The Ithaca InterSystems
EPROM Programmer

PRELIMINARY

Ithaca InterSystems, Inc.
Edition 1.0

TABLE OF CONTENTS

1.0 INTRODUCTION

Non-volatile, programmable memory is an important component of many microcomputer systems. The InterSystems EPROM Programmer supports this facility conveniently and at reasonable cost. Flexible enough to use with many different kinds of EPROMs, providing optimal burn timing for each, the card has been designed so that the you may concentrate on the system and software, and devote only a minimal amount of time and effort to the mechanics of EPROM burning.

The InterSystems EPROM Programmer Utility is a convenient, interactive program, designed to efficiently guide you through the programming process.

This manual provides the information required to set up, install and operate the EPROM Programmer board. Section 1 of the manual provides a general description of the Programmer board; section 2 is a functional overview of the board. These sections contain vital information that will save you a great deal of trouble later if you read them carefully now. Section 3 is a comprehensive tutorial introduction to the EPROM Programmer Utility.

You should be completely familiar with your S-100 bus microcomputer and the CP/M operating system features. If you're not, don't go any further until you've read the appropriate system documentation and CP/M user documentation. If you are using an InterSystems ITHACA 800, ITHACA 525, or PDS-80 system, you should read the System Overview manual, located in the User Manual binder.

Like all InterSystems products, the EPROM Programmer is completely tested before shipment, and should provide many years of trouble-free service. It is important, however, that the board be inspected upon receiving; and in any event, service may some day be needed. The InterSystems Policies manual contains information about these subjects.

The remainder of this section of the manual contains a description of the board and a summary of the features available on the EPROM Programmer card.

PRELIMINARY

1.1 PHYSICAL DESCRIPTION

The EPROM Programmer is a non-standard 6 by 10 inch (15.24 by 25.4 cm), S-100 bus, plug-in circuit board. The non-standard height allows easy access to the card's burn socket above other cards in a system mainframe while still allowing installation of the mainframe cover when the board is in the system but not being used.

The board has a 28-pin zero insertion force (ZIF) burn EPROM socket to accept 24- and 28-pin EPROMs being programmed or read.

The board features a 50-pin top edge connector to accept interchangeable personality modules used to configure the board to function with individual EPROM types. Personality modules are 1.5 by 3.25 inches (38 by 84 cm). InterSystems includes two standard personality modules as specified by the buyer. Additional personality modules for other EPROM types may be purchased separately from Ithaca InterSystems.

1.2 FEATURES

The Ithaca InterSystems EPROM Programmer can read and program 1/2K to 8K EPROMs. Interchangable personality modules are used to configure the board for the type of EPROM inserted. Programming is controlled by off-board software, the InterSystems EPROM Programmer utility, or in many cases by front panel operations.

The EPROM Programmer burns the data byte on the S-100 Even Data bus into the EPROM memory location specified on the S-100 address bus when a MWRITE pulse occurs. During the MWRITE pulse, the EPROM card places the mainframe CPU into a wait state, so that the address is stable for the 0.7 to 50 ms burn period.

The card provides the EPROM programming function in a convenient package that includes features that make using the card completely straightforward. Board features are listed below.

- The board reads and programs the following EPROM types: 2704, 2708, 2758, TI 2716 (TMS 2716JL*), Intel 2716 (TMS 2516), 2732, 2732A, TMS 2532, Intel 2764, TMS 2564, and the MCM 68764. The board has provisions for future expansion to operate with 16Kx8 (128K) EPROMs.

- Low cost personality modules configure the EPROM Programmer for the particular EPROM type:

  - The size of the EPROMs memory space is automatically varied (even though various types of EPROMs occupy different spaces in memory, the card always adjusts to this specification, occupying no more than the necessary address space).
  - The high voltage power supply is maintained at the optimal level for each EPROM type (21, 25, or 26 V).
  - The program pulse rise/fall times are set to optimal times for each EPROM type.
  - The program pulse widths are set to optimal times for each EPROM.

- EPROMs can be inserted or removed without turning off the computer. The SAFE TO REMOVE/ON switch offers complete safety by removing critical voltages from the ZIF socket and personality module connector.

- The READ ONLY/PROGRAM POWER switch removes the programming voltage to prevent accidental writes.

- The board can be used in systems with a 2, 4, or 6MHz system clock. All critical programming time constants are determined by on-board hardware.

