ø     ;GET A 16 BIT ZERO
Ø24B ØA             AHE1:   LDAX    B       ;FETCH ASCII DIGIT
Ø24C B7                     ORA     A       ;SET ZERO FLAG
Ø24D C8                     RZ              ;RETURN IF DONE
Ø24E 29                     DAD     H       ;LEFT SHIFT
Ø24F 29                     DAD     H       ;LEFT SHIFT
Ø25Ø 29                     DAD     H       ;LEFT SHIFT
Ø251 29                     DAD     H       ;LEFT SHIFT
Ø252 CD5FØ2                 CALL    AHS1    ;CONVERT TO BINARY
Ø255 FE1Ø                   CPI     1ØH     ;CHECK FOR LEGAL VALUE
Ø257 3F                     CMC
Ø258 D8                     RC              ;RETURN IF ERROR
Ø259 85                     ADD     L
Ø25A 6F                     MOV     L.A
Ø25B Ø3                     INX     B       ;INCREMENT POINTER
Ø25C C34BØ2                 JMP     AHE1
                    ;
                    ; THIS SUBROUTINE CONVERTS ASCII HEX DIGITS INTO BINARY
                    ;
Ø25F D63Ø           AHS1:   SUI     48      ;ASCII BIAS
Ø261 FEØA                   CPI     1Ø      ;DIGIT Ø-1Ø
Ø263 D8                     RC
Ø264 D6Ø7                   SUI     7       ;ALPHA BIAS
Ø266 C9                     RET
                    ;
                    ;
                    ; THIS ROUTINE CONVERTS A BINARY VALUE TO ASCII
                    ; HEXADECIMAL AND OUTPUTS THE CHARACTERS TO THE TTY.
                    ;
Ø267 CDACØ2         HOUT:   CALL    BINH    ;CONVERT VALUE
Ø26A 21741Ø                 LXI     H HCON  ;CONVERSION AREA
Ø26D 46             CHOT:   MOV     B.M     ;FETCH OUTPUT CHARACTER
Ø26E CDØ6Ø1                 CALL    OUT8    ;OUTPUT CHARACTER
Ø271 23                     INX     H
Ø272 46                     MOV     B M     ;FETCH CHARACTER
Ø273 C3Ø6Ø1                 JMP     OUT8    ;OUTPUT CHARACTER AND RETURN
                    ;
                    ; THIS ROUTINE DOES THE SAME AS ABOVE BUT OUTPUTS A
                    ; BLANK AFTER THE LAST CHARACTER
                    ;
Ø276 CD67Ø2         HOTB:   CALL    HOUT    ;CONVERT AND OUTPUT
                    ;
                    ; THIS ROUTINE OUTPUTS A BLANK.
                    ;
Ø279 Ø62Ø           BLK1:   MVI     B,' '   ;GET A BLANK
Ø27B C3Ø6Ø1                 JMP     OUT8    ;OUTPUT IT AND RETURN
                    ;
                    ; THIS ROUTINE CONVERTS A BINARY VALUE TO ASCII
                    ; DECIMAL DIGITS AND OUTPUTS THE CHARACTERS TO THE TTY
                    ;
Ø27E CDC9Ø2         DOUT:   CALL    BIND    ;CONVERT VALUE
Ø281 CD6AØ2                 CALL    HOUT 3  ;OUTPUT VALUE  2 DIGITS
Ø284 23                     INX     H
Ø285 46                     MOV     B M     ;GET LAST DIGIT
Ø286 C3Ø6Ø1                 JMP     OUT8    ;OUTPUT AND RETURN
                    ;
```

```
                ; 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.
                ;
0289 2A8A10     ACHK:   LHLD    BBUF     ;FETCH START ADDRESS
028C 3A8D10             LDA     BBUF 3   ;STOP ADRESS  HIGH
028F BC                 CMP     H        ;COMPARE ADDRESSES
0290 C29B02             JNZ     ACH1
0293 3A8C10             LDA     BBUF 2   ;STOP ADDRESS  LOW
0296 BD                 CMP     L        ;COMPARE ADDRESSES
0297 C29B02             JNZ     ACH1
029A 37                 STC              ;SET CARRY IF EQUAL
029B 23         ACH1:   INX     H        ;INCREMENT START ADDRESS
029C 228A10             SHLD    BBUF     ;STORE START ADDRESS
029F C9                 RET
                ;
                ;
                ; THIS ROUTINE OUTPUTS CHARACTERS OF A STRING
                ; UNTIL A CARRIAGE RETURN IS FOUND.
                ;
02A0 46         SCRN:   MOV     B M      ;FETCH CHARACTER
02A1 3E0D               MVI     A,13     ;CARRIAGE RETURN
02A3 B8                 CMP     B        ;CHARACTER = CR?
02A4 C8                 RZ
02A5 CD0601             CALL    OUT8     ;OUTPUT CHARACTER
02A8 23                 INX     H        ;INCREMENT ADDRESS
02A9 C3A002             JMP     SCRN
                ;
                ;
                ; THIS ROUTINE CONVERTS THE BINARY VALUE IN REG A INTO
                ; ASCII HEXADECIMAL DIGITS AND STORES THEM IN MEMORY
                ;
02AC 217410     BINH:   LXI     H.HCON   ;CONVERSION
02AF 47                 MOV     B A      ;SAVE VALUE
02B0 1F                 RAR
02B1 1F                 RAR
02B2 1F                 RAR
02B3 1F                 RAR
02B4 CDBF02             CALL    BIN1
02B7 77                 MOV     M A
02B8 23                 INX     H
02B9 78                 MOV     A B
02BA CDBF02             CALL    BIN1     ;CONVERT TO ASCII
02BD 77                 MOV     M,A
02BE C9                 RET
                ;
                ; THIS ROUTINE CONVERTS A VALUE TO HEXADECIMAL
                ;
02BF E60F       BIN1:   ANI     0FH      ;LOW 4 BITS
02C1 C630               ADI     48       ;CONVERT TO ASCII
02C3 FE3A               CPI     58       ;DIGIT 0-9
02C5 D8                 RC
02C6 C607               ADI     7        ;MODIFY FOR A-F
02C8 C9                 RET
                ;
                ;
```

```
                    ; THIS ROUTINE CONVERTS THE BINARY VALUE IN REG A INTO
                    ; ASCII DECIMAL DIGITS AND STORES THEM IN MEMORY
                    ;
02C9 217410         BIND:   LXI     H,HCON   ;CONVERSION ADDRESS
02CC 0664                   MVI     B,100
02CE CDDA02                 CALL    BID1     ;CONVERT HUNDREDS DIGIT
02D1 060A                   MVI     B,10
02D3 CDDA02                 CALL    BID1     ;CONVERT TENS DIGIT
02D6 C630                   ADI     '0'      ;GET UNITS DIGIT
02D8 77                     MOV     M,A      ;STORE IN MEMORY
02D9 C9                     RET
                    ;
                    ; THIS ROUTINE CONVERTS A VALUE TO DECIMAL
                    ;
02DA 362F           BID1:   MVI     M,'0'-1 ;INITIALIZE DIGIT COUNT
02DC 34                     INR     M
02DD 90                     SUB     B        ;CHECK DIGIT
02DE D2DC02                 JNC     BID1+2
02E1 80                     ADD     B        ;RESTORE VALUE
02E2 23                     INX     H
02E3 C9                     RET
                    ;
                    ; LEGAL COMMAND TABLE
                    ;
02E4 44554D50       CTAB:   DB      'DUMP'   ;DUMP COMMAND
02E8 2F03                   DW      DUMP     ;COMMAND ADDRESS
02EA 45584543               DB      'EXEC'   ;EXECUTE COMMAND
02EE 4C01                   DW      EXEC     ;COMMAND ADDRESS
02F0 454E5452               DB      'ENTR'   ;ENTER COMMAND
02F4 9D04                   DW      ENTR
02F6 46494C45               DB      'FILE'   ;FILE COMMAND
02FA 6503                   DW      FILE     ;COMMAND ADDRESS
02FC 4C495354               DB      'LIST'   ;LIST COMMAND
0300 F605                   DW      LIST     ;COMMAND ADDRESS
0302 44454C54               DB      'DELT'   ;DELETE COMMAND
0306 0D06                   DW      DELL     ;COMMAND ADDRESS
0308 4153534D               DB      'ASSM'   ;ASSEMBLE COMMAND
030C 8406                   DW      ASSM     ;COMMAND ADDRESS
030E 50414745               DB      'PAGE'   ;PAGE TRANSFER COMMAND
0312 4903                   DW      PAGE     ;COMMAND ADDRESS
0314 43555354               DB      'CUST'   ;CUSTOMER COMMAND
0318 0020                   DW      2000H    ;COMMAND ADDRESS
031A 4252454B               DB      'BREK'   ;BREAKPOINT COMMAND
031E F50C                   DW      BREAK    ;COMMAND ADDRESS
0320 50524F43               DB      'PROC'   ;;PROCEED COMMAND
0324 AF0D                   DW      PROC     ;COMMAND ADDRESS
0326 FF                     DB      0FFH
                    ;
                    ; THIS ROUTINE CHECKS IF ANY PARAMETERS WERE ENTERED
                    ; WITH THE COMMAND  IF NOT AN ERROR MESSAGE IS ISSUED
                    ;
0327 3A7E10         VCHK:   LDA     ABUF     ;FETCH PARAMETER BYTE
032A B7                     ORA     A        ;SET FLAGS
032B CA8104                 JZ      WHAT     ;NO PARAMETER
032E C9                     RET
                    ;
                    ;
                    ; THIS ROUTINE DUMPS OUT THE CONTENTS OF MEMORY FROM
```

```
                      ; THE START TO FINAL ADDRESSES GIVEN IN THE COMMAND.
                      ;
Ø32F CD27Ø3           DUMP:   CALL    VCHK     ;CHECK FOR PARAMETERS
Ø332 CD3AØ1           DUMS:   CALL    CRLF     ;START NEW LINE
Ø335 2A8A1Ø           DUM1:   LHLD    BBUF     ;FETCH MEMORY ADDRESS
Ø338 7E                       MOV     A,M
Ø339 CD76Ø2                   CALL    HOTB     ;OUTPUT VALUE
Ø33C CD89Ø2                   CALL    ACEK     ;CHECK ADDRESS
Ø33F D8                       RC               ;RETURN IF FINISHED
Ø34Ø 7D                       MOV     A.L      ;IS NEXT ADDRESS
Ø341 E6ØF                     ANI     ØFH      ;  DIVISIBLE BY 16?
Ø343 C235Ø3                   JNZ     DUM1
Ø346 C332Ø3                   JMP     DUMS
                      ;
                      ;
                      ; THIS ROUTINE WILL MOVE 256 BYTES FROM 1ST ADDRESS
                      ; GIVEN IN COMMAND TO 2ND ADDRESS IN COMMAND
                      ;
Ø349 CD27Ø3           PAGE:   CALL    VCHK     ;CHECK FOR PARAMETER
Ø34C 3A821Ø                   LDA     ABUF+4   ;FETCH 2ND PARAMETER
Ø34F B7                       ORA     A        ;DOES 2ND PARAMETER EXIST?
Ø35Ø CA81Ø4                   JZ      WHAT
Ø353 2A8A1Ø                   LHLD    BBUF     ;FETCH MOVE TO ADDRESS
Ø356 EB                       XCHG
Ø357 2A8C1Ø                   LHLD    BBUF+2   ;FETCH MOVE TO ADDRESS
Ø35A Ø6ØØ                     MVI     B,Ø      ;SET COUNTER
Ø35C 1A               PAG1:   LDAX    D
Ø35D 77                       MOV     M.A
Ø35E 23                       INX     H
Ø35F 13                       INX     D
Ø36Ø Ø5                       DCR     B        ;DECREMENT COUNT
Ø361 C25CØ3                   JNZ     PAG1
Ø364 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
                      ;
Ø365 CD3AØ1           FILE:   CALL    CRLF
                      ; CHECK FOR FILE PARAMETERS
Ø368 3A761Ø                   LDA     FBUF
Ø36B B7                       ORA     A
Ø36C CAEØØ3                   JZ      FOUT     ;NO - GO LIST
Ø36F CD3FØ4                   CALL    FSEA     ;LOOK UP FILE
Ø372 EB                       XCHG             ;PNTR IN DE
Ø373 C28AØ3                   JNZ     TEST
                      ; NO ENTRY
Ø376 3A7E1Ø                   LDA     ABUF     ;CHECK FOR PARAM
Ø379 B7                       ORA     A
Ø37A CA84Ø4                   JZ      WHA1     ;NO?? - ERROR
                      ; CHECK FOR ROOM IN DIRECTORY
Ø37D 3A7D1Ø                   LDA     FEF
Ø38Ø B7                       ORA     A
Ø381 C29FØ3                   JNZ     ROOM
Ø384 2192Ø4                   LXI     H EMES1
Ø387 C387Ø4                   JMP     MESS
```

```
                    ; ENTRY FOUND ARE THESE PARAMETERS
038A 3A7E10     TEST:   LDA     ABUF
038D B7                 ORA     A
038E CAB203             JZ      SWAPS
0391 2A8A10             LHLD    BBUF
0394 7C                 MOV     A,H
0395 B5                 ORA     L
0396 CAB203             JZ      SWAPS
0399 219704             LXI     H,EMES2 ;NO-NO CAN'T DO
039C C38704             JMP     MESS    ;IT - DELETE FIRST
                    ; MOVE FILE NAME TO BLOCK POINTED TO BY FREAD
039F 2A7B10     ROOM:   LHLD    FREAD
03A2 EB                 XCHG
03A3 217610             LXI     H,FBUF  ;FILE NAME POINTER IN H L
03A6 D5                 PUSH    D
03A7 0E05               MVI     C,NMLEN ;NAME LENGTH COUNT
03A9 7E         MOV23:  MOV     A,M
03AA 12                 STAX    D
03AB 13                 INX     D
03AC 0D                 DCR     C       ;TEST COUNT
03AD 23                 INX     H
03AE C2A903             JNZ     MOV23
03B1 D1                 POP     D       ;RESTORE ENTRY POINTER
                    ; MAKE FILE POINTED TO BY D,E CURRENT
03B2 212410     SWAPS:  LXI     H,FILE0
03B5 0E0D               MVI     C,FELEN ;ENTRY LENGTH
03B7 1A         SWAP:   LDAX    D
03B8 46                 MOV     B,M
03B9 77                 MOV     M,A     ;EXCHANGE
03BA 78                 MOV     A,B
03BB 12                 STAX    D
03BC 13                 INX     D
03BD 23                 INX     H       ;BUMP POINTERS
03BE 0D                 DCR     C       ;TEST COUNT
03BF C2B703             JNZ     SWAP
                    ; CHECK FOR 2ND PARAMETER
03C2 3A7E10             LDA     ABUF
03C5 B7                 ORA     A
03C6 CAEA03             JZ      FOOT    ;NO SECOND PARAMETER
                    ; PROCESS 2ND PARAMETER
03C9 2A8A10             LHLD    BBUF    ;GET ADDRESS
03CC 222910             SHLD    BOFP    ;SET BEGIN
03CF 222B10             SHLD    EOFP    ;SET END
03D2 7D                 MOV     A,L     ;IS ADDRESS ZERO?
03D3 B4                 ORA     H
03D4 CAD903             JZ      FIL35   ;YES
03D7 3601       FIL30:  MVI     M,1     ;NON-ZERO - SET EOF
03D9 AF         FIL35:  XRA     A       ;AND MAX LINE #
03DA 322D10             STA     MAXL
03DD C3EA03             JMP     FOOT    ;OUTPUT PARAMETERS
03E0 3ACA10     FOUT:   LDA     IBUF+4
03E3 FE53               CPI     'S'     ;IS COMMAND FILES?
03E5 0E06               MVI     C,MAXFIL
03E7 CAEC03             JZ      FOUL
03EA 0E01       FOOT:   MVI     C,1
                    ; OUTPUT THE # OF ENTRIES IN C
03EC 212410     FOUL:   LXI     H,FILE0
03EF 79                 MOV     A,C
```

```
Ø3FØ 327D1Ø     FINE:   STA     FOCNT   ;SAVE COUNT
Ø3F3 E5                 PUSH    H
Ø3F4 11Ø5ØØ             LXI     D,NMLEN
Ø3F7 19                 DAD     D
Ø3F8 7E                 MOV     A,M
Ø3F9 B7                 ORA     A
Ø3FA C2ØAØ4             JNZ     FOOD    ;NON ZERO  OK TO OUTPUT
Ø3FD 23                 INX     H
Ø3FE 86                 ADD     M
Ø3FF 23                 INX     H
Ø4ØØ C2ØAØ4             JNZ     FOOD
Ø4Ø3 33                 INX     SP
Ø4Ø4 33                 INX     SP
Ø4Ø5 23                 INX     H
Ø4Ø6 23                 INX     H
Ø4Ø7 C31FØ4             JMP     FEET
                ; HAVE AN ENTRY TO OUTPUT
Ø4ØA E1         FOOD:   POP     H       ;PTR
Ø4ØB ØEØ5               MVI     C,NMLEN
Ø4ØD 46         FAST:   MOV     B,M     ;LOAD CHARACTER TO B
Ø4ØE CDØ6Ø1             CALL    OUT8
Ø411 ØD                 DCR     C
Ø412 23                 INX     H
Ø413 C2ØDØ4             JNZ     FAST    ;DO THE REST
                ; NOW OUTPUT BEGIN END PTRS
Ø416 CD2BØ4             CALL    FOOL    ;OUTPUT BEGIN
Ø419 CD2BØ4             CALL    FOOL    ;OUTPUT END
Ø41C CD3AØ1             CALL    CRLF    ;AND C R
                ; TEST COUNT  H L POINTS PAST EOFP
Ø41F 11Ø4ØØ     FEET:   LXI     D,FELEN-NMLEN-4
Ø422 19                 DAD     D       ;MOVE TO NEXT ENTRY
Ø423 3A7D1Ø             LDA     FOCNT
Ø426 3D                 DCR     A       ;TEST COUNT
Ø427 C2FØØ3             JNZ     FINE    ;MORE TO DO
Ø42A C9                 RET             ;DONE!
                ; OUTPUT NUMBER POINTED TO BY H L
                ; ON RET. H,L POINT 2 WORDS LATER
Ø42B CD79Ø2     FOOL:   CALL    BLK1    ;SPACE
Ø42E 23                 INX     H
Ø42F 7E                 MOV     A,M
Ø43Ø 2B                 DCX     H
Ø431 E5                 PUSH    H
Ø432 CD67Ø2             CALL    HOUT    ;OUTPUT
Ø435 E1                 POP     H
Ø436 7E                 MOV     A,M
Ø437 23                 INX     H
Ø438 23                 INX     H
Ø439 E5                 PUSH    H
Ø43A CD76Ø2             CALL    HOTB    ;OUTPUT
Ø43D E1                 POP     H       ;RESTORE H L
Ø43E 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 >Ø AND FREAD TO THE ADDR OF ENTRY
                ;
Ø43F AF         FSEA:   XRA     A
```