- Programming may be conducted either in block mode (as required by 2704, 2708, and Texas Instrument TMS2716 EPROMs) or in "single shot" mode. In the latter case, a standard front panel deposit can perform the function, or it may be controlled by software.

PRELIMINARY

1.3 PERSONALITY MODULE OPTIONS

The EPROM Programmer is supplied with the two personality modules selected by the buyer. Additional personality modules are available from InterSystems to support the following EPROMs.

Three Supply EPROMs: +5 V, -5 V, +12 V

EPROM Type	Personality Module
2704	001622
2708	001623
T.I. 2716	001624

One Supply EPROMs: +5 V

EPROM Type	Personality Module
2516, Intel 2716, 2758, and 2508	001633
TMS 2532	001634
2732	001635
2732A	001636
2764	001637
TMS 2564	001638
MCM68764	001639

PRELIMINARY

2.0 FUNCTIONAL OVERVIEW

This section contains information about the jumper, switches and personality modules of the EPROM Programmer card. It is important that you read this section carefully. You will need to be familiar with this descriptive information whenever you use the Programmer board. It may be useful for you to refer to your board while you are reading the section.

Figure 1 shows the location of key components on the EPROM Programmer. Detailed descriptions of jumper options and switch positions are provided.

2.1 AVAILABLE FEATURES

In addition to the hardware features described in this section, the board has been designed to be hardware modifiable to support the following features. Contact InterSystems Customer Service for further information about these modifications.

- Phantom disable for using the board in systems that do not support Phantom. The EPROM Programmer is be set to drive PHANTOM whenever it is accessed. This allows memory that responds to PHANTOM to be overlayed by the EPROM Programmer. The board is shipped with the feature enabled.

- 128 kbyte EPROM programming. The board can be modified to provide the addressing requirements of the respective personality modules.

2.2 JUMPER J1: EXTENDED ADDRESSING OPTION

See Figure 1. The EPROM Programmer can operate in systems with either standard 16-bit addressing or extended 24-bit addressing. Jumper J1 is set to either enable or disable extended addressing.

24-Bit Addressing Option:	J1
Disabled (16-bit)	A
Enabled (24-bit)	B

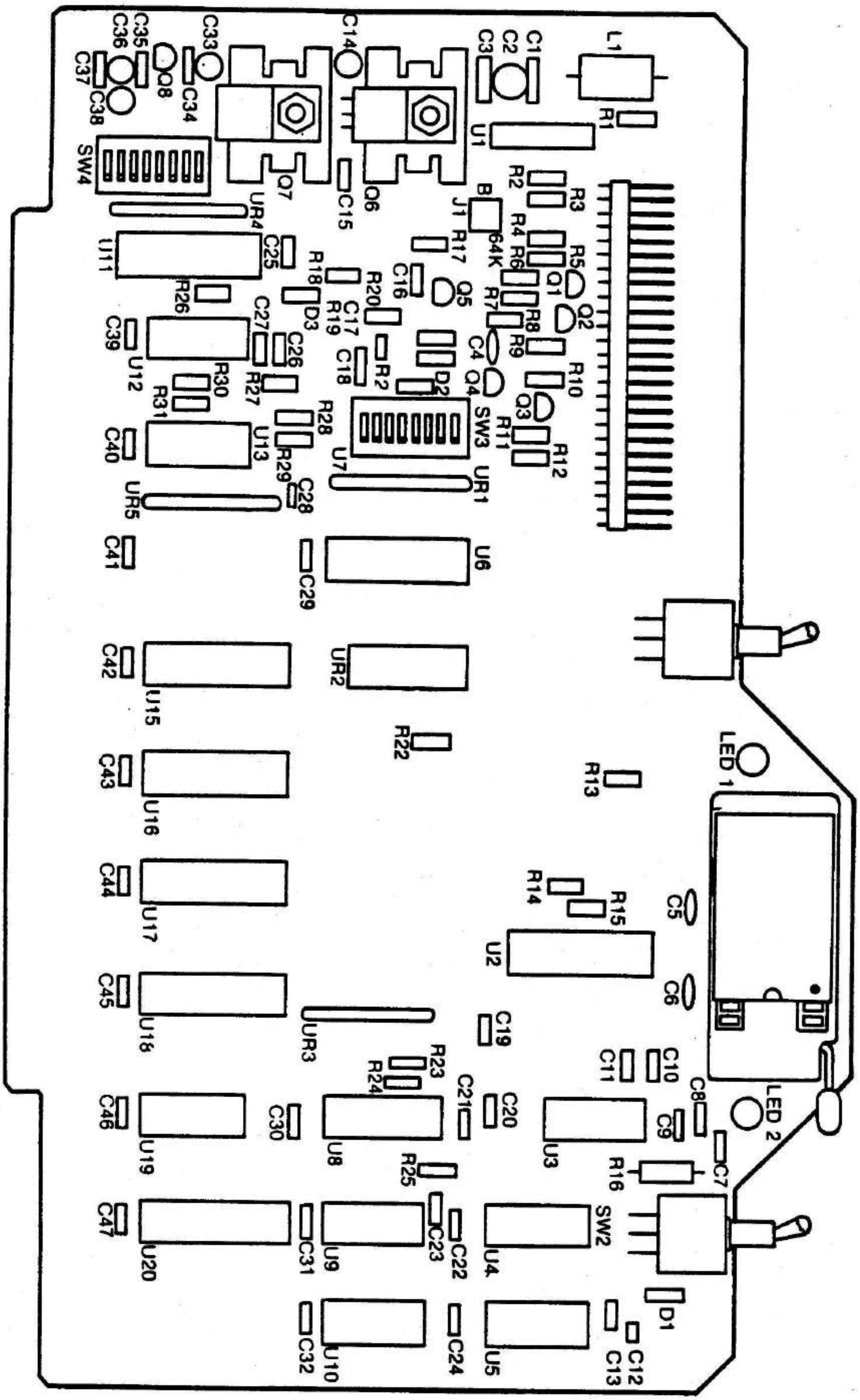


Figure 1

PRELIMINARY

2.3 SWITCHES

SW1: READ ONLY/PROGRAM POWER

See Figure 2.

When SW1 is in the READ ONLY position, the 21 to 26 volt programming voltage is defeated. LED1 (red) is OFF, indicating that there is no danger of an EPROM in the ZIF socket being programmed.

With SW1 in the PROGRAM POWER position, the required programming voltage is applied. LED1 is ON indicating that the high voltage supply is enabled.

SW2: SAFE TO REMOVE/ON

See Figure 2.

In the SAFE TO REMOVE position, power to the ZIF socket and personality module are defeated, and the EPROM may be safely removed from the card. LED2 (green) is ON, indicating a "safe" condition. The SAFE TO REMOVE position of the switch also de-selects the board, turning off all reads, writes, wait requests, and Phantom.

With SW2 in the ON position, +5 V is supplied to the burn EPROM and the personality module. LED2 (green) is OFF, indicating that it is "not safe" to insert or remove an EPROM or a personality module on the board.

When SW2 is in the ON position, the Burn EPROM appears in the system memory map as set by switches SW3 and SW4. (See the description of SW3 and SW4 below.)

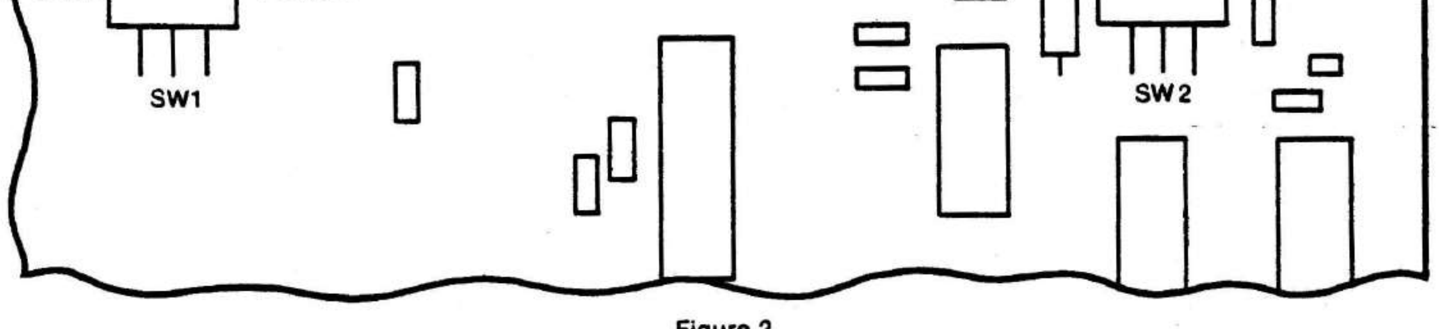


Figure 2

SW3(b): Burn EPROM Address Selection

The burn EPROM may reside at any 8K boundary in a 64K memory space, either the default 64K space of a 16-bit address system, or the 64K space specified by SW4 in a system with 24-bit extended addressing. The size of the space that the EPROM occupies is set by the personality module.