```
0440 327D10             STA     FEF     ;CLAIM NO FREE ENTRIES
0443 0606               MVI     B,MAXFIL        ;COUNT OF ENTRIES
0445 112410             LXI     D,FILE0 ;TABLE ADDRESS
0448 217610     FSE10:  LXI     H,FBUF
044B 0E05               MVI     C,NMLEN
044D CD8001             CALL    SEAR    ;TEST STRINGS
0450 F5                 PUSH    PSW     ;SAVE FLAG
0451 D5                 PUSH    D
0452 1A                 LDAX    D       ;GET BOFF
0453 B7                 ORA     A       ;EMPTY ENTRY?
0454 C27504             JNZ     FSE20
0457 13                 INX     D       ;STORE OTHER WORD
0458 1A                 LDAX    D
0459 B7                 ORA     A
045A C27504             JNZ     FSE20   ;NOPE-GO TEST FOR MATCH
045D EB                 XCHG
045E 11FAFF             LXI     D,-NMLEN-1
0461 19                 DAD     D       ;MOV TO BEGINNING
0462 227B10             SHLD    FREAD   ;SAVE ADDR
0465 7A                 MOV     A,D
0466 327D10             STA     FEF     ;SET FREE ENTRY FOUND
0469 E1                 POP     H       ;RESTOR INTERIM PTR
046A F1                 POP     PSW     ;UNJUNK STACK
                ; MOVE TO NEXT ENTRY
046B 110800     FSE15:  LXI     D,FELEN+NMLEN
046E 19                 DAD     D
046F EB                 XCHG            ;NEXT ENTRY ADDR IN DE
0470 05                 DCR     B       ;TEST COUNT
0471 C8                 RZ              ;DONE--NOPE
0472 C34804             JMP     FSE10   ;TRY NEXT
                ;ENTRY WASN'T FREE. TEST FOR MATCH
0475 E1         FSE20:  POP     H
0476 F1                 POP     PSW
0477 C26B04             JNZ     FSE15   ;IF ZERO CLEAR. NO MATCH
                ;ENTRY FOUND
047A 11FBFF             LXI     D,-NMLEN        ;BACKUP
047D 19                 DAD     D       ;H.L POINTS TO ENTRY
047E 7A                 MOV     A,D
047F B7                 ORA     A       ;CLEAR ZERO
0480 C9                 RET             ;THAT'S ALL
                ;
                ;
                ; OUTPUT ERROR MESSAGE FOR ILLEGAL COMMAND
                ;
0481 CD3A01     WHAT:   CALL    CRLF    ;OUT CRLF
0484 218D04     WHA1:   LXI     H,EMES  ;MESSAGE ADDRESS
0487 CDA002     MESS:   CALL    SCRN
048A C36700             JMP     EOR
                ;
048D 574841540DEMES:    DB      'WHAT',13
0492 46554C4C0DEMES1:   DB      'FULL',13
0497 4E4F204F4FEMES2:   DB      'NO NO',13
                ;
                ;
                ; 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
                    ;
049D CD2703         ENTR:   CALL    VCHK     ;CHECK FOR PARAMETERS
04AØ CDA9Ø4                 CALL    ENTS
04A3 DA81Ø4                 JC      WHAT
04A6 C33AØ1                 JMP     CRLF
                    ;
                    ;
ØØ2F =              EEND    EQU     '/'      ;TERMINATION CHAR
04A9 CD3AØ1         ENTS:   CALL    CRLF
04AC CD85ØØ                 CALL    READ     ;READ INPUT DATA
04AF 21C61Ø                 LXI     H,IBUF   ;SET LINE POINTER
04B2 22961Ø                 SHLD    PNTR     ;SAVE POINTER
04B5 CD93Ø1         ENT1:   CALL    ZBUF     ;CLEAR BUFFER
04B8 CD39Ø9                 CALL    SBLK     ;CAN TO FIRST VALUE
04BB DAA9Ø4                 JC      ENTS     ;JUMP IF CR FOUND
04BE FE2F                   CPI     EEND
04CØ C8                     RZ               ;RETURN CARRY IS ZERO
04C1 CD9AØB                 CALL    ALPS     ;PLACE VALUE IN BUFFER
04C4 78                     MOV     A,B      ;GET DIGIT COUNT
04C5 FEØ3                   CPI     3        ;CHECK NUR OF DIGITS
04C7 3F                     CMC
04C8 D8                     RC               ;RETURN IF MORE THAN    2 DIGITS
04C9 Ø17E1Ø                 LXI     B,ABUF   ;CONVERSION ADDRESS
04CC CD49Ø2                 CALL    AHEX     ;CONVERT VALUE
04CF D8                     RC               ;ERROR IN HEX CHARACTER
04DØ 7D                     MOV     A,L
04D1 2A8A1Ø                 LHLD    BBUF     ;FETCH MEMORY ADDRESS
04D4 77                     MOV     M,A      ;PUT IN MEMORY
04D5 CD9BØ2                 CALL    ACH1     ;INCREMENT MEMORY LOCATION
04D8 C3B5Ø4                 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 (ØØØØ 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.
                    ;
04DB 3A241Ø         LINE:   LDA     FILEØ    ;IS A FILE DEFINED? ..
04DE B7                     ORA     A
04DF CA81Ø4                 JZ      WHAT     ;ABORT IF NOT
04E2 ØEØ4                   MVI     C,4      ;NO OF DIGITS TO CHECK
04E4 21C51Ø                 LXI     H,IBUF-1         ;INITIALIZE ADDRESS
04E7 23             LICK:   INX     H
04E8 7E                     MOV     A,M      ;FETCH LINE DIGIT
04E9 FE3Ø                   CPI     'Ø'      ;CHECK FOR VALID NUMBER
04EB DA81Ø4                 JC      WHAT
04EE FE3A                   CPI     '9'+1
04FØ D281Ø4                 JNC     WHAT
```

```
04F3 ØD                 DCR     C
04F4 C2E7Ø4             JNZ     LICK
04F7 22741Ø             SHLD    ADDS    ;FIND ADDRESS
04FA 11301Ø             LXI     D.MAXL-3        ;GET ADDRESS
04FD CDC8Ø5             CALL    COMØ
Ø5ØØ D22ØØ5             JNC     INSR
                ; GET HERE IF NEW LINE IS GREATER THAN MAXIMUM LINE #
Ø5Ø3 23                 INX     H
Ø5Ø4 CDB8Ø5             CALL    LODM    ;GET NEW LINE NUMBER
Ø5Ø7 21301Ø             LXI     H.MAXL+3
Ø5ØA CDCØØ5             CALL    STOM    ;MAKE IT MAXIMUM LINE NUMBER
Ø5ØD 11C51Ø             LXI     D.IBUF-1
Ø51Ø 2A2B1Ø             LHLD    EOFP    ;END OF FILE POSITION
Ø513 ØEØ1               MVI     C.1
Ø515 CDA6Ø5             CALL    LMOV    ;PLACE LINE IN FILE
Ø519 36Ø1       SFOF:   MVI     M 1     ;END OF FILE INDICATOR
Ø51A 222B1Ø             SHLD    EOFP    ;END OF FILE ADDRESS
Ø51D C367ØØ             JMP     EOR
                ; GET HERE IF NEW LINE MUST BE INSERTED INTO ALREADY
                ; EXISTING FILE AREA
Ø52Ø CD78Ø5     INSR:   CALL    FIN1    ;FIND LINE IN FILE
Ø523 ØEØ2               MVI     C 2
Ø525 CA29Ø5             JZ      EQUL
Ø528 ØD                 DCR     C       ;NEW LN NOT EQUAL TO SOME OLD LN
Ø529 46         EQUL:   MOV     B.M
Ø52A 2B                 DCX     H
Ø52B 36Ø2               MVI     M.2     ;MOVE LINE INDICATOR
Ø52D 22721Ø             SHLD    INSP    ;INSERT LINE POSITION
Ø53Ø 3AC51Ø             LDA     IBUF-1  ;NEW LN COUNT
Ø533 ØD                 DCR     C
Ø534 CA3EØ5             JZ      LT      ;NEW LN NOT = OLD LN
Ø537 9Ø                 SUB     B       ;COUNT DIFFERENCE
Ø538 CA61Ø5             JZ      ZERO    ;LINE LENGTHS EQUAL
Ø53B DA51Ø5             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 #
Ø53E 2A2B1Ø     LT:     LHLD    EOFP    ;END OF FILE ADDRESS
Ø541 54                 MOV     D H
Ø542 5D                 MOV     E.L
Ø543 CDA1Ø5             CALL    ADR     ;MOVE TO ADDRESS
Ø546 222B1Ø             SHLD    EOFP    ;NEW END OF FILE ADDRESS
Ø549 ØEØ2               MVI     C,2
Ø54B CDAFØ5             CALL    RMOV    ;OPEN UP FILE AREA
Ø54E C361Ø5             JMP     ZERO
                ; GET HERE IF # OF CHARS IN OLD LINE < # OF CHARS IN
                ; NEW LINE.
Ø551 2F         GT:     CMA
Ø552 3C                 INR     A       ;COUNT DIFFERENCE
Ø553 54                 MOV     D H
Ø554 5D                 MOV     E,L
Ø555 CDA1Ø5             CALL    ADR
Ø558 EB                 XCHG
Ø559 CDA6Ø5             CALL    LMOV    ;DELETE EXCESS CHAR IN FILE
Ø55C 36Ø1               MVI     M 1     ;E-O-F INDICATOR
Ø55E 222B1Ø             SHLD    EOFP    ;E-O-F ADDRESS
                ; GET HERE TO INSERT CURRENT LINE INTO FILE AREA
Ø561 2A721Ø     ZERO:   LHLD    INSP    ;INSERT ADDRESS
```

```
Ø564 36ØD               MVI     M,ASCR
Ø566 23                 INX     H
Ø567 11C51Ø             LXI     D,IBUF-1        ;NEW LINE ADDRESS
Ø56A ØEØ1               MVI     C,1     ;CHECK VALUE
Ø56C CDA6Ø5             CALL    LMOV    ;PLACE LINE IN FILE
Ø56F C367ØØ             JMP     EOR
                ;
                ;
                ; THIS ROUTINE IS USED TO FIND A LN IN THE FILE AREA
                ; WHICH IS GREATER THAN OR EQUAL TO THE CURRENT LINE #
Ø572 21811Ø     FIND:   LXI     H,ABUF+3        ;BUFFER ADDRESS
Ø575 22741Ø             SHLD    ADDS    ;SAVE ADDRESS
Ø578 2A791Ø     FIN1:   LHLD    BOFP    ;BEGIN FILE ADDRESS
Ø57B 7C                 MOV     A,H     ;RETURN TO MONITOR IF
Ø57C B5                 ORA     L       ;  FILE IS EMPTY...
Ø57D CA67ØØ             JZ      EOR
Ø58Ø CD9AØ5     FI1:    CALL    EO1     ;CHECK FOR END OF FILE
Ø583 EB                 XCHG
Ø584 2A741Ø             LHLD    ADDS    ;FETCH FIND ADDRESS
Ø587 EB                 XCHG
Ø588 3EØ4               MVI     A,4
Ø58A CDA1Ø5             CALL    ADR     ;BUMP LINE ADDRESS
Ø58D CDC8Ø5             CALL    COMØ    ;COMPARE LINE NUMBERS
Ø59Ø D8                 RC
Ø591 C8                 RZ
Ø592 7E         FI2:    MOV     A,M
Ø593 CDA1Ø5             CALL    ADR     ;NEXT LINE ADDRESS
Ø596 C38ØØ5             JMP     FI1
                ;
                ;
                ; WHEN SEARCHING THROUGH THE FILE AREA  THIS ROUTINE
                ; CHECKS TO SEE IF THE CURRENT ADDRESS IS THE END OF
                ; FILE
                ;
Ø599 23         EOF:    INX     H
Ø59A 3EØ1       EO1:    MVI     A,1     ;E-O-F INDICATOR
Ø59C BE                 CMP     M
Ø59D CØ                 RNZ
Ø59E C367ØØ             JMP     EOR
                ;
                ;
                ; THIS ROUTINE IS USED TO ADD A VALUE TO AN ADDRESS
                ; CONTAINED IN REGISTER H L
                ;
Ø5A1 85         ADR:    ADD     L
Ø5A2 6F                 MOV     L,A
Ø5A3 DØ                 RNC
Ø5A4 24                 INR     H
Ø5A5 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.
                ;
```

```
05A6 1A         LMOV:   LDAX    D       ;FETCH CHARACTER
05A7 13                 INX     D       ;INCREMENT FETCH ADDRESS
05A8 B9                 CMP     C       ;TERMINATION CHARACTER
05A9 C8                 RZ
05AA 77                 MOV     M,A     ;STORE CHARACTER
05AB 23                 INX     H       ;INCREMENT STORE ADDRESS
05AC C3A605             JMP     LMOV
                ;
                ;
                ; THIS ROUTINE IS SIMILAR TO ABOVE EXCEPT THAT THE
                ; CHARACTER ADDRESS IS DECREMENTED AFTER EACH FETCH
                ; AND STORE
                ;
05AF 1A         RMOV:   LDAX    D       ;FETCH CHARACTER
05B0 1B                 DCX     D       ;DECREMENT FETCH ADDRESS
05B1 B9                 CMP     C       ;TERMINATION CHARACTER
05B2 C8                 RZ
05B3 77                 MOV     M,A     ;STORE CHARACTER
05B4 2B                 DCX     H       ;DECREMENT STORE ADDRESS
05B5 C3AF05             JMP     RMOV
                ;
                ;
                ; THIS ROUTINE IS USED TO LOAD FOUR CHARACTERS FROM
                ; MEMORY INTO REGISTERS
                ;
05B8 46         LODM:   MOV     B,M     ;FETCH CHARACTER
05B9 23                 INX     H
05BA 4E                 MOV     C,M     ;FETCH CHARACTER
05BB 23                 INX     H
05BC 56                 MOV     D,M     ;FETCH CHARACTER
05BD 23                 INX     H
05BE 5E                 MOV     E,M     ;FETCH CHARACTER
05BF C9                 RET
                ;
                ;
                ; THIS ROUTINE STORES FOUR CHARACTERS FROM THE REGISTERS
                ; INTO MEMORY
                ;
05C0 73         STOM:   MOV     M,E     ;STORE CHARACTER
05C1 2B                 DCX     H
05C2 72                 MOV     M,D     ;STORE CHARACTER
05C3 2B                 DCX     H
05C4 71                 MOV     M,C     ;STORE CHARACTER
05C5 2B                 DCX     H
05C6 70                 MOV     M,B     ;STORE CHARACTER
05C7 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 ADDRESSED
                ; BY D E WAS GREATER THAN OR EQUAL TO CHARACTER STRING
                ; ADDRESSED BY H,L
                ;
05C8 0601       COM0:   MVI     B,1     ;EQUAL COUNTER
05CA 0E04               MVI     C,4     ;STRING LENGTH
05CC B7                 ORA     A       ;CLEAR CARRY
05CD 1A         CO1:    LDAX    D       ;FETCH CHARACTER
```

```
Ø5CE 9E                 SBB     M           ;COMPARE CHARACTERS
Ø5CF CAD3Ø5             JZ      CO2
Ø5D2 Ø4                 INR     B           ;INCREMENT EQUAL COUNTER
Ø5D3 1B         CO2:    DCX     D
Ø5D4 2B                 DCX     H
Ø5D5 ØD                 DCR     C
Ø5D6 C2CDØ5             JNZ     CO1
Ø5D9 Ø5                 DCR     B
Ø5DA C9                 RET
                ;
                ; THIS ROUTINE IS SIMILAR TO THE ABOVE ROUTINE EXCEPT ON
                ; RETURN CARRY FLAG = Ø MEANS THAT CHARACTER STRING
                ; ADDRESSED BY D,E IS ONLY > STRING ADDRESSED BY H L
                ;
Ø5DB ØEØ4       COM1:   MVI     C 4         ;STRING LENGTH
Ø5DD 1A                 LDAX    D           ;TCH CHARACTER
Ø5DE D6Ø1               SUI     1
Ø5EØ C3CEØ5             JMP     CO1+1
                ;
                ;
                ; THIS ROUTINE WILL TAKE ASCII CHARACTERS AND ADD ANY
                ; NECESSARY ASCII ZEROES, SO THE RESULT IS A 4 CHARACTER
                ; ASCII VALUE
                ;
Ø5E3 CDB8Ø5     NORM:   CALL    LODM        ;LOAD CHARACTERS
Ø5E6 AF                 XRA     A           ;FETCH A ZERO
Ø5E7 B8                 CMP     B
Ø5E8 C8                 RZ
Ø5E9 BB         NOR1:   CMP     E
Ø5EA C4CØØ5             CNZ     STOM        ;STORE VALUES
Ø5ED CØ                 RNZ
Ø5EE 5A                 MOV     E,D         ;NORMALIZE VALUE
Ø5EF 51                 MOV     D,C
Ø5FØ 48                 MOV     C B
Ø5F1 Ø63Ø               MVI     B,'Ø'
Ø5F3 C3E9Ø5             JMP     NOR1
                ;
                ;
                ; THIS ROUTINE IS USED TO LIST THE CONTENTS OF THE FILE
                ; AREA STARTING AT THE LINE NUMBER GIVEN IN THE COMMAND
                ;
Ø5F6 CD3AØ1     LIST:   CALL    CRLF
Ø5F9 CD72Ø5             CALL    FIND        ;FIND STARTING LN
Ø5FC 23         LISTØ:  INX     H           ;OUTPUT LINE.
Ø5FD CDAØØ2             CALL    SCRN
Ø6ØØ CD3AØ1             CALL    CRLF
Ø6Ø3 CD99Ø5             CALL    EOF         ;CHECK FOR END OF FILE
Ø6Ø6 CDEDØØ             CALL    INK         ;CHECK FOR CTRL-X
Ø6Ø9 C2FCØ5             JNZ     LISTØ       ;LOOP IF NO CTRL-X
Ø6ØC 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.
                ;
Ø6ØD CD27Ø3     DELL:   CALL    VCHK        ;CHECK FOR PARAMETER
Ø61Ø CD72Ø5             CALL    FIND        ;FIND LINE IN FILE AREA
```