Switches SW3-6 through SW3-8 are set to the three most significant bits of the starting binary address of the burn EPROM, where SW3-6 through SW3-8 correspond to address lines A15 through A13, respectively. A switch set to OPEN corresponds to a logic ONE on the address line. A switch in the CLOSED position corresponds to a logic ZERO. See Figure 3.

For example, if the burn EPROM is to reside at 8000 Hex, the three most significant bits of the address (A15 through A13) are 1, 0, and 0; consequently, SW3-6 is set OPEN, switches SW3-7 and SW3-8 are set CLOSED.

SW4: Extended Address Selection

SW4 is used to enable the operation of the EPROM Programmer in one of 256 possible 64K memory segments (providing a total address range of 16 megabytes). In systems using 16-bit addressing (defining a 64K address space), SW4 is disabled by setting jumper J1 to the disable position, A; in systems using extended addressing, J1 is set to position B.

Switches SW4-1 through SW4-8 are set to the starting binary address of the desired 64K segment, and correspond to address lines A23 through A16 respectively. A switch set to the OPEN position corresponds to a logic ONE on the address line. A switch set to the CLOSED position corresponds to a logic ZERO on the address line.

For example, if the card resides at 23,XXXX Hex, switches SW4-3, SW4-7 and SW4-8 are set to OPEN, and the remaining switches are set CLOSED.

Figure 3 shows the location of J1 and switches SW3 and SW4.

2.4 LEDs

The EPROM Programmer's two LEDs will provide you with a visual indication of the status of the board during the programming operation.

LED1 RED.

When ON, LED1 indicates the burn voltage is being applied to the drivers. SW1 is set to PROGRAM POWER. Do not insert or remove a personality module or the EPROM being programmed.

When OFF, LED1 indicates the board is in a READ ONLY mode. SW1 is set to READ ONLY. The burn voltage is not applied to the drivers. Note that LED1 must be OFF (and LED2 ON) before you insert or remove the module or EPROM.

LED2 GREEN.

When ON, indicates a SAFE TO REMOVE condition. SW2 is set to SAFE TO REMOVE. It is safe to insert or remove the EPROM being programmed or the personality module.

When OFF, indicates that +5 V is applied to the burn socket and personality module. SW2 is set to the ON position. It is not safe to insert or remove the personality module or EPROM to be programmed.

PRELIMINARY

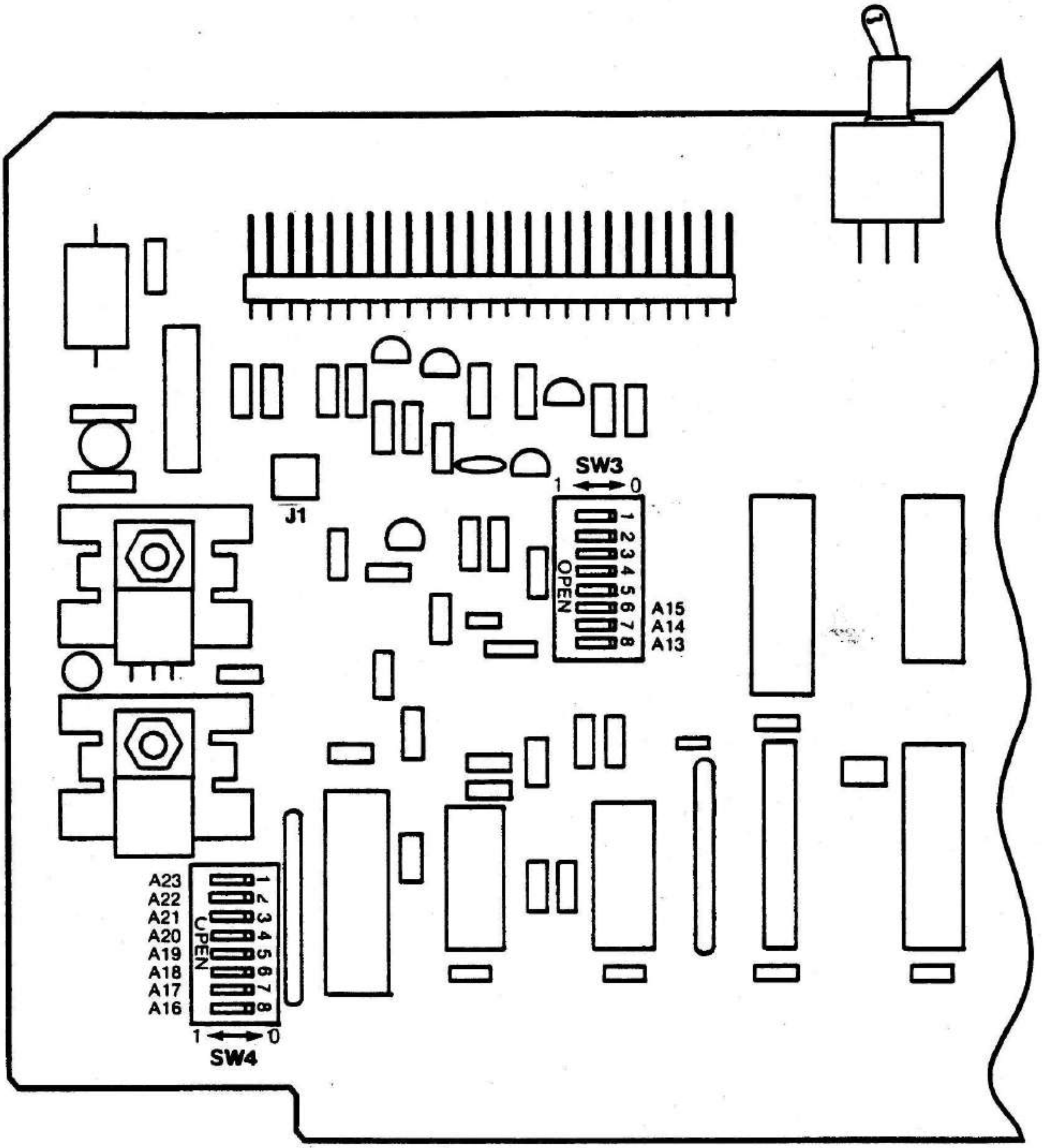


Figure 3

PRELIMINARY

2.5 PERSONALITY MODULES

The Programmer board is supplied with the two personality modules selected by the purchaser. Additional personality modules for other EPROM types are available from InterSystems.

The personality modules are used to configure the EPROM Programmer for use with specific EPROMs. Personality modules establish:

* Size of EPROM memory space
* Optimal high voltage level for Burn Voltage
* Program pulse rise/fall times
* Program pulse width
* Appropriate input/output connections for pulse generators, driver circuits, and power supplies.

Personality modules should be removed or installed only when LED1 is OFF (SW1 set to READ ONLY), LED2 is ON (SW2 set to SAFE TO REMOVE), and no EPROM is inserted in the ZIF socket. Damage to the personality module or board may occur if these conditions are not met.

Personality modules are attached to the the Programmer board as shown in Figure 4. Each personality module has a 50-pin, two-row socket that connects to the Programmer board 50-pin connector. Personality modules are keyed so that they can not be installed incorrectly.

Two basic types of personality modules are are available.

Type 1 Used with three-power-supply EPROMs (such as 2708's)
Type 2 Used with single voltage EPROMs