```
Ø613 22721Ø             SHLD    DELP     ;SAVE DELETE POSITION
Ø616 21851Ø             LXI     H ABUF-7
Ø619 7E                 MOV     A.M      ;CHECK FOR 2ND PARAMETER
Ø61A B7                 ORA     A        ;SET FLAGS
Ø61B C221Ø6             JNZ     DEL1
Ø61E 21811Ø             LXI     H ABUF-3          ;USE FIRST PARAMETER
Ø621 22741Ø     DEL1:   SHLD    ADDS     ;SAVE FIND ADDRESS
Ø624 EB                 XCHG
Ø625 21301Ø             LXI     H MAXL-3
Ø628 CDC8Ø5             CALL    COMØ     ;COMPARE LINE NUMBERS
Ø62B 2A721Ø             LHLD    DELP     ;LOAD DELETE POSITION
Ø62E DA6FØ6             JC      NOVR
                ; GET HERE IF DELETION INVOLVES END OF FILE
Ø631 22231Ø             SHLD    EOFP     ;CHANGE E-O-F POSITION
Ø634 36Ø1               MVI     M.1      ;SET E-O-F INDICATOR
Ø636 EB                 XCHG
Ø637 2A291Ø             LHLD    BOFP     ;GET BEGIN FILE ADDRESS
Ø63A EB                 XCHG
Ø63B Ø6ØD               MVI     B 13     ;SET SCAN SWITCH
Ø63D 2B                 DCX     H        ;CHECK FOR BOF
Ø63E 7D         DEL2:   MOV     A.L
Ø63F 93                 SUB     E
Ø64Ø 7C                 MOV     A.H
Ø641 9A                 SBB     D
Ø642 3EØD               MVI     A.ASCR   ;LOOK FOR CR
Ø644 DA66Ø6             JC      DEL4     ;DECREMENTED PAST BOF
Ø647 Ø5                 DCR     B
Ø648 2B                 DCX     H
Ø649 BE                 CMP     M        ;FIND NEW MAX LN
Ø64A C23EØ6             JNZ     DEL2
Ø64D 2B                 DCX     H
Ø64E 7D                 MOV     A.L
Ø64F 93                 SUB     E
Ø65Ø 7C                 MOV     A.H
Ø651 9A                 SBB     D
Ø652 DA67Ø6             JC      DEL5
Ø655 BE                 CMP     M        ;END OF PREVIOUS LINE
Ø656 23                 INX     H
Ø657 23                 INX     H
Ø658 CA5CØ6             JZ      DEL3
Ø65B 23                 INX     H
Ø65C CDB8Ø5     DEL3:   CALL    LODM     ;LOAD NEW MAX LN
Ø65F 21301Ø             LXI     H.MAXL+3          ;SET ADDRESS
Ø662 CDCØØ5             CALL    STOM     ;STORE NEW MAX LN
Ø665 C9                 RET
Ø666 B8         DEL4:   CMP     B        ;CHECK SWITCH
Ø667 EB         DEL5:   XCHG
Ø668 C25BØ6             JNZ     DEL3-1
Ø66B 322D1Ø             STA     MAXL     ;MAKE MAX LN A SMALL NUMBER
Ø66E C9                 RET
                ; GET HERE IF DELETION IS IN MIDDLE OF FILE AREA
Ø66F CD8ØØ5     NOVR:   CALL    FI1      ;FIND END OF DELETE AREA
Ø672 CC92Ø5             CZ      FI2      ;NEXT LINE IF THIS LN EQUAL
Ø675 EB         NOV1:   XCHG
Ø676 2A721Ø             LHLD    DELP     ;CHAR MOVE TO POSITION
Ø679 ØEØ1               MVI     C.1      ;MOVE TERMINATOR
Ø67B CDA6Ø5             CALL    LMOV     ;COMPACT FILE AREA
Ø67E 222B1Ø             SHLD    EOFP     ;SET EOF POSITION
```

```
0681 36Ø1              MVI     M,1      ;SET EOF INDICATOR
0683 C9                RET
                ;
                ; STARTING HERE IS THE SELF ASSEMBLER PROGRAM
                ; THIS PROGRAM ASSEMBLES PROGRAMS WHICH ARE
                ; IN THE FILE AREA
                ;
Ø684 CD27Ø3     ASSM:  CALL    VCHK     ;CHECK FOR PARAMETER
Ø687 3A821Ø            LDA     ABUF-4   ;GET 2ND PARAMETER
Ø68A B7                ORA     A        ;CHECK FOR PARAMETERS
Ø68B C294Ø6            JNZ     ASM4
Ø68E 2A8A1Ø            LHLD    BBUF     ;FETCH 1ST PARAMETER
Ø691 228C1Ø            SHLD    BBUF+2   ;STORE INTO 2ND PARAMETER
Ø694 3ACA1Ø     ASM4:  LDA     IBUF-4   ;FETCH INPUT CHARACTER
Ø697 D645              SUI     'E'      ;RESET A IF ERRORS ONLY
Ø699 328E1Ø            STA     AERR     ;SAVE ERROR FLAG
Ø69C 3EFF              MVI     A,ØFFH   ;PUT MARKER IN SYMBOL TABLE...
Ø69E 321A11            STA     SYMT
Ø6A1 AF                XRA     A        ;GET A ZERO
Ø6A2 32941Ø     ASM3:  STA     PASI     ;SET PASS INDICATOR
Ø6A5 CD3AØ1            CALL    CRLF     ;INDICATE START OF PASS
Ø6A8 2A8A1Ø            LHLD    BBUF     ;FETCH ORIGIN
Ø6AB 22921Ø            SHLD    ASPC     ;INITIALIZE PC
Ø6AE 2A291Ø            LHLD    BOFP     ;GET START OF FILE
Ø6B1 22721Ø            SHLD    APNT
Ø6B4 2A721Ø     ASM1:  LHLD    APNT     ;FETCH LINE POINTER
Ø6B7 31B11Ø            LXI     SP,AREA+18
Ø6BA 7E                MOV     A,M      ;FETCH CHARACTER
Ø6BB FEØ1              CPI     1        ;END OF FILE?
Ø6BD CA2DØ9            JZ      EASS     ;JUMP IF END OF FILE
Ø6CØ EB                XCHG
Ø6C1 13                INX     D        ;INCREMENT ADDRESS
Ø6C2 21B11Ø            LXI     H,OBUF   ;BLANK START ADDRESS
Ø6C5 3EC1              MVI     A,IBUF-5 AND ØFFH ;BLANK END ADDRESS
Ø6C7 CDE5ØØ            CALL    CLER     ;BLANK OUT BUFFER
Ø6CA ØEØD              MVI     C,ASCR   ;STOP CHARACTER
Ø6CC CDA6Ø5            CALL    LMOV     ;MOVE LINE INTO BUFFER
Ø6CF 71                MOV     M,C      ;PLACE CR IN BUFFER
Ø6DØ EB                XCHG
Ø6D1 22721Ø            SHLD    APNT     ;SAVE ADDRESS
Ø6D4 3A941Ø            LDA     PASI     ;FETCH PASS INDICATOR
Ø6D7 B7                ORA     A        ;SET FLAGS
Ø6D8 C2E1Ø6            JNZ     ASM2     ;JUMP IF PASS 2
Ø6DB CDØØØ7            CALL    PAS1
Ø6DE C3B4Ø6            JMP     ASM1
Ø6E1 CDC9Ø7     ASM2:  CALL    PAS2
Ø6E4 CDEAØ6            CALL    AOUT     ;OUTPUT LINE
Ø6E7 C3B4Ø6            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.
                ;
Ø6EA 3A8E1Ø     AOUT:  LDA     AERR     ;FETCH ERROR SWITCH
Ø6ED B7                ORA     A        ;SET FLAGS
Ø6EE C2F7Ø6            JNZ     AOU1     ;OUTPUT ALL LINES
Ø6F1 3AB11Ø            LDA     OBUF     ;FETCH ERROR INDICATOR
```

```
Ø6F4 FE2Ø               CPI     ' '     ;CHECK FOR AN ERROR
Ø6F6 C8                 RZ              ;RETURN IF NO ERROR
Ø6F7 21B11Ø     AOU1:   LXI     H,OBUF  ;OUTPUT BUFFER ADDRESS
Ø6FA CDAØØ2             CALL    SCRN    ;OUTPUT LINE. .
Ø6FD C33AØ1             JMP     CRLF
                ;
                ; PASS1 OF ASSEMBLER. USED TO FORM SYMBOL TABLE
                ;
Ø7ØØ CD93Ø1     PAS1:   CALL    ZBUF    ;CLEAR BUFFER
Ø7Ø3 32941Ø             STA     PASI    ;SET FOR PASS1
Ø7Ø6 21C61Ø             LXI     H,IBUF  ;INITIALIZE LINE POINTER
Ø7Ø9 CD61Ø7             CALL    PATCH   ;CHECK FOR LABEL OR COMMENT
                ;
                ; PROCESS LABEL
                ;
Ø7ØC CD5ØØB             CALL    SLAB    ;GET AND CHECK LABEL
Ø7ØF DAØFØB             JC      OP5     ;ERROR IN LABEL
Ø712 CAEAØC             JZ      ERRD    ;DUPLICATE LABEL
Ø715 CD4FØ7             CALL    LCHK    ;CHECK CHARACTER AFTER LABEL
Ø718 C2ØFØB             JNZ     OP5     ;ERROR IF NO BLANK
Ø71B ØEØ5               MVI     C,LLAB  ;LENGTH OF LABELS
Ø71D 217E1Ø             LXI     H,ABUF  ;SET BUFFER ADDRESS
Ø72Ø 7E         MLAB:   MOV     A,M     ;FETCH NEXT CHARACTER
Ø721 12                 STAX    D       ;STORE IN SYMBOL TABLE
Ø722 13                 INX     D
Ø723 23                 INX     H
Ø724 ØD                 DCR     C
Ø725 C22ØØ7             JNZ     MLAB
Ø728 EB                 XCHG
Ø729 22901Ø             SHLD    TABA    ;SAVE TABLE ADDRESS FOR EQU
Ø72C 3A931Ø             LDA     ASPC-1  ;FETCH PC HIGH
Ø72F 77                 MOV     M,A
Ø73Ø 23                 INX     H
Ø731 3A921Ø             LDA     ASPC    ;FETCH PC LOW
Ø734 77                 MOV     M,A     ;STORE IN TABLE
Ø735 23                 INX     H       ;STORE END-OF-TABLE MARKER ..
Ø736 36FF               MVI     M,ØFFH
                ;
                ; PROCESS OPCODE
                ;
Ø738 CD93Ø1     OPC:    CALL    ZBUF    ;ZERO WORKING BUFFER
Ø73B CD39Ø9             CALL    SBLK    ;SCAN TO OPCODE
Ø73E DA36ØB             JC      OERR    ;FOUND CARRIAGE RETURN
Ø741 CD9AØB             CALL    ALPS    ;PLACE OPCODE IN BUFFER
Ø744 FE2Ø               CPI     ' '     ;CHECK FOR BLANK AFTER OPCODE
Ø746 DA95ØA             JC      OPCD    ;CR OR TAB AFTER OPCODE
Ø749 C236ØB             JNZ     OERR    ;ERROR IF NO BLANK
Ø74C C395ØA             JMP     OPCD    ;CHECK OPCODE
                ;
                ; THIS ROUTINE CHECKS THE CHARACTER AFTER A LABEL
                ; FOR A BLANK OR A COLON.
                ;
Ø74F 2A961Ø     LCHK:   LHLD    PNTR
Ø752 7E                 MOV     A,M     ;GET CHARACTER AFTER LABEL
Ø753 FE2Ø               CPI     ' '     ;CHECK FOR A BLANK
Ø755 C8                 RZ              ;RETURN IF A BLANK
Ø756 FEØ9               CPI     Ø9H     ;CHECK FOR A TAB
Ø758 C8                 RZ              ;RETURN IF A TAB
```

```
Ø759 FE3A               CPI     ':'     ;CHECK FOR A COLON
Ø75B CØ                 RNZ
Ø75C 23                 INX     H
Ø75D 22961Ø             SHLD    PNTR    ;SAVE POINTER
Ø76Ø C9                 RET
                ;
                ; CHECK FOR LABELS OR COMMENTS
                ;
Ø761 33         PATCH:  INX     SP      ;BUMP SP PAST RETURN ADDRESS ..
Ø762 33                 INX     SP
Ø763 22961Ø             SHLD    PNTR    ;SAVE POINTER
Ø766 7E                 MOV     A,M     ;FETCH CHARACTER
Ø767 FE2Ø               CPI     ' '     ;CHECK FOR A BLANK
Ø769 CA38Ø7             JZ      OPC     ;JUMP IF NO LABEL
Ø76C FEØ9               CPI     Ø9H     ;CHECK FOR A TAB
Ø76E CA38Ø7             JZ      OPC     ;JUMP IF NO LABEL
Ø771 FE2A               CPI     '*'     ;CHECK FOR COMMENT
Ø773 C8                 RZ              ;RETURN TO HIGHER LEVEL
Ø774 FE3B               CPI     ';'     ;ALSO CHECK FOR COMMENT...
Ø776 C8                 RZ
Ø777 3B                 DCX     SP      ;POINT SP AT IMMEDIATE RETURN...
Ø778 3B                 DCX     SP
Ø779 C9                 RET
                ;
                ; PROCESS ANY PSEUDO OPS THAT NEED TO BE IN PASS 1
                ;
Ø77A CD39Ø9     PSU1:   CALL    SBLK    ;SCAN TO OPERAND
Ø77D 1A                 LDAX    D       ;FETCH VALUE
Ø77E B7                 ORA     A       ;SET FLAGS
Ø77F CA96Ø7             JZ      ORG1    ;ORG OPCODE
Ø782 FAC6Ø7             JM      DAT1    ;DATA STATEMENT
Ø785 E2ABØ7             JPO     EQU1    ;EQU OPCODE
Ø788 FEØ5               CPI     5
Ø78A DABEØ7             JC      RES1    ;RES OPCODE
Ø78D C22DØ9             JNZ     EASS    ;JUMP IF END
                ; DO DW PSEUDO/OP
Ø79Ø ØEØ2       ACO1:   MVI     C,2     ;2 BYTE INSTRUCTION
Ø792 AF                 XRA     A       ;GET A ZERO
Ø793 C325ØB             JMP     OCN1    ;ADD VALUE TO PROGRAM CNTR
                ; DO ORG PSEUDO-OP
Ø796 CDBCØB     ORG1:   CALL    ASCN    ;GET OPERAND
Ø799 3AB11Ø             LDA     OBUF    ;FETCH ERROR INDICATOR
Ø79C FE2Ø               CPI     ' '     ;CHECK FOR AN ERROR
Ø79E CØ                 RNZ
Ø79F 22921Ø             SHLD    ASPC    ;STORE NEW ORIGIN
Ø7A2 3AC61Ø             LDA     IBUF    ;GET FIRST CHARACTER
Ø7A5 FE21               CPI     ' '+1   ;CHECK FOR LABEL
Ø7A7 D8                 RC              ;NO LABEL
Ø7A8 C3B6Ø7             JMP     EQUS    ;CHANGE LABEL VALUE
                ; DO EQU PSEUDO-OP
Ø7AB CDBCØB     EQU1:   CALL    ASCN    ;GET OPERAND
Ø7AE 3AC61Ø             LDA     IBUF    ;FETCH 1ST CHARACTER
Ø7B1 FE21               CPI     ' '-1   ;CHECK FOR LABEL
Ø7B3 DAC3ØC             JC      ERRM    ;MISSING LABEL
Ø7B6 EB         EQUS:   XCHG
Ø7B7 2A9Ø1Ø             LHLD    TABA    ;SYMBOL TABLE ADDRESS
Ø7BA 72                 MOV     M,D     ;STORE LABEL VALUE
Ø7BB 23                 INX     H
```

```
Ø7BC 73                 MOV     M,E
Ø7BD C9                 RET
               ; DO DS PSEUDO-OP
Ø7BE CDBCØB    RES1:    CALL    ASCN     ;GET OPERAND
Ø7C1 44                 MOV     B,H
Ø7C2 4D                 MOV     C,L
Ø7C3 C31AØ8             JMP     RES21    ;ADD VALUE TO PROGRAM COUNTER
               ;
               ; DO DB PSEUDO-OP
               ;
Ø7C6 C321Ø8    DAT1:    JMP     DAT2A
               ;
               ; PERFORM PASS 2 OF THE ASSEMBLER
               ;
Ø7C9 21B31Ø    PAS2:    LXI     H,OBUF+2 ;SET OUTPUT BUFFER ADDRESS
Ø7CC 3A931Ø             LDA     ASPC+1   ;FETCH PC HIGH
Ø7CF CDAFØ2             CALL    BINH+3   ;CONVERT FOR OUTPUT
Ø7D2 23                 INX     H
Ø7D3 3A921Ø             LDA     ASPC     ;FETCH PC LOW
Ø7D6 CDAFØ2             CALL    BINH+3   ;CONVERT FOR OUTPUT
Ø7D9 23                 INX     H
Ø7DA 229D1Ø             SHLD    OIND     ;SAVE OUTPUT ADDRESS
Ø7DD CD93Ø1             CALL    ZBUF     ;CLEAR BUFFER
Ø7EØ 21C61Ø             LXI     H,IBUF   ;INITIALIZE LINE POINTER
Ø7E3 CD61Ø7    PABL:    CALL    PATCH
Ø7E6 CD5ØØB             CALL    SLAB     ;SCAN OFF LABEL
Ø7E9 DAE5ØC             JC      ERRL     ;ERROR IN LABEL
Ø7EC CD4FØ7             CALL    LCHK     ;CHECK FOR A BLANK OR COLON
Ø7EF C2E5ØC             JNZ     ERRL     ;ERROR IF NOT A BLANK
Ø7F2 C338Ø7             JMP     OPC
               ;
               ;
               ;
               ; PROCESS PSEUDO OPS FOR PASS2
Ø7F5 1A        PSU2:    LDAX    D
Ø7F6 B7                 ORA     A        ;SET FLAGS
Ø7F7 CA39Ø8             JZ      ORG2     ;ORG OPCODE
Ø7FA FA1EØ8             JM      DAT2     ;DATA OPCODE
Ø7FD E227Ø8             JPO     EQU2     ;EQUATE PSEUDE-OP
Ø8ØØ FEØ5               CPI     5
Ø8Ø2 DAØEØ8             JC      RES2     ;RES OPCODE
Ø8Ø5 C22DØ9             JNZ     EASS     ;END OPCODE
               ; DO DW PSEUDO-OP
Ø8Ø8 CDØDØ9    ACO2:    CALL    TYS6     ;GET VALUE
Ø8ØB C39ØØ7             JMP     ACO1
               ; DO DS PSEUDO-OP
Ø8ØE CDB9ØB    RES2:    CALL    ASBL     ;GET OPERAND
Ø811 44                 MOV     B,H
Ø812 4D                 MOV     C,L
Ø813 2A8C1Ø             LHLD    BBUF+2   ;FETCH STORAGE COUNTER
Ø816 Ø9                 DAD     B        ;ADD VALUE
Ø817 228C1Ø             SHLD    BBUF+2
Ø81A AF        RES21:   XRA     A        ;GET A ZERO
Ø81B C328ØB             JMP     OCN2
               ; DO DB PSEUDO-OP
Ø81E CDCCØ8    DAT2:    CALL    TYS5     ;GET OPERAND
Ø821 AF        DAT2A:   XRA     A        ;MAKE A ZERO
Ø822 ØEØ1               MVI     C,1      ;BYTE COUNT
```