PRELIMINARY

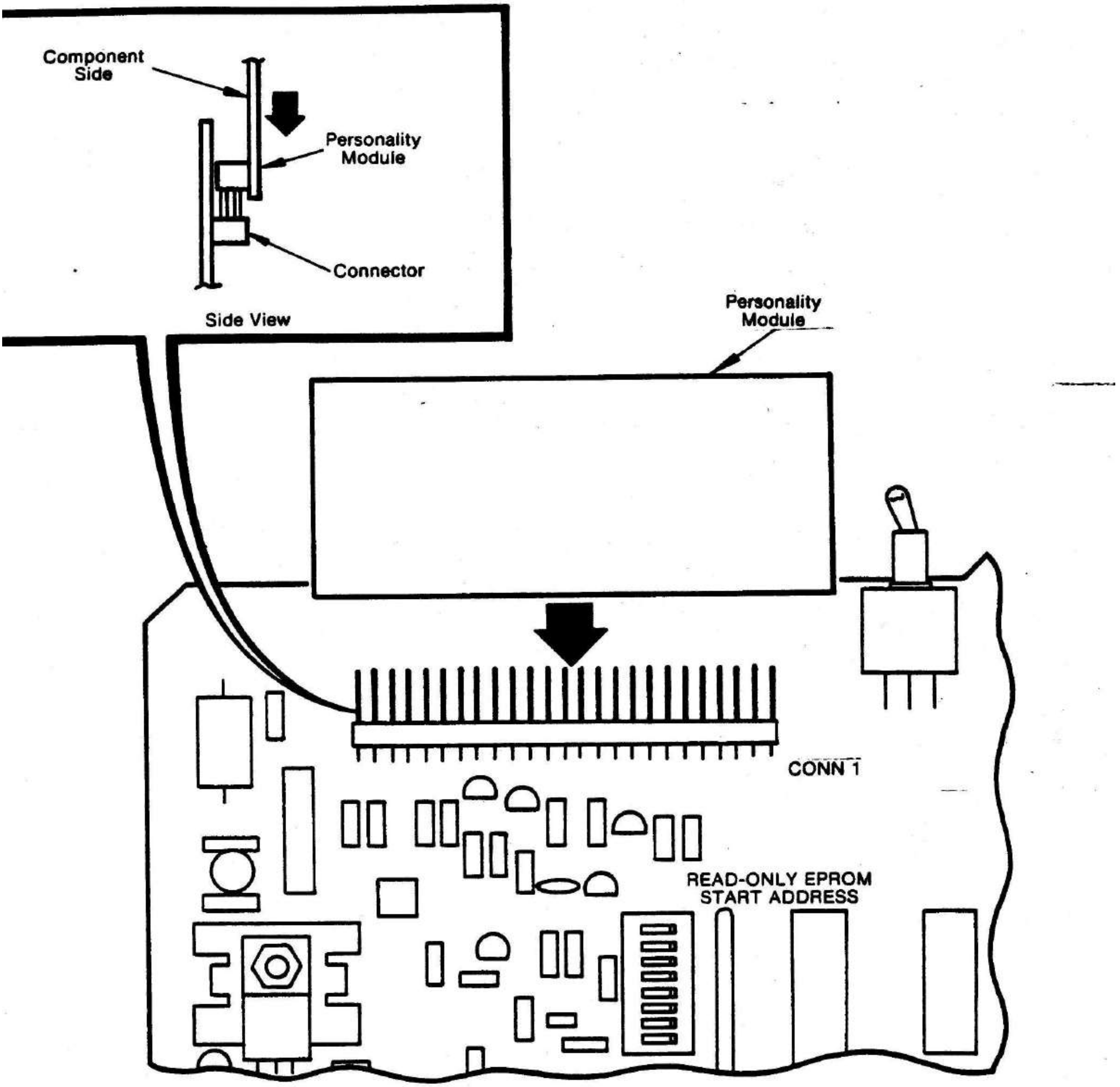


Figure 4

PRELIMINARY

2.6 PROGRAMMING ZIF SOCKET

A 28-pin zero insertion force (ZIF) socket is provided for the EPROM being programmed or read.

The socket accepts both 24-pin and 28-pin EPROMs. 24-pin EPROMs are inserted LEFT justified. The board is marked to identify the correct orientation for Pin 1 of both EPROMs. Figure 5 illustrates the use of a TEXTOOL socket. To install an EPROM in the socket, move the handle so that it is positioned perpendicular to the board. After inserting the EPROM, reposition the handle parallel to the board. To remove the EPROM, position the handle perpendicular to the board. (Programmer boards may be fitted with sockets that open with the lever parallel to the board and close with the lever perpendicular to the board.)