```
Ø824 C325ØB             JMP     OCN1
                ;
                ; HANDLE EQUATES ON 2ND PASS.
                ;
Ø827 CDB9ØB     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
                ;
Ø82A EB         BINAD:  XCHG            ;PUT VALUE INTO DE
Ø82B 21B31Ø             LXI     H,OBUF-2 ;POINTER TO ADDR IN OBUF
Ø82E 7A                 MOV     A,D     ;STORE HI BYTE....
Ø82F CDAFØ2             CALL    BINH-3
Ø832 23                 INX     H
Ø833 7B                 MOV     A,E     ;STORE LO BYTE...
Ø834 CDAFØ2             CALL    BINH-3
Ø837 23                 INX     H
Ø838 C9                 RET
                ; DO ORG PSEUDO-OP
Ø839 CDB9ØB     ORG2:   CALL    ASBL    ;GET NEW ORIGIN
Ø83C 3AB11Ø             LDA     OBUF    ;GET ERROR INDICATOR
Ø83F FE2Ø               CPI     ' '     ;CHECK FOR AN ERROR
Ø841 CØ                 RNZ             ;DON'T MODIFY PC IF ERROR
Ø842 CD2AØ8             CALL    BINAD   ;STORE NEW ADDR IN OBUF
Ø845 2A921Ø             LHLD    ASPC    ;FETCH PC
Ø848 EB                 XCHG
Ø849 22921Ø             SHLD    ASPC    ;STORE NEW PC
Ø84C 7D                 MOV     A,L
Ø84D 93                 SUB     E       ;FORM DIFFERENCE OF ORIGINS
Ø84E 5F                 MOV     E,A
Ø84F 7C                 MOV     A,H
Ø85Ø 9A                 SBB     D
Ø851 57                 MOV     D,A
Ø852 2A8C1Ø             LHLD    BBUF-2  ;FETCH STORAGE POINTER
Ø855 19                 DAD     D       ;MODIFY
Ø856 228C1Ø             SHLD    BBUF-2  ;SAVE
Ø859 C9                 RET
                ;
                ; PROCESS 1 BYTE INSTRUCTIONS WITHOUT OPERANDS
                ;
Ø85A C31AØ9     TYP1:   JMP     ASTO    ;STORE VALUE IN MEMORY AND RETURN
                ;
                ; PROCESS STAX AND LDAX INSTRUCTIONS
                ;
Ø85D CDB9ØB     TYP2:   CALL    ASBL    ;FETCH OPERAND
Ø86Ø C4A5ØC             CNZ     ERRR    ;ILLEGAL REGISTER
Ø863 7D                 MOV     A,L     ;GET LOW ORDER OPERAND
Ø864 B7                 ORA     A       ;SET FLAGS
Ø865 CA81Ø8             JZ      TY31    ;OPERAND = Ø
Ø868 FEØ2               CPI     2       ;OPERAND = 2
Ø86A C4A5ØC             CNZ     ERRR    ;ILLEGAL REGISTER
Ø86D C381Ø8             JMP     TY31
                ;
                ; PROCESS PUSH,POP,INX,DCX,DAD INSTRUCTIONS
                ;
Ø87Ø CDB9ØB     TYP3:   CALL    ASBL    ;FETCH OPERAND
Ø873 C4A5ØC             CNZ     ERRR    ;ILLEGAL REGISTER
```

```
Ø876 7D                 MOV     A L      ;GET LOW ORDER OPERAND
Ø877 ØF                 RRC              ;CHECK LOW ORDER BIT
Ø878 DCA5ØC             CC      ERRR     ;ILLEGAL REGISTER
Ø87B 17                 RAL              ;RESTORE
Ø87C FEØ8               CPI     8
Ø87E D4A5ØC             CNC     ERRR     ;ILLEGAL REGISTER
Ø881 Ø7         TY31:   RLC              ;MULTIPLY BY 8
Ø882 17                 RAL
Ø883 17                 RAL
Ø884 47         TY32:   MOV     B,A
Ø885 1A                 LDAX    D        ;FETCH OPCODE BASE
Ø886 8Ø                 ADD     B        ;FORM OPCODE
Ø887 FE76               CPI     118      ;CHECK FOR MOV M,M
Ø889 CCA5ØC             CZ      ERRR     ;ILLEGAL REGISTER
Ø88C C35AØ8             JMP     TYP1
                ;
                ; PROCESS ACCUMULATOR  INR DCR,MOV.RST INSTRUCTIONS
                ;
Ø88F CDB9ØB     TYP4:   CALL    ASBI     ;FETCH OPERAND
Ø892 C4A5ØC             CNZ     ERRR     ;ILLEGAL REGISTER
Ø895 7D                 MOV     A,L      ;GET LOW ORDER OPERAND
Ø896 FEØ8               CPI     8
Ø898 D4A5ØC             CNC     ERRR     ;ILLEGAL REGISTER
Ø89B 1A                 LDAX    D        ;FETCH OPCODE BASE
Ø89C FE4Ø               CPI     64       ;CHECK FOR MOV INSTRUCTION
Ø89E CAADØ8             JZ      TY41
Ø8A1 FEC7               CPI     199
Ø8A3 7D                 MOV     A.L
Ø8A4 CA81Ø8             JZ      TY31     ;RST INSRUCTION
Ø8A7 FA84Ø8             JM      TY32     ;ACCUMULATOR INSTRUCTION
Ø8AA C381Ø8             JMP     TY31     ;INR.DCR
                ; PROCESS MOV INSTRUCTION
Ø8AD 29         TY41:   DAD     H        ;MULTIPLY OPERAND BY 8
Ø8AE 29                 DAD     H
Ø8AF 29                 DAD     H
Ø8BØ 85                 ADD     L        ;FORM OPCODE
Ø8B1 12                 STAX    D        ;SAVE OPCODE
Ø8B2 CDEBØ8             CALL    MPNT
Ø8B5 CDBCØB             CALL    ASCN
Ø8B8 C4A5ØC             CNZ     ERRR     ;INCREMENT POINTER
Ø8BB 7D                 MOV     A.L      ;FETCH LOW ORDER OPERAND
Ø8BC FEØ8               CPI     8
Ø8BE D4A5ØC             CNC     ERRR     ;ILLEGAL REGISTER
Ø8C1 C384Ø8             JMP     TY32
                ;
                ; PROCESS IMMEDIATE INSTRUCTIONS
                ; IMMEDIATE BYTE CAN BETWEEN -256 AND 255
                ; MVI INSTRUCTION IS A SPECIAL CASE AND CONTAINS
                ; 2 ARGUMENTS IN OPERAND
Ø8C4 FEØ6       TYP5:   CPI     6        ;CHECK FOR MVI INSTRUCTION
Ø8C6 CCD9Ø8             CZ      TY56
Ø8C9 CD1AØ9             CALL    ASTO     ;STORE OBJECT BYTE
Ø8CC CDB9ØB     TYS5:   CALL    ASBL     ;GET IMMEDIATE ARGUMENT
Ø8CF 3C                 INR     A
Ø8DØ FEØ2               CPI     2        ;CHECK OPERAND FOR RANGE
Ø8D2 D4BEØC             CNC     ERRV     ;OPERAND OUT OF RANGE
Ø8D5 7D                 MOV     A.L
Ø8D6 C35AØ8             JMP     TYP1
```

```
                    ;
                    ; FETCH 1ST ARG FOR MVI AND LXI INSTRUCTIONS
                    ;
Ø8D9 CDB9ØB         TY56:   CALL    ASBL    ;FETCH ARG
Ø8DC C4A5ØC                 CNZ     ERRR    ;ILLEGAL REGISTER
Ø8DF 7D                     MOV     A.L     ;GET LOW ORDER ARGUMENT
Ø8EØ FEØ8                   CPI     8
Ø8E2 D4A5ØC                 CNC     ERRR    ;ILLEGAL REGISTER
Ø8E5 29                     DAD     H
Ø8E6 29                     DAD     H
Ø8E7 29                     DAD     H
Ø8E8 1A                     LDAX    D       ;FETCH OPCODE BASE
Ø8E9 85                     ADD     L       ;FOR OPCODE
Ø8EA 5F                     MOV     E A     ;SAVE OBJECT BYTE
Ø8EB 2A961Ø         MPNT:   LHLD    PNTR    ;FETCH POINTER
Ø8EE 7E                     MOV     A M     ;FETCH CHARACTER
Ø8EF FE2C                   CPI     ','     ;CHECK FOR COMMA
Ø8F1 23                     INX     H       ;INCREMENT POINTER
Ø8F2 22961Ø                 SHLD    PNTR
Ø8F5 C2AEØC                 JNZ     ERRS    ;SYNTAX ERROR IF NO COMMA
Ø8F8 7B                     MOV     A E
Ø8F9 C9                     RET
                    ;
                    ; PROCESS 3 BYTE INSTRUCTIONS
                    ; LXI INSTRUCTION IS A SPECIAL CASE
                    ;
Ø8FA FEØ1           TYP6:   CPI     1       ;CHECK FOR LXI INSTRUCTION
Ø8FC C2ØAØ9                 JNZ     TY6     ;JUMP IF NOT LXI
Ø8FF CDD9Ø8                 CALL    TY56    ;GET REGISTER
Ø9Ø2 E6Ø8                   ANI     Ø8H     ;CHECK FOR ILLEGAL REGISTER
Ø9Ø4 C4A5ØC                 CNZ     ERRR    ;REGISTER ERROR
Ø9Ø7 7B                     MOV     A.E     ;GET OPCODE
Ø9Ø8 E6F7                   ANI     ØF7H    ;CLEAR BIT IN ERROR
Ø9ØA CD1AØ9         TY6:    CALL    ASTO    ;STORE OBJECT BYTE
Ø9ØD CDB9ØB         TY56·   CALL    ASBL    ;FETCH OPERAND
Ø91Ø 7D                     MOV     A,L
Ø911 54                     MOV     D H
Ø912 CD1AØ9                 CALL    ASTO    ;STORE 2ND BYTE
Ø915 7A                     MOV     A.D
Ø916 C35AØ8                 JMP     TYP1
Ø919 C9                     RET
                    ;
                    ; THIS ROUTINE IS USED TO STORE OBJECT CODE PRODUCED
                    ; BY THE ASSEMBLER DURING PASS 2 INTO MEMORY
                    ;
Ø91A 2A8C1Ø         ASTO:   LHLD    BBUF 2  ;FETCH STORAGE ADDRESS
Ø91D 77                     MOV     M,A     ;STORE OBJECT BYTE
Ø91E 23                     INX     H       ;INCREMENT LOCATION
Ø91F 228C1Ø                 SHLD    BBUF+2
Ø922 2A9D1Ø                 LHLD    OIND    ;FETCH OUTPUT ADDRESS
Ø925 23                     INX     H
Ø926 CDAFØ2                 CALL    BINH-3  ;CONVERT OBJECT BYTE
Ø929 229D1Ø                 SHLD    OIND
Ø92C 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.
                    ;
Ø92D 3A941Ø         PASS:   LDA     PASI    ;FETCH PASS INDICATOR
Ø93Ø B7                     ORA     A       ;SET FLAGS
Ø931 C267ØØ                 JNZ     EOR     ;JUMP IF FINISHED
Ø934 3EØ1                   MVI     A,1     ;PASS INDICATOR FOR 2ND PASS
Ø936 C3A2Ø6                 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.
                    ;
Ø939 2A961Ø         SBLK:   LHLD    PNTR    ;FETCH ADDRESS
Ø93C 7E             SBL1:   MOV     A,M     ;FETCH CHARACTER
Ø93D FEØ9                   CPI     Ø9H     ;IS IT A TAB?
Ø93F CA45Ø9                 JZ      SBL2    ;TREAT LIKE A BLANK
Ø942 FE2Ø                   CPI     ' '     ;CHECK FOR A BLANK
Ø944 CØ                     RNZ             ;RETURN IF NON-BLANK
Ø945 23             SBL2:   INX     H       ;INCREMENT
Ø946 22961Ø                 SHLD    PNTR    ;SAVE POINTER
Ø949 C33CØ9                 JMP     SBL1
                    ;
                    ;
                    ; THIS ROUTINE IS USED TO CHECK THE CONDITION
                    ; CODE MNEMONICS FOR CONDITIONAL JUMPS, CALLS
                    ; AND RETURNS.
                    ;
Ø94C 217F1Ø         COND:   LXI     H,ABUF-1
Ø94F 22741Ø                 SHLD    ADDS
Ø952 Ø6Ø2                   MVI     B,2     ;2 CHARACTERS
Ø954 C38ØØA                 JMP     COPC
                    ;
                    ;
                    ; THE FOLLOWING IS THE OPCODE TABLE
                    ;
                    ;
Ø957 4F5247         OTAB:   DB      'ORG'
Ø95A ØØ                     DB      Ø
Ø95B ØØ                     DB      Ø
Ø95C 455155                 DB      'EQU'
Ø95F ØØ                     DB      Ø
Ø96Ø Ø1                     DB      1
Ø961 4442                   DB      'DB'
Ø963 ØØ                     DB      Ø
Ø964 ØØ                     DB      Ø
Ø965 FF                     DB      255
Ø966 4453                   DB      'DS'
Ø968 ØØ                     DB      Ø
Ø969 ØØ                     DB      Ø
Ø96A Ø3                     DB      3
Ø96B 4457                   DB      'DW'
Ø96D ØØ                     DB      Ø
Ø96E ØØ                     DB      Ø
Ø96F Ø5                     DB      5
Ø97Ø 454E44                 DB      'END'
Ø973 ØØ                     DB      Ø
Ø974 Ø6                     DB      6
```

```
Ø975 ØØ                 DB      Ø
Ø976 484C54             DB      'HLT'
Ø979 76                 DB      118
Ø97A 524C43             DB      'RLC'
Ø97D Ø7                 DB      7
Ø97E 525243             DB      'RRC'
Ø981 ØF                 DB      15
Ø982 52414C             DB      'RAL'
Ø985 17                 DB      23
Ø986 524152             DB      'RAR'
Ø989 1F                 DB      31
Ø98A 524554             DB      'RET'
Ø98D C9                 DB      2Ø1
Ø98E 434D41             DB      'CMA'
Ø991 2F                 DB      47
Ø992 535443             DB      'STC'
Ø995 37                 DB      55
Ø996 444141             DB      'DAA'
Ø999 27                 DB      39
Ø99A 434D43             DB      'CMC'
Ø99D 3F                 DB      63
Ø99E 4549               DB      'EI'
Ø9AØ ØØ                 DB      Ø
Ø9A1 FB                 DB      251
Ø9A2 4449               DB      'DI'
Ø9A4 ØØ                 DB      Ø
Ø9A5 F3                 DB      243
Ø9A6 4E4F5Ø             DB      'NOP'
Ø9A9 ØØ                 DB      Ø
Ø9AA ØØ                 DB      Ø
Ø9AB 58434847           DB      'XCHG'
Ø9AF EB                 DB      235
Ø9BØ 5854484C           DB      'XTHL'
Ø9B4 E3                 DB      227
Ø9B5 535Ø484C           DB      'SPHL'
Ø9B9 F9                 DB      249
Ø9BA 5Ø43484C           DB      'PCHL'
Ø9BE E9                 DB      233
Ø9BF ØØ                 DB      Ø
Ø9CØ 53544158           DB      'STAX'
Ø9C4 Ø2                 DB      2
Ø9C5 4C444158           DB      'LDAX'
Ø9C9 ØA                 DB      1Ø
Ø9CA ØØ                 DB      Ø
Ø9CB 5Ø555348           DB      'PUSH'
Ø9CF C5                 DB      197
Ø9DØ 5Ø4F5Ø             DB      'POP'
Ø9D3 ØØ                 DB      Ø
Ø9D4 C1                 DB      193
Ø9D5 494E58             DB      'INX'
Ø9D8 ØØ                 DB      Ø
Ø9D9 Ø3                 DB      3
Ø9DA 444358             DB      'DCX'
Ø9DD ØØ                 DB      Ø
Ø9DE ØB                 DB      11
Ø9DF 444144             DB      'DAD'
Ø9E2 ØØ                 DB      Ø
Ø9E3 Ø9                 DB      9
```

```
Ø9E4 ØØ                DB        Ø
Ø9E5 494E52            DB        'INR'
Ø9E8 Ø4                DB        4
Ø9E9 444352            DB        'DCR'
Ø9EC Ø5                DB        5
Ø9ED 4D4F56            DB        'MOV'
Ø9FØ 4Ø                DB        64
Ø9F1 414444            DB        'ADD'
Ø9F4 8Ø                DB        128
Ø9F5 414443            DB        'ADC'
Ø9F8 88                DB        136
Ø9F9 535542            DB        'SUB'
Ø9FC 9Ø                DB        144
Ø9FD 534242            DB        'SBB'
ØAØØ 98                DB        152
ØAØ1 414E41            DB        'ANA'
ØAØ4 AØ                DB        16Ø
ØAØ5 585241            DB        'XRA'
ØAØ8 A8                DB        168
ØAØ9 4F5241            DB        'ORA'
ØAØC BØ                DB        176
ØAØD 434D5Ø            DB        'CMP'
ØA1Ø B8                DB        184
ØA11 525354            DB        'RST'
ØA14 C7                DB        199
ØA15 ØØ                DB        Ø
ØA16 414449            DB        'ADI'
ØA19 C6                DB        198
ØA1A 414349            DB        'ACI'
ØA1D CE                DB        2Ø6
ØA1E 535549            DB        'SUI'
ØA21 D6                DB        214
ØA22 534249            DB        'SBI'
ØA25 DE                DB        222
ØA26 414E49            DB        'ANI'
ØA29 E6                DB        23Ø
ØA2A 585249            DB        'XRI'
ØA2D EE                DB        238
ØA2E 4F5249            DB        'ORI'
ØA31 F6                DB        246
ØA32 435Ø49            DB        'CPI'
ØA35 FE                DB        254
ØA36 494E              DB        'IN'
ØA38 ØØ                DB        Ø
ØA39 DB                DB        219
ØA3A 4F5554            DB        'OUT'
ØA3D D3                DB        211
ØA3E 4D5649            DB        'MVI'
ØA41 Ø6                DB        6
ØA42 ØØ                DB        Ø
ØA43 4A4D5Ø            DB        'JMP'
ØA46 ØØ                DB        Ø
ØA47 C3                DB        195
ØA48 43414C4C          DB        'CALL'
ØA4C CD                DB        2Ø5
ØA4D 4C5849            DB        'LXI'
ØA5Ø ØØ                DB        Ø
ØA51 Ø1                DB        1
```

```
0A52 4C4441             DB      'LDA'
0A55 00                 DB      0
0A56 3A                 DB      58
0A57 535441             DB      'STA'
0A5A 00                 DB      0
0A5B 32                 DB      50
0A5C 53484C44           DB      'SHLD'
0A60 22                 DB      34
0A61 4C484C44           DB      'LHLD'
0A65 2A                 DB      42
0A66 00                 DB      0
               ; CONDITION CODE TABLE
0A67 4E5A               DB      'NZ'
0A69 00                 DB      0
0A6A 5A                 DB      'Z'
0A6B 00                 DB      0
0A6C 08                 DB      8
0A6D 4E43               DB      'NC'
0A6F 10                 DB      16
0A70 43                 DB      'C'
0A71 00                 DB      0
0A72 18                 DB      24
0A73 504F               DB      'PO'
0A75 20                 DB      32
0A76 5045               DB      'PE'
0A78 28                 DB      40
0A79 50                 DB      'P'
0A7A 00                 DB      0
0A7B 30                 DB      48
0A7C 4D                 DB      'M'
0A7D 00                 DB      0
0A7E 38                 DB      56
0A7F 00                 DB      0
               ;
               ; THIS ROUTINE IS USED TO CHECK A GIVEN OPCODE
               ; AGAINST THE LEGAL OPCODES IN THE OPCODE TABLE
               ;
0A80 2A7410    COPC:    LHLD    ADDS
0A83 1A                 LDAX    D           ;FETCH CHARACTER
0A84 B7                 ORA     A           ;SET FLAGS
0A85 CA920A             JZ      COP1        ;JUMP IF TERMINATION CHARACTER
0A88 48                 MOV     C,B
0A89 CD8001             CALL    SEAR
0A8C 1A                 LDAX    D
0A8D C8                 RZ                  ;RETURN IF MATCH
0A8E 13                 INX     D           ; NEXT STRING
0A8F C3800A             JMP     COPC        ;CONTINUE SEARCH
0A92 3C        COP1:    INR     A           ;CLEAR ZFRO FLAG
0A93 13                 INX     D           ;INCREMENT ADDRESS
0A94 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 INDEXET JUMP CAN BE MADE TO
               ; PROCESS THE OPCODE FOR PASS 2.
               ;
               ;
```

```
ØA95 217E1Ø     OPCD:   LXI     H,ABUF  ;GET ADDRESS
ØA98 22741Ø             SHLD    ADDS
ØA9B 1157Ø9             LXI     D,OTAB  ;OPCODE TABLE ADDRESS
ØA9E Ø6Ø4               MVI     B,4     ;CHARACTER COUNT
ØAAØ CD8ØØA             CALL    COPC    ;CHECK OPCODES
ØAA3 CA3EØB             JZ      PSEU    ;JUMP IF A PSEUDO-OP
ØAA6 Ø5                 DCR     B       ;3 CHARACTER OPCODES
ØAA7 CD8ØØA             CALL    COPC
ØAAA CAB1ØA             JZ      OP1
ØAAD Ø4                 INR     B       ;4 CHARACTER  OPCODES
ØAAE CD8ØØA             CALL    COPC
ØAB1 215AØ8     OP1:    LXI     H,TYP1  ;TYPE 1 INSTRUCTIONS
ØAB4 ØEØ1       OP2:    MVI     C,1     ;1 BYTE INSTRUCTIONS
ØAB6 CA11ØB             JZ      OCNT
                ;
ØAB9 CD8ØØA     OPC2:   CALL    COPC    ;CHECK FOR STAX LDAX
ØABC 215DØ8             LXI     H,TYP2
ØABF CAB4ØA             JZ      OP2
ØAC2 CD8ØØA             CALL    COPC    ;CHECK FOR PUSH,POP INX
                ;               ;       ;DCX AND DAD
ØAC5 217ØØ8             LXI     H,TYP3
ØAC8 CAB4ØA             JZ      OP2
ØACB Ø5                 DCR     B       ;3 CHAR OPCODES
ØACC CD8ØØA             CALL    COPC    ;ACCUMULATOR INSTRUCTIONS
                ;               ;       ;INR DCR,MOV,RST
ØACF 218FØ8             LXI     H,TYP4
ØAD2 CAB4ØA             JZ      OP2
                ;
ØAD5 CD8ØØA     OPC3:   CALL    COPC    ;IMMEDIATE INSTRUCTIONS
ØAD8 21C4Ø8             LXI     H,TYP5
ØADB ØEØ2               MVI     C,2     ;2 BYTE INSTRUCTIONS
ØADD CA11ØB             JZ      OCNT
ØAEØ Ø4                 INR     B       ;4 CHAR OPCODES
ØAE1 CD8ØØA             CALL    COPC    ;JMP CALL,LXI LDA STA
                ;               ;       ;LHLD,SHLD OPCODES
ØAE4 CAØCØB             JZ      OP4
ØAE7 CD4CØ9             CALL    COND    ;CONDITIONAL INSTRUCTIONS
ØAEA C236ØB             JNZ     OERR    ;ILLEGAL OPCODE
ØAED C6CØ               ADI     192     ;ADD BASE VALUE TO RETURN
ØAEF 57                 MOV     D,A
ØAFØ Ø6Ø3               MVI     B,3     ;3 CHARACTER OPCODES
ØAF2 3A7E1Ø             LDA     ABUF    ;FETCH FIRST CHARACTER
ØAF5 4F                 MOV     C,A     ;SAVE CHARACTER
ØAF6 FE52               CPI     'R'     ;CONDITIONAL RETURN
ØAF8 7A                 MOV     A,D
ØAF9 CAB1ØA             JZ      OP1
ØAFC 79                 MOV     A,C
ØAFD 14                 INR     D       ;FORM CONDITIONAL JUMP
ØAFE 14                 INR     D
ØAFF FE4A               CPI     'J'     ;CONDITIONAL JUMP
ØBØ1 CAØBØB             JZ      OPAD
ØBØ4 FE43               CPI     'C'     ;CONDITIONAL CALL
ØBØ6 C236ØB             JNZ     OERR    ;ILLEGAL OPCODE
ØBØ9 14                 INR     D       ;FORM CONDIITIONAL CALL
ØBØA 14                 INR     D
ØBØB 7A         OPAD:   MOV     A,D     ;GET OPCODE
ØBØC 21FAØ8     OP4:    LXI     H,TYP6
ØBØF ØEØ3       OP5:    MVI     C,3     ;3 BYTE INSTRUCTION
```

```
ØB11 329C1Ø      OCNT:   STA     TEMP     ;SAVE OPCODE
                 ;
                 ; CHECK FOR OPCODE ONLY CONTAINING THE CORRECT NUMBER OF
                 ; CHARACTERS. THUS ADDQ, SAY, WOULD GIVE AN ERROR
                 ;
ØB14 3E7E                MVI     A.ABUF AND ØFFH ;LOAD BUFFER ADDRESS
ØB16 8Ø                  ADD     B        ;ADD LENGTH OF OPCODE
ØB17 5F                  MOV     E.A
ØB18 3E1Ø                MVI     A.ABUF/256
ØB1A CEØØ                ACI     Ø        ;GET HIGH ORDER ADDRESS
ØB1C 57                  MOV     D.A
ØB1D 1A                  LDAX    D        ;FETCH CHARACTER AFTER OPCODE
ØB1E B7                  ORA     A        ;IT SHOULD BE ZERO
ØB1F C2360B              JNZ     OERR     ;OPCODE ERROR
ØB22 3A941Ø              LDA     PASI     ;FETCH PASS INDICATOR
ØB25 Ø6ØØ        OCN1:   MVI     B.Ø
ØB27 EB                  XCHG
ØB28 2A921Ø      OCN2:   LHLD    ASPC     ;FETCH PROGRAM COUNTER
ØB2B Ø9                  DAD     B        ;ADD IN BYTE COUNT
ØB2C 22921Ø              SHLD    ASPC     ;STORE PC
ØB2F B7                  ORA     A        ;WHICH PASS?
ØB3Ø C8                  RZ               ;RETURN IF PASS 1
ØB31 3A9C1Ø              LDA     TEMP     ;FETCH OPCODE
ØB34 EB                  XCHG
ØB35 E9                  PCHL
                 ;
ØB36 21DØØC      OERR:   LXI     H.ERRO   ;SET ERROR ADDRESS
ØB39 ØEØ3                MVI     C.3      ;LEAVE 3 BYTES FOR PATCH
ØB3B C3220B              JMP     OCN1-3
                 ;
ØB3E 21821Ø      PSEU:   LXI     H.ABUF-4         ;SET BUFFER ADDRESS
ØB41 7E                  MOV     A.M      ;FETCH CHARACTER AFTER OPCODE
ØB42 B7                  ORA     A        ;SHOULD BE A ZERO
ØB43 C236ØB              JNZ     OERR
ØB46 3A941Ø              LDA     PASI     ;FETCH PASS INDICATOR
ØB49 B7                  ORA     A
ØB4A CA7AØ7              JZ      PSU1
ØB4D C3F5Ø7              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.
                 ;
ØB5Ø FE41        SLAB:   CPI     'A'      ;CHECK FOR LEGAL CHAR
ØB52 D8                  RC
ØB53 FE5B                CPI     'Z'+1    ;CHECK FOR ILLEGAL CHAR
ØB55 3F                  CMC
ØB56 D8                  RC               ;RETURN IF ILLEGAL CHAR
ØB57 CD9AØB              CALL    ALPS     ;PLACE SYMBOL IN BUFFER
ØB5A 217E1Ø              LXI     H.ABUF   ;SET BUFFER ADDRESS
ØB5D 22741Ø              SHLD    ADDS     ;SAVE ADDRESS
ØB6Ø Ø5                  DCR     B        ;CHECK IF ONE CHARACTER
ØB61 C2740B              JNZ     SLA1
```

```
                    ; CHECK IF PREDEFINED REGISTER NAME
ØB64 Ø4                     INR     B       ;SET B=1
ØB65 1185ØB                 LXI     D.RTAB  ;REGISTER TABLE ADDRESS
ØB68 CD8ØØA                 CALL    COPC    ;CHECK NAME OF REGISTER
ØB6B C274ØB                 JNZ     SLA1    ;NOT A PREDEFINED REGISTER
ØB6E 6F                     MOV     L.A     ;SET VALUE HIGH
ØB6F 26ØØ                   MVI     H,Ø
ØB71 C382ØB                 JMP     SLA2
ØB74 111A11         SLA1:   LXI     D.SYMT  ;SET SYMBOL TABLE ADDRESS
ØB77 3EØ5                   MVI     A LLAB  ;FETCH LENGTH OF LABEL
ØB79 32951Ø                 STA     NCHR
ØB7C CD65Ø1                 CALL    COMS    ;CHECK TABLE
ØB7F 4C                     MOV     C.H     ;SWAP H AND L
ØB8Ø 65                     MOV     H.L
ØB81 69                     MOV     L C
ØB82 37             SLA2:   STC             ;SET CARRY
ØB83 3F                     CMC             ;CLEAR CARRY
ØB84 C9                     RET             ;RETURN
                    ;
                    ; PREDEFINE REGISTER VALUES IN THIS TABLE
                    ;
ØB85 41             RTAB:   DB      'A'
ØB86 Ø7                     DB      7
ØB87 42                     DB      'B'
ØB88 ØØ                     DB      Ø
ØB89 43                     DB      'C'
ØB8A Ø1                     DB      1
ØB8B 44                     DB      'D'
ØB8C Ø2                     DB      2
ØB8D 45                     DB      'E'
ØB8E Ø3                     DB      3
ØB8F 48                     DB      'H'
ØB9Ø Ø4                     DB      4
ØB91 4C                     DB      'L'
ØB92 Ø5                     DB      5
ØB93 4D                     DB      'M'
ØB94 Ø6                     DB      6
ØB95 5Ø                     DB      'P'
ØB96 Ø6                     DB      6
ØB97 53                     DB      'S'
ØB98 Ø6                     DB      6
ØB99 ØØ                     DB      Ø       ;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 Ø-9 OR A-Z IS FOUND.
                    ;
ØB9A Ø6ØØ           ALPS:   MVI     B.Ø     ;SET COUNT
ØB9C 12             ALP1:   STAX    D       ;STORE CHARACTER IN BUFFER
ØB9D Ø4                     INR     B       ;INCREMENT COUNT
ØB9E 78                     MOV     A.B     ;FETCH COUNT
ØB9F FEØB                   CPI     11      ;MAXIMUM BUFFER SIZE
ØBA1 DØ                     RNC             ;RETURN IF BUFFER FILLED
ØBA2 13                     INX     D       ;INCREMENT BUFFER
ØBA3 23                     INX     H       ;INCREMENT INPUT POINTER
ØBA4 22961Ø                 SHLD    PNTR    ;SAVE LINE POINTER
ØBA7 7E                     MOV     A.M     ;FETCH CHARACTER
ØBA8 FE3Ø                   CPI     'Ø'     ;CHECK FOR LEGAL CHARACTERS
```

```
ØBAA D8                 RC
ØBAB FE3A               CPI     '9'-1
ØBAD DA9CØB             JC      ALP1
ØBBØ FE41               CPI     'A'
ØBB2 D8                 RC
ØBB3 FE5B               CPI     'Z'+1
ØBB5 DA9CØB             JC      ALP1
ØBB8 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
                ;
ØBB9 CD39Ø9     ASBL:   CALL    SBLK      ;GET FIRST ARGUMENT
ØBBC 21ØØØØ     ASCN:   LXI     H,Ø       ;GET A ZERO
ØBBF 22991Ø             SHLD    OPRD      ;INITIALIZE OPERAND
ØBC2 24                 INR     H
ØBC3 229A1Ø             SHLD    OPRI-1    ;INITIALIZE OPERAND INDICATOR
ØBC6 2A961Ø     NXT1:   LHLD    PNTR      ;FETCH SCAN POINTER
ØBC9 2B                 DCX     H
ØBCA CD93Ø1             CALL    ZBUF      ;CLEAR BUFFER
ØBCD 32991Ø             STA     SIGN      ;ZERO SIGN INDICATOR
ØBDØ 23         NXT2:   INX     H         ;INCREMENT POINTER
ØBD1 7E                 MOV     A,M       ;FETCH NEXT CHARACTER
ØBD2 FE21               CPI     ' '+1
ØBD4 DA78ØC             JC      SEND      ;JUMP IF CR OR BLANK
ØBD7 FE2C               CPI     ','       ;FIELD SEPARATOR
ØBD9 CA78ØC             JZ      SEND
                ; CHECK FOR OPERATORS
ØBDC FE2B               CPI     '+'       ;CHECK FOR PLUS
ØBDE CAE9ØB             JZ      ASC1
ØBE1 FE2D               CPI     '-'       ;CHECK FOR MINUS
ØBE3 C2F9ØB             JNZ     ASC2
ØBE6 32981Ø             STA     SIGN
ØBE9 3A9B1Ø     ASC1:   LDA     OPRI      ;FETCH OPERAND INDICATOR
ØBEC FEØ2               CPI     2         ;CHECK FOR TWO OPERATORS
ØBEE CAAEØC             JZ      ERRS      ;SYNTAX ERROR
ØBF1 3EØ2               MVI     A,2
ØBF3 329B1Ø             STA     OPRI      ;SET INDICATOR
ØBF6 C3DØØB             JMP     NXT2
                ;CHECK FOR OPERANDS
ØBF9 4F         ASC2:   MOV     C,A       ;SAVE CHARACTER
ØBFA 3A9B1Ø             LDA     OPRI      ;GET INDICATOR
ØBFD B7                 ORA     A         ;CHECK FOR TWO OPERANDS
ØBFE CAAEØC             JZ      ERRS      ;SYNTAX ERROR
ØCØ1 79                 MOV     A,C
ØCØ2 FE24               CPI     '$'       ;LC EXPRESSION
ØCØ4 C211ØC             JNZ     ASC3
ØCØ7 23                 INX     H         ;INCREMENT POINTER
ØCØ8 22961Ø             SHLD    PNTR      ;SAVE POINTER
ØCØB 2A921Ø             LHLD    ASPC      ;FETCH LOCATION COUNTER
ØCØE C34DØC             JMP     AVAL
                ;CHECK FOR ASCII CHARACTERS
ØC11 FE27       ASC3:   CPI     27H       ;CHECK FOR SINGLE QUOTE
ØC13 C23DØC             JNZ     ASC5      ;JUMP IF NOT QUOTE
ØC16 11ØØØØ             LXI     D,Ø       ;GET A ZERO
ØC19 ØEØ3               MVI     C,3       ;CHARACTER COUNT
ØC1B 23         ASC4:   INX     H         ;BUMP POINTER
ØC1C 22961Ø             SHLD    PNTR      ;SAVE
```