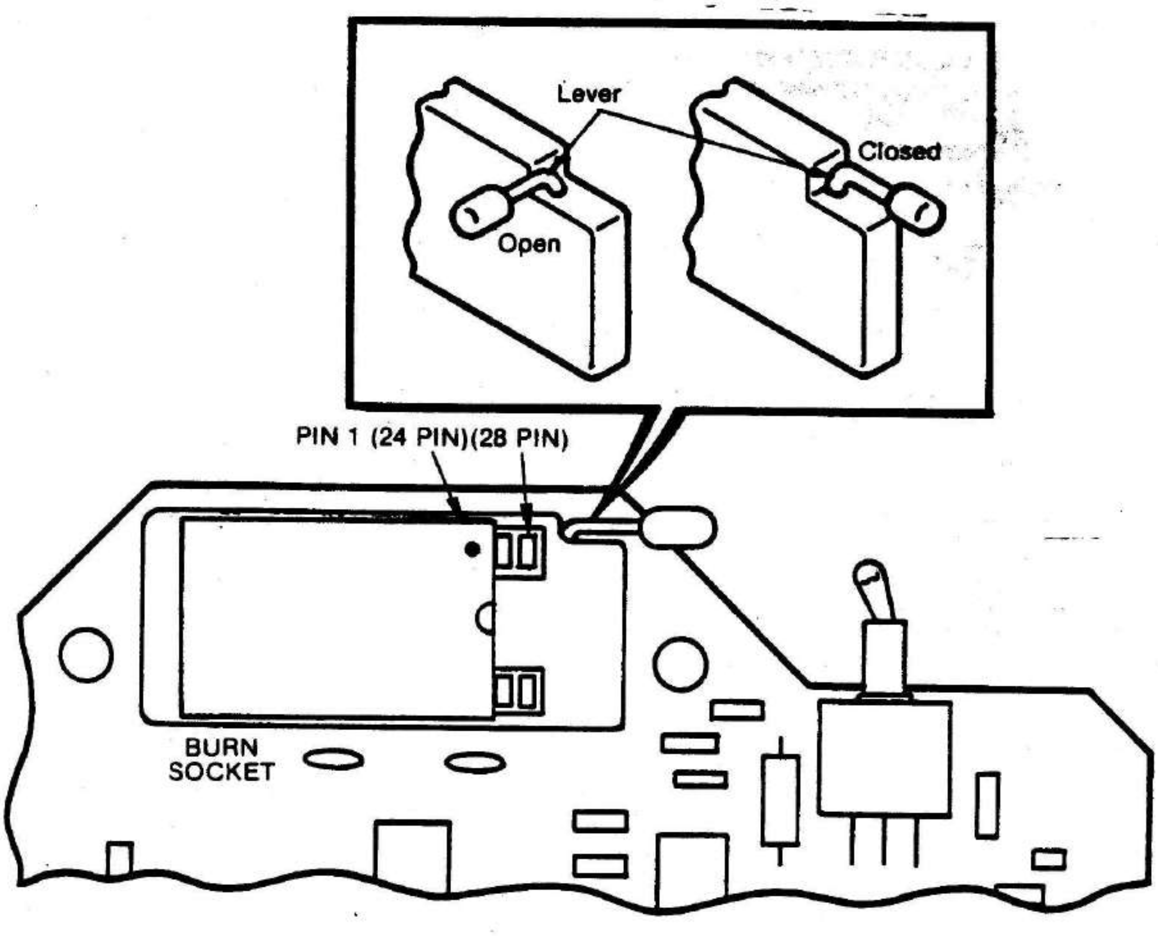


Figure 5

PRELIMINARY

2.6 PHANTOM

The EPROM Programmer is set to drive the Phantom line whenever the board is accessed. This allows memory boards in the system that decode Phantom to be overlayed by the EPROM Programmer. The board is shipped with the feature enabled. The feature can be disabled by a hardware modification. Contact InterSystems Customer Service for further information.

3.0 EPROM PROGRAMMER UTILITY

This section of the EPROM Programmer manual provides the information required to burn EPROMs using the EPROM Programmer utility (EPU.COM) with the EPROM Programmer board. You will find that using the utility allows you to perform required programming functions easily and efficiently. If you elect to not use this utility program, you should refer to section 4 or 5 of this manual. Section 4 describes the basic steps to take in using the Programmer board with user-generated software and section 5 gives you the information you will need to program EPROMs from a system front panel (such as the InterSystems Front Panel board).

3.1 GETTING STARTED

Before you do anything with the EPROM Programmer utility you should have read both the first section of this manual, that introduces the EPROM Programmer board, and the second section, that describes how the Programmer board is used in an S-100 bus system. If you haven't read either or both, please go back and do so. Both sections contain important information that pertains directly to using the utility described in this section.

You must also be familiar with your microcomputer and terminal. If you're not, take time now to read the manuals provided with your system. If you are using an InterSystems computer, refer to the system overview located in the front of your User Manual binder.

This chapter is organized into general descriptive information and detailed procedural information. After reading sections 1.0 and 2.0 of this manual, you may wish to use this one as follows.

1) Read section 3.2 to learn about the features and capabilities of the EPROM Programmer utility.
2) Refer to section 3.3 for instructions for using the floppy disk supplied with the EPROM Programmer utility package.
3) Use section 3.4 as a reference when you use the utility program. You may also wish to refer to section 3.5, in which error handling is described.