```
ØC1F 7E                 MOV     A,M     ;FETCH NEXT CHARACTER
ØC2Ø FEØD               CPI     ASCR    ;IS IT A CR?
ØC22 CACBØC             JZ      ERRA    ;ARGUMENT ERROR
ØC25 FE27               CPI     27H     ;IS IT QUOTE
ØC27 C234ØC             JNZ     SSTR
ØC2A 23                 INX     H       ;INCREMENT POINTER
ØC2B 22961Ø             SHLD    PNTR    ;SAVE
ØC2E 7E                 MOV     A,M     ;FETCH NEXT CHAR
ØC2F FE27               CPI     27H     ;CHECK FOR 2 QUOTES IN A ROW
ØC31 C24EØC             JNZ     AVAL+1  ;TERMINAL QUOTE
ØC34 ØD         SSTR:   DCR     C       ;CHECK COUNT
ØC35 CACBØC             JZ      ERRA    ;TOO MANY CHARACTERS
ØC38 53                 MOV     D,E
ØC39 5F                 MOV     E,A     ;SET CHARACTER IN BUFFER
ØC3A C31BØC             JMP     ASC4
ØC3D FE3Ø       ASC5:   CPI     'Ø'     ;CHECK FOR NUMERIC
ØC3F DACBØC             JC      ERRA    ;ILLEGAL CHARACTER
ØC42 FE3A               CPI     '9'+1
ØC44 D26CØC             JNC     ALAB
ØC47 CD88ØC             CALL    NUMS    ;GET NUMERIC VALUE
ØC4A DACBØC             JC      ERRA    ;ARGUMENT ERROR
ØC4D EB         AVAL:   XCHG
ØC4E 2A991Ø             LHLD    OPRD    ;FETCH OPERAND
ØC51 AF                 XRA     A       ;GET A ZERO
ØC52 329B1Ø             STA     OPRI    ;STOR IN OPERAND INDICATOR
ØC55 3A981Ø             LDA     SIGN    ;GET SIGN INDICATOR
ØC58 B7                 ORA     A       ;SET FLAGS
ØC59 C263ØC             JNZ     ASUB
ØC5C 19                 DAD     D       ;FORM RESULT
ØC5D 22991Ø     ASC7:   SHLD    OPRD    ;SAVE RESULT
ØC6Ø C3C6ØF             JMP     NXT1
ØC63 7D         ASUB:   MOV     A,L
ØC64 93                 SUB     E
ØC65 6F                 MOV     L,A
ØC66 7C                 MOV     A,H
ØC67 9A                 SBB     D
ØC68 67                 MOV     H,A
ØC69 C35DØC             JMP     ASC7
ØC6C CD5ØØB     ALAB:   CALL    SLAB
ØC6F CA4DØC             JZ      AVAL
ØC72 DACBØC             JC      ERRA    ;ILLEGAL SYMBOL
ØC75 C3B9ØC             JMP     ERRU    ;UNDEFINED SYMBOL
                ;
                ; GET HERE WHEN TERMINATING CHARACTER IS FOUND.
                ; CHECK FOR LEADING FIELD SEPARATOR
                ;
ØC78 3A9B1Ø     SEND:   LDA     OPRI    ;FETCH OPERAND INDICATOR
ØC7B B7                 ORA     A       ;SET FLAGS
ØC7C C2AEØC             JNZ     ERRS    ;SYNTAX ERROR
ØC7F 2A991Ø             LHLD    OPRD
ØC82 7C         SEN1:   MOV     A,H     ;GET HIGH ORDER BYTE
ØC83 119C1Ø             LXI     D,TEMP  ;GET ADDRESS
ØC86 B7                 ORA     A       ;SET FLAGS
ØC87 C9                 RET
                ;
                ; GET A NUMERIC VALUE WHICH IS EITHER HEXADECIMAL OR
                ; DECIMAL   ON RETURN. CARRY SET INDICATES AN ERROR
                ;
```

```
ØC88 CD9AØB    NUMS:   CALL    ALPS     ;GET NUMERIC
ØC8B 1B                DCX     D
ØC8C 1A                LDAX    D        ;GET LAST CHARACTER
ØC8D Ø17E1Ø            LXI     B.ABUF   ;SET BUFFER ADDRESS
ØC9Ø FE48              CPI     'H'      ;IS IT HEXADECIMAL?
ØC92 CAAØØC            JZ      NUM2
ØC95 FE44              CPI     'D'      ;IS IT DECIMAL
ØC97 C29CØC            JNZ     NUM1
ØC9A AF                XRA     A        ;GET A ZERO
ØC9B 12                STAX    D        ;CLEAR D FROM BUFFER
ØC9C CD2EØ2    NUM1:   CALL    ADEC     ;CONVERT DECIMAL VALUE
ØC9F C9                RET
ØCAØ AF        NUM2:   XRA     A        ;GET A ZERO
ØCA1 12                STAX    D        ;CLEAR H FROM BUFFER
ØCA2 C348Ø2            JMP     AHEX
               ;
               ; PROCESS REGISTER ERROR
ØCA5 3E52      ERRR:   MVI     A.'R'    ;GET INDICATOR
ØCA7 21ØØØØ            LXI     H Ø      ;GET A Ø
ØCAA 32B11Ø            STA     OBUF     ;SET IN OUTPUT BUFFER
ØCAD C9                RET
               ;PROCESS SYNTAX ERROR
ØCAE 3E53      ERRS:   MVI     A,'S'    ;GET INDICATOR
ØCBØ 32B11Ø            STA     OBUF     ;STORE IN OUTPUT BUFFER
ØCB3 21ØØØØ            LXI     H.Ø
ØCB6 C382ØC            JMP     SEN1
               ;PROCESS UNDEFINED SYMBOL ERROR
ØCB9 3E55      ERRU:   MVI     A.'U'    ;GET INDICATOR
ØCBB C3BØØC            JMP     ERRS-2
               ;PROCESS VALUE ERROR
ØCBE 3E56      ERRV:   MVI     A,'V'    ;GET INDICATOR
ØCCØ C3A7ØC            JMP     ERRR-2
               ;PROCESS MISSING LABEL ERROR
ØCC3 3E4D      ERRM:   MVI     A.'M'    ;GET INDICATOR
ØCC5 32B11Ø            STA     OBUF     ;STORE IN OUTPUT BUFFER
ØCC8 C3F7Ø6            JMP     AOU1     ;DISPLAY ERROR AND RETURN
               ;PROCESS ARGUMENT ERROR
ØCCB 3E41      ERRA:   MVI     A 'A'    ;GET INDICATOR
ØCCD C3BØØC            JMP     ERRS 2
               ; PROCESS OPCODE ERROR
               ; STORE 3 BYTES OF ZERO IN OBJECT CODE TO PROVIDE
               ; FOR A PATCH.
ØCDØ 3E4F      ERRO:   MVI     A.'O'    ;GET INDICATOR
ØCD2 32B11Ø            STA     OBUF     ;STORE IN OUTPUT BUFFER
ØCD5 3A941Ø            LDA     PASI     ;FETCH PASS INDICATOR
ØCD8 B7                ORA     A        ;WHICH PASS
ØCD9 C8                RZ               ;RETURN IF PASS1
ØCDA ØEØ3              MVI     C,3      ;NEED 3 BYTES
ØCDC AF        ERO1:   XRA     A        ;GET A ZERO
ØCDD CD1AØ9            CALL    ASTO     ;PUT IN LISTING AND MEMORY
ØCEØ ØD                DCR     C
ØCE1 C2DCØC            JNZ     ERO1
ØCE4 C9                RET
               ; PROCESS LABEL ERROR
ØCE5 3E4C      ERRL:   MVI     A,'L'    ;GET INDICATOR
ØCE7 C3D2ØC            JMP     ERRO+2
               ;PROCESS DUPLICATE LABEL ERROR
ØCEA 3E44      ERRD:   MVI     A,'D'    ;GET ERROR INDICATOR
```

```
ØCEC 32B11Ø                 STA     OBUF     ;STORE IN OUTPUT BUFFER
ØCEF CDEAØ6                 CALL    AOUT     ;DISPLAY ERROR
ØCF2 C338Ø7                 JMP     OPC      ;PROCESS OPCODE
                 ;
                 ;
                 ; THIS ROUTINE SETS OR CLEARS BREAKPOINTS
                 ;
ØCF5 3A7E1Ø      BREAK:     LDA     ABUF     ;CHECK FOR AN ARG
ØCF8 B7                     ORA     A
ØCF9 CA37ØD                 JZ      CLRB     ;IF NO ARG, GO CLEAR BREAKPOINTS
ØCFC 16Ø8                   MVI     D,NBR    ;ELSE  GET NUMBER OF BREAKPOINTS
ØCFE 21ØC1Ø                 LXI     H,BRT    ;AND ADDR OF TABLE
ØDØ1 7E          B1:        MOV     A,M      ;GET HI BYTE OF ENTRY
ØDØ2 23                     INX     H
ØDØ3 46                     MOV     B,M      ;GET LO BYTE OF ENTRY
ØDØ4 BØ                     ORA     B        ;CHECK FOR EMPTY ENTRY
ØDØ5 CA11ØD                 JZ      B2       ;BRANCH IF EMPTY
ØDØ8 23                     INX     H        ;ELSE GO ON TO NEXT ENTRY
ØDØ9 23                     INX     H
ØDØA 15                     DCR     D        ;BUMP COUNT
ØDØB C2Ø1ØD                 JNZ     B1       ;AND TRY AGAIN
ØDØE C381Ø4                 JMP     WHAT     ;OOPS, NO ROOM
ØD11 2B          B2:        DCX     H
ØD12 EB                     XCHG
ØD13 2A8A1Ø                 LHLD    BBUF     ;GET ADDRESS
ØD16 EB                     XCHG             ;IN D,E
ØD17 7A                     MOV     A,D      ;CHECK FOR ADDR   11D
ØD18 B7                     ORA     A
ØD19 C222ØD                 JNZ     B3
ØD1C 7B                     MOV     A,E
ØD1D FEØB                   CPI     11
ØD1F DA81Ø4                 JC      WHAT     ;OOPS. TOO LOW
ØD22 72          B3:        MOV     M,D      ;SAVE ADDRESS
ØD23 23                     INX     H
ØD24 73                     MOV     M,E
ØD25 23                     INX     H
ØD26 1A                     LDAX    D        ;PICK UP INSTRUCTION
ØD27 77                     MOV     M,A      ;SAVE IT
ØD28 3ECF                   MVI     A,ØCFH   ;REPLACE IT WITH A
ØD2A 12                     STAX    D        ;RESTART INSTRUCTION
ØD2B 3EC3                   MVI     A,ØC3H   ;SET UP LO MEMORY
ØD2D 32Ø8ØØ                 STA     8        ;WITH A JUMP TO BRKP
ØD3Ø 2151ØD                 LXI     H,BRKP
ØD33 22Ø9ØØ                 SHLD    9
ØD36 C9                     RET              ;THEN RETURN
                 ;
                 ; THIS ROUTINE CLEARS ALL BREAKPOINTS
                 ;
ØD37 21ØC1Ø      CLRB:      LXI     H,BRT    ;GET TABLE ADDRESS
ØD3A ØEØ8                   MVI     C,NBR    ;GET NUMBER OF BREAKPOINTS
ØD3C AF          CLBL:      XRA     A        ;GET A ZERO
ØD3D 56                     MOV     D,M      ;GET HI-BYTE OF ENTRY
ØD3E 77                     MOV     M,A
ØD3F 23                     INX     H
ØD4Ø 5E                     MOV     E,M      ;GET LO-BYTE OF ENTRY
ØD41 77                     MOV     M,A
ØD42 23                     INX     H
ØD43 46                     MOV     B,M      ;GET INST BYTE
```

```
ØD44 23                 INX     H
ØD45 7A                 MOV     A,D      ;WAS THIS A NULL ENTRY
ØD46 B3                 ORA     E
ØD47 CA4CØD             JZ      CL2      ;BRANCH IF IT WAS
ØD4A 78                 MOV     A B
ØD4B 12                 STAX    D        ;ELSE  PLUG INST BACK IN
ØD4C ØD         CL2:    DCR     C        ;BUMP COUNT
ØD4D C23CØD             JNZ     CLBL     ;GO DO NEXT ONE
ØD5Ø C9                 RET              ;RETURN WHEN DONE
                ;
                ; COME HERE WHEN WE HIT A BREAKPOINT
                ;
ØD51 22Ø81Ø     BRKP:   SHLD    HOLD+8   ;SAVE H L
ØD54 E1                 POP     H        ;GET PC
ØD55 2B                 DCX     H        ;ADJUST IT
ØD56 22ØA1Ø             SHLD    HOLD+1Ø  ;SAVE IT
ØD59 F5                 PUSH    PSW      ;SAVE FLAGS
ØD5A E1                 POP     H        ;GET THEM INTO HL
ØD5B 22ØØ1Ø             SHLD    HOLD     ;NOW STORE THEM FOR USER
ØD5E 21ØØØØ             LXI     H,Ø
ØD61 39                 DAD     SP       ;GET STACK POINTER
ØD62 31Ø81Ø             LXI     SP,HOLD+8 ;SET NEW SP
ØD65 E5                 PUSH    H        ;SAVE OLD SP
ØD66 D5                 PUSH    D        ;SAVE D E
ØD67 C5                 PUSH    B        ;SAVE B C
ØD68 2F                 CMA              ;COMPLEMENT ACC
ØD69 D3FF               OUT     ØFFH     ;DISPLAY IT IN THE LIGHTS
ØD6B 31B11Ø             LXI     SP AREA+18 ;SET SP AGAIN
ØD6E 2AØA1Ø             LHLD    HOLD+1Ø  ;GET PC
ØD71 EB                 XCHG             ;INTO D.E
ØD72 21ØC1Ø             LXI     H.BRT    ;GET ADDR OF TABLE
ØD75 Ø6Ø8               MVI     B.NBR    ;AND NUMBER OF ENTRIES
ØD77 7E         BL1:    MOV     A M      ;GET AN ENTRY FROM THE TABLE
ØD78 23                 INX     H
ØD79 BA                 CMP     D        ;DOES IT MATCH
ØD7A C282ØD             JNZ     BL2      ;BRANCH IF NOT
ØD7D 7E                 MOV     A.M      ;ELSE GET NEXT BYTE
ØD7E BB                 CMP     E        ;CHECK IT
ØD7F CA8BØD             JZ      BL3      ;IT MATCHES!
ØD82 23         BL2:    INX     H        ;BUMP AROUND THIS ENTRY
ØD83 23                 INX     H
ØD84 Ø5                 DCR     B        ;BUMP COUNT
ØD85 CA81Ø4             JZ      WHAT     ;NOT IN OUR TABLE!
ØD88 C377ØD             JMP     BL1
                ;
ØD8B 23         BL3:    INX     H
ØD8C 7E                 MOV     A.M      ;GET INSTR BYTE
ØD8D 12                 STAX    D        ;PUT IT BACK
ØD8E AF                 XRA     A        ;CLEAR ENTRY IN TABLE
ØD8F 2B                 DCX     H
ØD9Ø 77                 MOV     M A
ØD91 2B                 DCX     H
ØD92 77                 MOV     M,A
ØD93 CD3AØ1             CALL    CRLF     ;RESTORE THE CARRIAGE
ØD96 3AØB1Ø             LDA     HOLD+11  ;GET HI-BYTE OF PC
ØD99 CD67Ø2             CALL    HOUT     ;TYPE IT
ØD9C 3AØA1Ø             LDA     HOLD+1Ø  ;GET LO-BYTE OF PC
ØD9F CD67Ø2             CALL    HOUT     ;TYEP IT
```

```
ØDA2 21A8ØD            LXI     H,BMES  ;TELL USER WHAT IT IS
ØDA5 C38704            JMP     MESS    ;GO BACK TO COMMAND LEVEL
                 ;
ØDA8 2042524541BMES:   DB      ' BREAK',13
                 ;
                 ; THIS ROUTINE PROCEEDS FROM A BREAKPOINT
                 ;
ØDAF 3A7E1Ø      PROC:  LDA     ABUF    ;CHECK FOR ARG
ØDB2 B7                ORA     A
ØDB3 CABCØD            JZ      P1      ;JMP IF NO ARG
ØDB6 2A8A1Ø            LHLD    BBUF    ;ELSE  GET ARG
ØDB9 22ØA1Ø            SHLD    HOLD+1Ø ;PLUG IT INTO PC SLOT
ØDBC 31ØØ1Ø      P1:    LXI     SP,HOLD ;SET SP TO POINT AT REG'S
ØDBF F1                POP     PSW     ;RESTORE PSW
ØDCØ C1                POP     B       ;RESTORE B.C
ØDC1 D1                POP     D       ;RESTORE D E
ØDC2 E1                POP     H       ;GET OLD SP
ØDC3 F9                SPHL            ;RESTORE IT
ØDC4 2AØA1Ø            LHLD    HOLD+1Ø ;GET PC
ØDC7 E5                PUSH    H       ;PUT IT ON STACK
ØDC8 2AØ81Ø            LHLD    HOLD+8  ;RESTORE H L
ØDCB C9                RET             ;AND PROCEED
                 ;
                 ; SYSTEM RAM
                 ;
1ØØØ                   ORG     1ØØØH
                 ;
                 ; DEFINE BREAKPOINT REGION
                 ;
ØØØ8 =           NBR     EQU     8       ;NUMBER OF BREAKPOINTS
1ØØØ             HOLD:   DS      12      ;REGISTER HOLD AREA
1ØØC             BRT:    DS      3*NBR   ;BREAKPOINT TABLE
                 ;
                 ; FILE AREA PARAMETERS
ØØØ6 =           MAXFIL  EQU     6       ;MAX # OF FILES
ØØØ5 =           NMLEN   EQU     5       ;NAME LENGTH
ØØØD =           FELEN   EQU     NMLEN+8 ;DIRECTORY ENTRY LENGTH
1Ø24             FILEØ:  DS      NMLEN
1Ø29             BOFP:   DS      2
1Ø2B             EOFP:   DS      2
1Ø2D             MAXL:   DS      4
1Ø31             FILTB:  DS      (MAXFIL-1)*FELEN
1Ø72             INSP:   DS      2       ;INSERT LINE POSITION
1Ø72 =           DELP    EQU     INSP    ;DELETE LINE POSITION
ØØØD =           ASCR    EQU     13      ;ASCII CARRIAGE RETURN VALUE
1Ø74             HCON:   DS      2
1Ø74 =           ADDS    EQU     HCON    ;FIND ADDRESS
1Ø76             FBUF:   DS      NMLEN   ;FILE NAME BUFFER
1Ø7B             FREAD:  DS      2       ;FREE ADDRESS IN DIRECTORY
1Ø7D             FEF:    DS      1       ;FREE ENTRY FOUND FLAG
1Ø7D =           FOCNT   EQU     FEF     ;OUTPUT COUNTER
1Ø7E             ABUF:   DS      12      ;ASCII BUFFER
1Ø8A             BBUF:   DS      4       ;BINARY BUFFER
1Ø8E             SCNT:   DS      1
1Ø8F             DCNT:   DS      1       ;DUMP ROUTINE COUNTER
1Ø9Ø             TABA:   DS      2       ;SYMBOL TABLE END ADDRESS
1Ø92             ASPC:   DS      2       ;ASSEMBLER PROGRAM COUNTER
1Ø94             PASI:   DS      1       ;PASS INDICATOR
```

```
1095            NCHR:   DS      1       ;LENGTH OF STRING FOR COMPARE
1096            PNTR:   DS      2       ;LINE POINTER STORAGE
1098            SIGN:   DS      1       ;SIGN STORAGE FOR SCAN
1099            OPRD:   DS      2       ;OPERAND STORAGE
109B            OPRI:   DS      1       ;OPERAND FOUND INDICATOR
109C            TEMP:   DS      1
1072 =          APNT    EQU     INSP    ;ASSEMBLE LINE POINTER
109E =          AERR    EQU     SCNT    ;ASSEMBLER ERROR PRINT SWITCH
109D            OIND:   DS      2       ;OUTPUT ADDRESS
0005 =          LLAB    EQU     5       ;LENGTH OF LABELS
109F            AREA:   DS      18
10B1            OBUF:   DS      16      ;OUTPUT BUFFER AREA
10C1                    DS      5
10C6            IBUF:   DS      83
1119            IPOS:   DS      1       ;TELETYPE LINE POSITION
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
                ;
111A                    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 rub-outs. 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 LCADER.
                ;
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
```

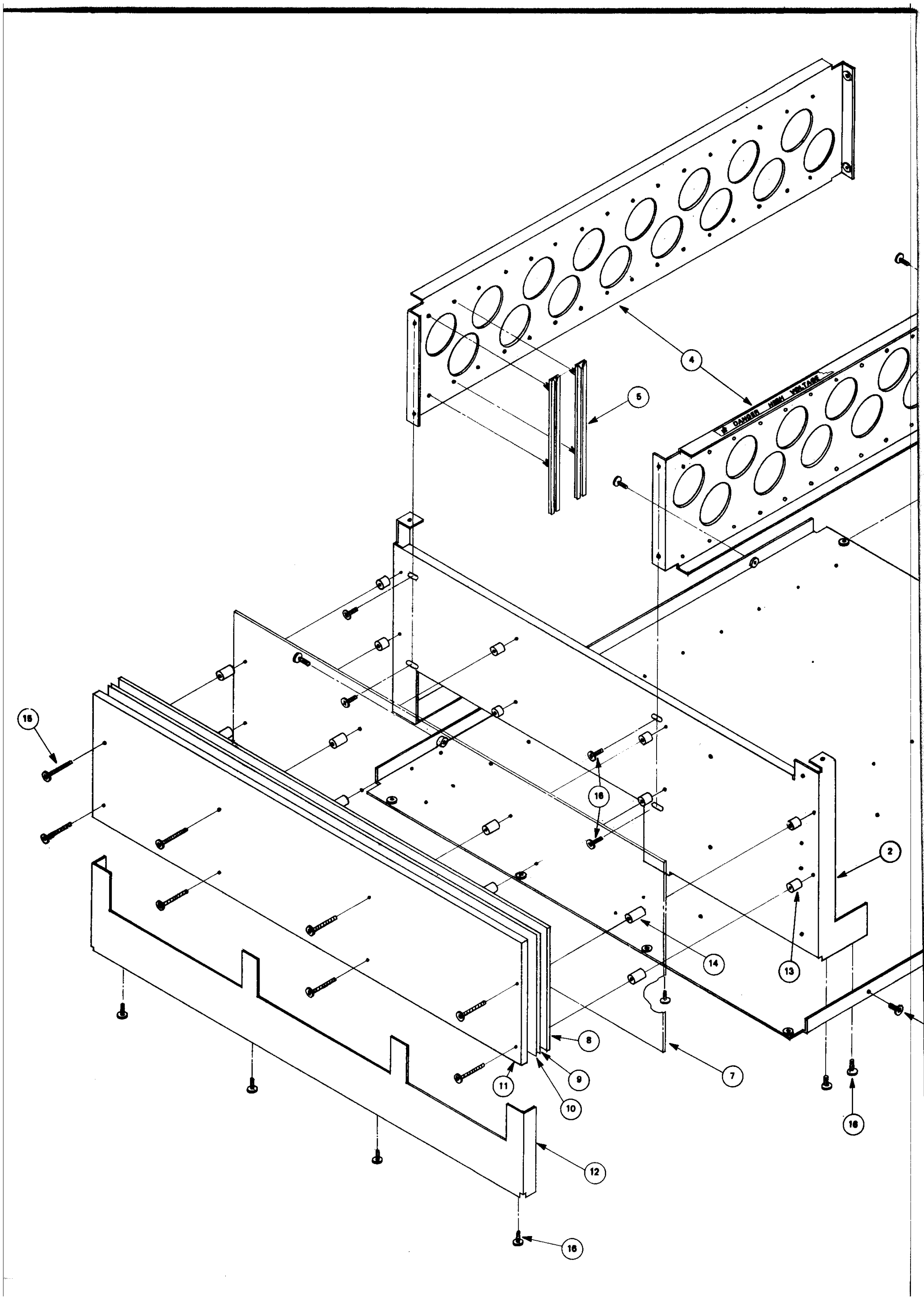
```
                      ;
                      ; *** IMSAI PAPER TAPE LOADER ***
                      ;
                      ;       REV 0   3/3/76
                      ;
                      ; THIS LOADER IS DESIGNED TO LOAD PAPER TAPES IN
                      ; THE STANDARD OBJECT FORMAT (SEE 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
```

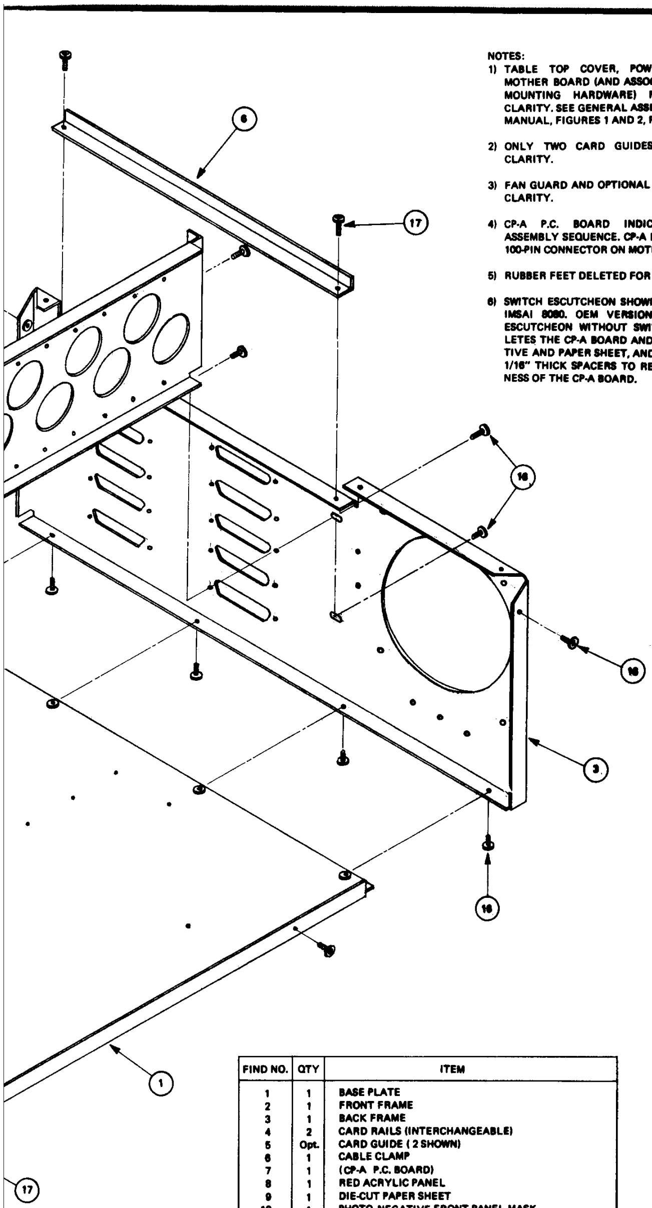
```
                   ;
FD31 DB03          LOOP2:  IN      3         ;GET SIO STATUS
FD33 E602                  ANI     2         ;CHECK FOR CHARACTER
FD35 CA31FD                JZ      LOOP2     ;KEEP WAITING
FD38 7B                    MOV     A,E       ;GET FLAG
FD39 B7                    ORA     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

CHASSIS CABINET
POWER SUPPLY
CPA
MPU-A
RAM 4A


4
5
DANGER HIGH VOLTAGE
15
16
2
13
14
8
7
9
10
11
16
12
16



NOTES:

1) TABLE TOP COVER, POWER SUPPLY, AND MOTHER BOARD (AND ASSOCIATED HOLES AND MOUNTING HARDWARE) NOT SHOWN FOR CLARITY. SEE GENERAL ASSEMBLY CHAPTER OF MANUAL, FIGURES 1 AND 2, FOR DETAILS.

2) ONLY TWO CARD GUIDES (5) SHOWN FOR CLARITY.

3) FAN GUARD AND OPTIONAL FAN DELETED FOR CLARITY.

4) CP-A P.C. BOARD INDICATED TO SHOW ASSEMBLY SEQUENCE. CP-A BOARD PLUGS INTO 100-PIN CONNECTOR ON MOTHER BOARD.

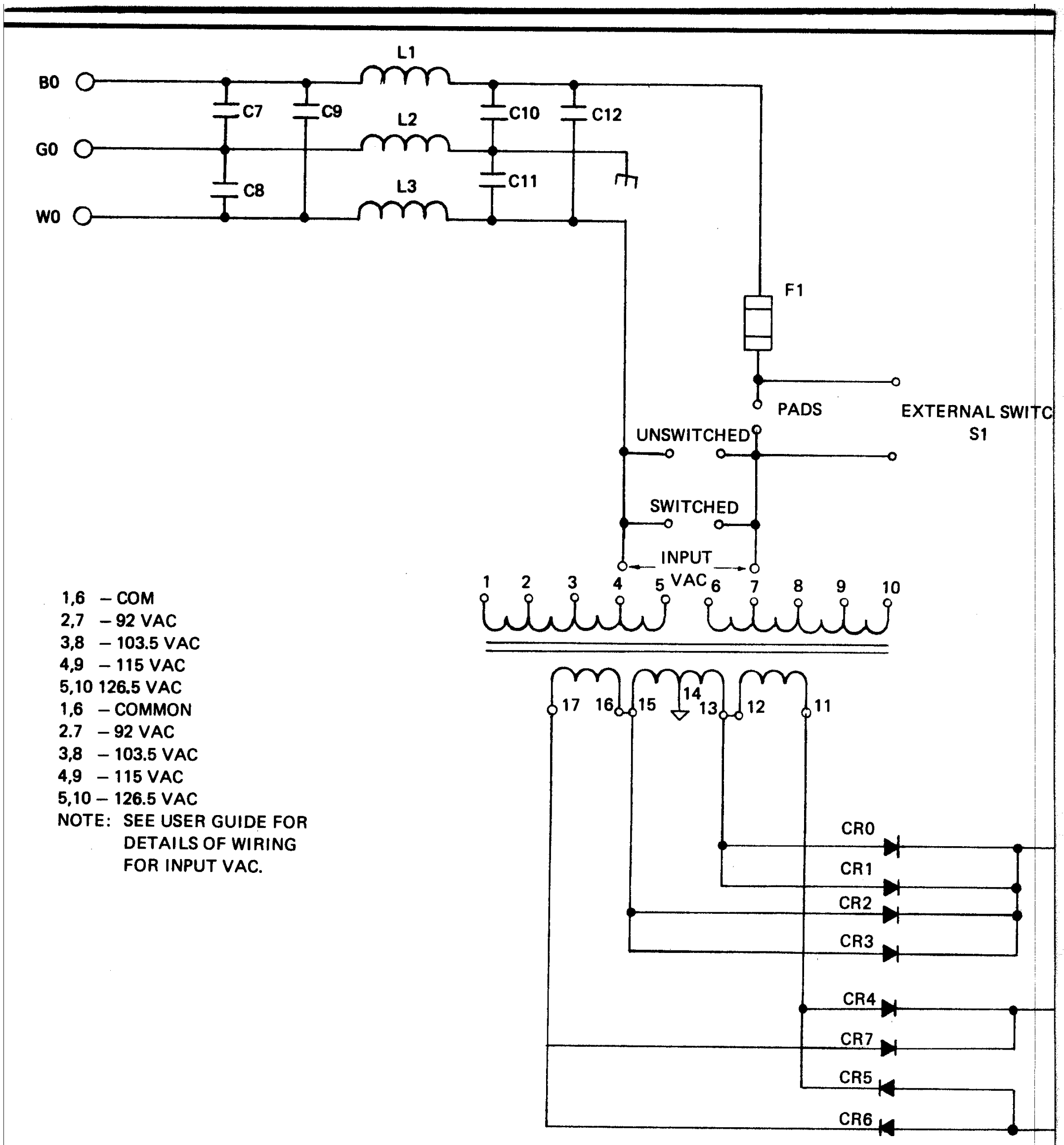
5) RUBBER FEET DELETED FOR CLARITY.

6) SWITCH ESCUTCHEON SHOWN IS FOR REGULAR IMSAI 8080. OEM VERSION HAS ALTERNATE ESCUTCHEON WITHOUT SWITCH CUTOUTS, DELETES THE CP-A BOARD AND THE PHOTO NEGATIVE AND PAPER SHEET, AND USES ADDITIONAL 1/16" THICK SPACERS TO REPLACE THE THICKNESS OF THE CP-A BOARD.

FIND NO.	QTY	ITEM
1	1	BASE PLATE
2	1	FRONT FRAME
3	1	BACK FRAME
4	2	CARD RAILS (INTERCHANGEABLE)
5	Opt.	CARD GUIDE (2 SHOWN)
6	1	CABLE CLAMP
7	1	(CP-A P.C. BOARD)
8	1	RED ACRYLIC PANEL
9	1	DIE-CUT PAPER SHEET
10	1	PHOTO NEGATIVE FRONT PANEL MASK
11	1	CLEAR ACRYLIC PANEL
12	1	SWITCH ESCUTCHEON
13	8	No. 6x¼" UNTHREADED NYLON SPACERS
14	8	No. 6x7/16" UNTHREADED NYLON SPACERS
15	8	6-32x1¼" ALLEN BUTTON HEAD MACHINE SCREWS, BLACK OXIDE
16	21	6-32x5/16" PHILLIPS PAN HEAD MACHINE SCREWS
17	6	6-32x½" PHILLIPS PAN HEAD MACHINE SCREWS

8080 CHASSIS CABINET ASSEMBLY EXPLODED VIEW

7/76



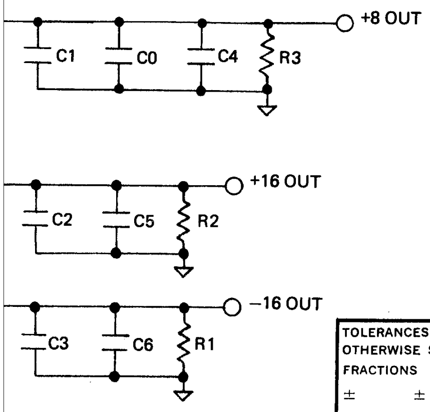
1,6 – COM
2,7 – 92 VAC
3,8 – 103.5 VAC
4,9 – 115 VAC
5,10 126.5 VAC
1,6 – COMMON
2.7 – 92 VAC
3,8 – 103.5 VAC
4,9 – 115 VAC
5,10 – 126.5 VAC

NOTE: SEE USER GUIDE FOR DETAILS OF WIRING FOR INPUT VAC.

REVISIONS

LTR	DESCRIPTION	DATE	APPROVED
0	ORIGINAL PS-C REV. 1	5/76	
1	REDRAW TO PS-C–U REV. 1	12/76	

C0 C1	95K uF
C2 C3	10K uF
C4 C5 C6	.1uF
C7 thru C12	.04 uF 500V or .01 uF 1000V
CR0 thru CR3	MR1121
CR4 thru CR7	MR501
F1	5A/2.5A
L1 thru L3	8uH
R1 R2	1K ½W
R3	470 ½W
S1	NOT SUPPLIED



TOLERANCES UNLESS OTHERWISE SPECIFIED
FRACTIONS ± DEC. ± ANGLES ±

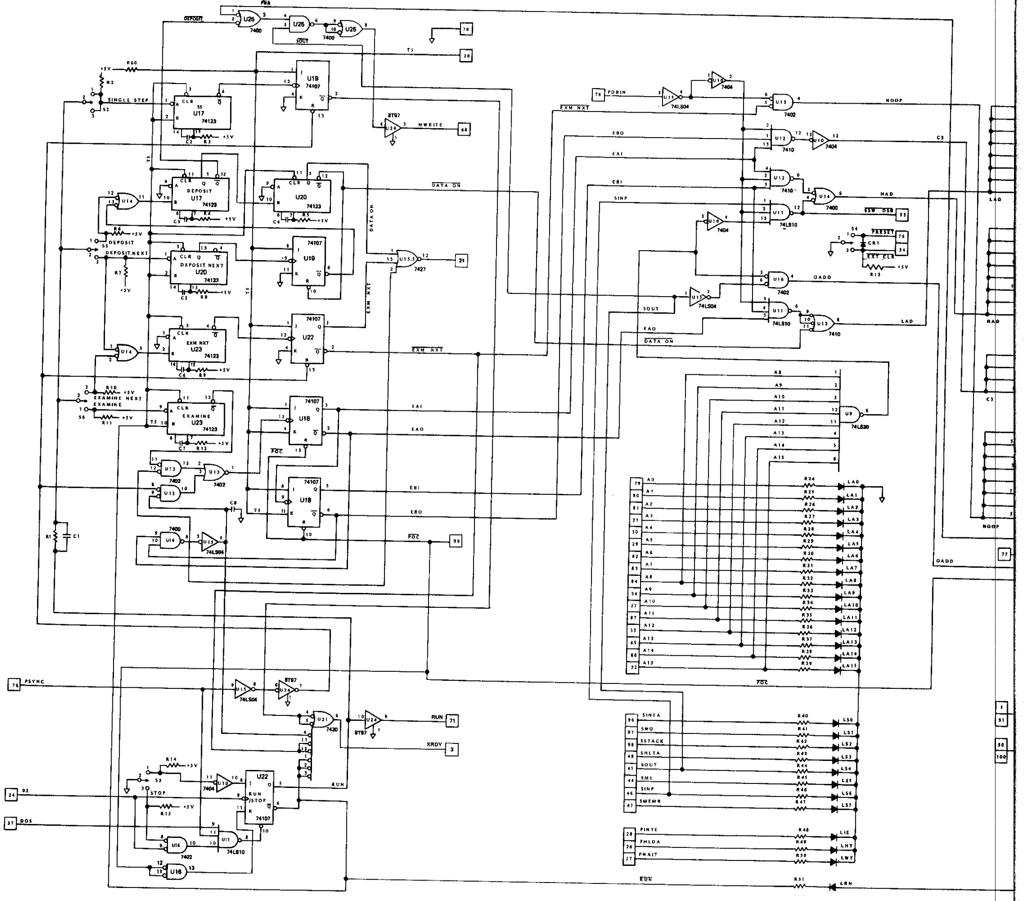
8080 SYSTEM
SCHEMATIC DIAGRAM
PS-C–U REV. 1 12/76

APPROVALS	DATE
DRAWN PRU	12/76
CHECKED	

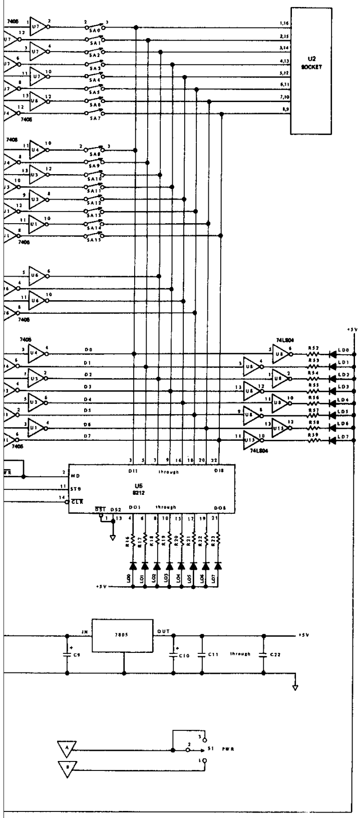
SCALE	SIZE	DRAWING NO.
—	B	91–2010015

DO NOT SCALE DRAWING

SHEET


PWR
DEPOSIT
U25
7400
SOUT
70
T5
20
+5V
R60
R2
SINGLE STEP
S2
SS
U17
74123
CLR
C2
R3
U19
74107
8T97
U24
MWRITE
68
U14
DEPOSIT
C3
R4
U20
C4
R5
DATA ON
R6
S5
DEPOSIT NEXT
R7
C5
R8
EXM NXT
U15.5
7427
21
U23
C6
R9
U22
R10
EXAMINE NEXT
EXAMINE
S6
R11
C7
R12
U18
EA1
EA0
POC
U13
7402
EB1
EB0
C8
T3
R1
C1
74LS04
99
76
PSYNC
U21
7430
RUN
71
XRDY
3
R14
S3
STOP
R15
U10
7404
RUN /STOP
24
Ø2
37
DOS
U11
74LS10
U16
78
PDBIN
NOOP
U12
7410
C3
HAD
SSW DSB
53
S4
PRESET
75
CR1
EXT CLR
54
R13
OADD
SINP
LAD
A8
A9
A10
A11
A12
A13
A14
A15
U9
74LS30
79 A0 R24 LA0
80 A1 R25 LA1
81 A2 R26 LA2
31 A3 R27 LA3
30 A4 R28 LA4
29 A5 R29 LA5
82 A6 R30 LA6
83 A7 R31 LA7
84 A8 R32 LA8
34 A9 R33 LA9
37 A10 R34 LA10
87 A11 R35 LA11
33 A12 R36 LA12
85 A13 R37 LA13
86 A14 R38 LA14
32 A15 R39 LA15
96 SINTA R40 LS0
97 SWO R41 LS1
98 SSTACK R42 LS2
48 SHLTA R43 LS3
45 SOUT R44 LS4
44 SM1 R45 LS5
46 SINP R46 LS6
47 SMEMR R47 LS7
28 PINTE R48 LIE
26 PHLDA R49 LHT
27 PWAIT R50 LWT
RUN R51 LRN
77
1
51
50
100

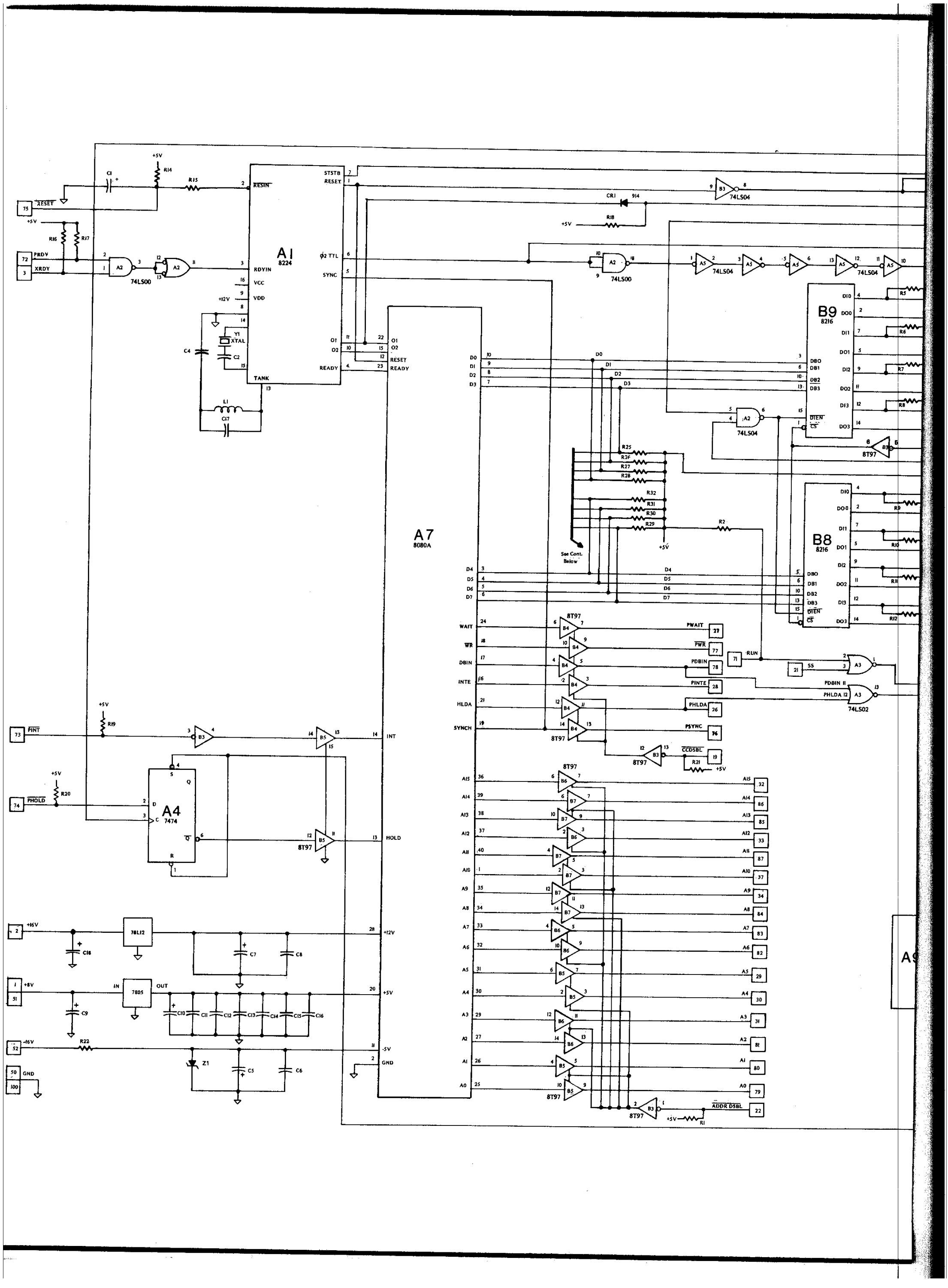
REVISIONS			
LTR	DESCRIPTION	DATE	APPROVED
0	ORIGINAL	12/75	
1	ECN76–0051 & ECO77–0039	11/76 & 4/77	
2	ECO77–0098	8/77	



U1 U3 U4 U6 U7	7405
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	.1 uF
C3 C8	.001 uF
C4	.01 uF
C9 C10	33 uF
CR1	1N914

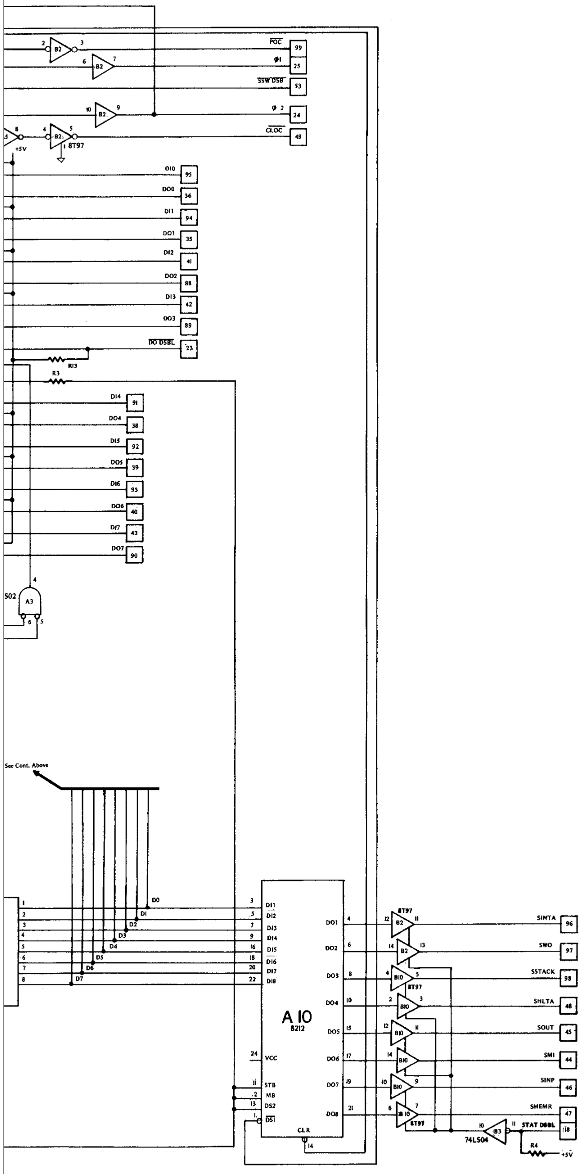
LO0 THRU LO7 LA0 THRU LA15 LD0 THRU LD7 LS0 THRU LS7 LIE LHD LRN LWT	LED'S
R1	270K ¼W
R2 R6 R7 R10 R11 R13 THRU R15 R60	1K ¼W
R3 R4 R5 R8 R9 R12	47K ¼W
R16 THRU R59	220 ¼W

TOLERANCES UNLESS OTHERWISE SPECIFIED FRACTIONS ± DEC. ± ANGLES ±		© 1977 IMSAI MFG. CORP., SAN LEANDRO, CA. ALL RIGHTS RESERVED WORLDWIDE MADE IN U.S.A.		
APPROVALS	DATE	IMSAI 8080 SYSTEM CPA REV. 4 SCHEMATIC DIAGRAM		
DRAWN				
CHECKED		SCALE	SIZE B	DRAWING NO.
		DO NOT SCALE DRAWING		SHEET


A1
8224
RESIN
RDYIN
VCC
VDD
STSTB
RESET
φ2 TTL
SYNC
O1
O2
READY
TANK
XTAL
Y1
C1
C2
C4
C17
L1
R14
R15
R16
R17
+5V
+12V
RESET
PRDY
XRDY
75
72
3
A2
74LS00
B3
74LS04
CR1
914
R18
A5
74LS04
B9
8216
DI0
DO0
DI1
DO1
DI2
DO2
DI3
DO3
DB0
DB1
DB2
DB3
DIEN
CS
R5
R6
R7
R8
8T97
A7
8080A
D0
D1
D2
D3
D4
D5
D6
D7
R25
R26
R27
R28
R32
R31
R30
R29
R2
See Cont. Below
B8
8216
R9
R10
R11
R12
WAIT
WR
DBIN
INTE
HLDA
SYNCH
B4
PWAIT
PWR
PDBIN
PINTE
PHLDA
PSYNC
27
77
78
28
26
76
RUN
71
SS
21
A3
74LS02
PDBIN
PHLDA
CCDSBL
19
R21
PINT
73
R19
B5
INT
PHOLD
74
R20
A4
7474
D
C
Q
R
HOLD
A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
B6
B7
32
86
85
33
87
37
34
84
83
82
29
30
31
81
80
79
ADDR DSBL
22
R1
+16V
78L12
C18
C7
C8
+12V
+8V
IN
7805
OUT
C9
C10
C11
C12
C13
C14
C15
C16
+5V
-16V
52
R22
Z1
C5
C6
-5V
GND
50
100
A9

REVISIONS

LTR	DESCRIPTION	DATE	AP
0	ORIGINAL REV.2	11/75	
1	REV.4	2/76	
2	UPDATE & MOD	2/77	



A1	8224
A2	74LS00
A3	74LS04
A4	7474
A5	74LS04
A7	8080A
A9	DATA BUS SOCKET
A10	8212
B2	8T97
B3	74LS04
B4 B5 B6 B7	8T97
B8 B9	8216
B10	8T97
C1 C5 C7 C9 C10	33uF
C4 C6 C8 C11 thru C16 C18	.1uF
C2	39pF
C17	56pF
CR1	IN914
Z1	IN751
L1	1.0uH
R1 thru R13 R15 thru R17 R19 thru R21	1K ¼w
R14 R18 R25 thru R32	4.7K ¼w
R22	470 ½w
Y1	18uHz

TOLERANCES UNLESS OTHERWISE SPECIFIED
FRACTIONS ± DEC. ± ANGLES ±

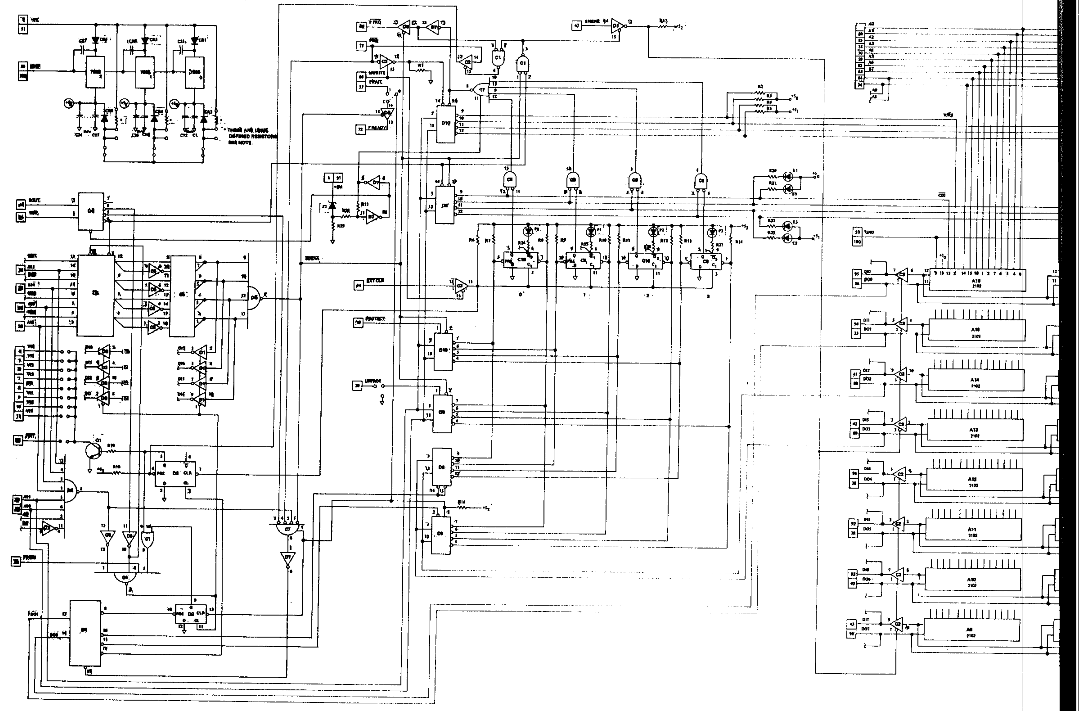
IMSAI MFG. CORP.
SAN LEANDRO, CA.

APPROVALS	DATE
DRAWN	
CHECKED	

SCHEMATIC DIAGRAM
MPU-A REV.4 2/76

SCALE | SIZE | DRAWING NO.

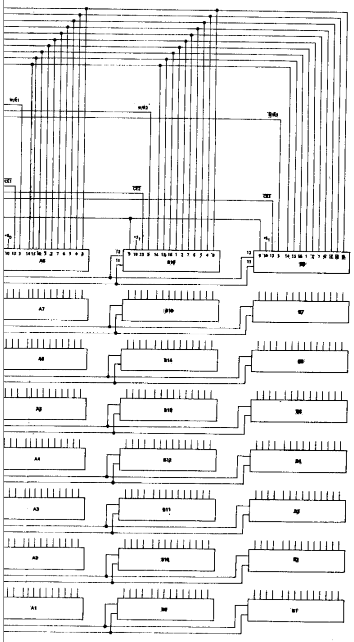
DO NOT SCALE DRAWING | SHEET



+8V
THESE ARE LOGIC DEFINED RESISTORS SEE NOTE.
A0
A1
A2
A3
A4
A5
A6
A7
A9
A8
W/R0
CE0
GND
DI0
DO0
DI1
DO1
DI2
DO2
DI3
DO3
DI4
DO4
DI5
DO5
DI6
DO6
DI7
DO7
A18
2102
A14
2102
A13
2102
A12
2102
A11
2102
A10
2102
A9
2102
PROTECT
UNPROT

REVISIONS

LTR	DESCRIPTION	DATE	APPROVED
0	REV. 2	5/76	
1	MOD & UPDATE	12/76	



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	74LS139
D5	74LS157
D6	7430
D7	7404
D8 D9 D10	74LS156
C1 thru C34	.1 uF
C35 thru C37	33 uF
CR1 thru CR6	1N4002

Z1	1N751A
Q1	2N3904
R1 thru R18	1K ¼W
R19 thru R28	220 ¼W
R29	100 ¼W
E0 thru E3	GREEN LED
P0 thru P3	RED LED

NOTE :
✗ THESE ARE USER DEFINED RESISTORS. REFER TO USERS GUIDE FOR EXPLANATION.

TOLERANCES UNLESS OTHERWISE SPECIFIED FRACTIONS ± DEC. ± ANGLES ±	IMSAI MFG. CORP. SAN LEANDRO, CA.
APPROVALS / DATE DRAWN CHECKED	RAM 4A–4 REV. 3 8/76 SCHEMATIC DIAGRAM
	SCALE / SIZE B / DRAWING NO.
	DO NOT SCALE DRAWING / SHEET