PRELIMINARY

3.2 INTRODUCTION TO THE UTILITY

The InterSystems EPROM Programmer utility provides you with a straighforward, interactive approach to programming EPROMs with the EPROM Programmer. The utility is menu driven. This section of the manual provides a description of the Programmer utility and item-by-item descriptions of the utility prompts.

3.2.1 Hardware Requirements for EPROM Programmer Utility

The InterSystems EPROM Program Utility should work with the EPROM Programmer board in any CP/M based, S-100 bus system with a minimum of 64 kbytes system memory.

WARNING

NEVER ATTEMPT USING THE EPROM PROGRAMMER UTILITY WITH A MEM TESTING BIOS.

If you are using the InterSystems Cache BIOS, it is necessary to use the ISET program to set the BIOS to the "Polled" mode of operation rather than the "Interrupt" mode. It is a feature of the interrupt mode that memory tests are performed whenever the system is idle. A description of the Cache Bios and ISET is located in the System Overview manual that accompanied your system.

EPROMs that are to be programmed should be verified as being thoroughly erased as specified by the EPROM device data sheets. An erasure check (for all FFs) is automatically done by the Programmer utility before programming.

InterSystems recommends that the Programmer board burn socket address be set at 8000H as described in section 2.3 of this manual.

In any case that a file is being read from or written to system memory, check that the location in RAM does not overlap the burn EPROM address.

3.2.2 Memory Map

Figure 6 illustrates the memory map.

3.2.3 Taking Care of the Utility Floppy Disk

The EPROM Programmer Utility was shipped to you on either a 5.25-inch or a 8-inch floppy disk. All software distribution disks supplied by InterSystems are write-protected. Never write onto your distribution disk. While the write-protect feature protects the disk from being inadvertently written-over by the user, you should protect the disk from any other damage. Since floppy disks are a moveable media, the distribution disk is prone to unexpected damage, as well as ordinary wear and tear. For this reason you are urged to create a "working" disk and store the distribution disk in a safe place. You should also plan to maintain a back-up copy of your work during a normal working session. The back-up copy should contain any files you are working with, in addition to the utility.

3.3 USING THE EPROM PROGRAMMER UTILITY

The following discussion is organized in the format that you will encounter program execution during a typical programming session. We have attempted to provide you with a complete description of utility prompts. Prompts are shown exactly as they are displayed on a video terminal. After each prompt, an indented paragraph describes the prompt, offers a description of the possible replies, and recommends a reply, if appropriate. In many cases you may need to enter a different response.

Responses to menu numbers in this program should be followed by a carriage return. Throughout this manual, the carriage return key is referred to as <cr>. The program accepts uppercase and lowercase letters for all responses. Generally speaking, any selection other than menu numbers uses the following syntax.

- If no information is provided, enter the required information and press <cr>. The program will subsequently default to the new specification until changed or the system is rebooted.

- If the information displayed (such as a start address) is correct, press <cr> to continue.

- If the information displayed in a prompt is not correct, enter the new information and enter <cr> to continue.

- If the prompt allows a (y/n) choice, YES can be selected by entering either

3.4 MENU OPTION SUMMARY

In section 3.2 and 3.3 we provided a general description of the EPROM Programmer utility's features. (If you haven't read these sections, go back and read them before proceding with this section.) In this section we provide details about using the two menus and options that control the features. Included is a discussion of the Main Option Menu and the Buffer Menu.

At any time information about the ERPOM being programmed or being read is required, the appropriate prompt will be displayed. If a value has been provided for the prompt earlier in the programming session, the value will be displayed as the default value.

The utility contains a catalog of the programming parameters for each of the EPROMs supported by the EPROM Programmer board and utility. In any case that an EPROM-type not supported by the utility is being programmed or read, prompts for the required information will be automatically displayed.

In most cases, all that is required of you is entry of the correct EPROM-type number for the EPROM to be programmed, and checking that the correct personality module is installed.

Each option of the Main Option Menu will have specific requirements for information. The utility will provide you with the prompts required to collect the necessary information. For your convenience, the description of the Main Option Menu is organized in six sections. Each section details the specific prompts that follow the selection of each of the primary Main Option Menu options.

You will note that for certain main options, there will be a long series of prompts.

The Buffer Menu will be displayed after you transfer code to the buffer. After you make a selection from the Buffer Menu, additional prompts will be provided so that you can tailor the procedure as required.

PRELIMINARY

3.4.1 Main Option Menu

The Main Option Menu is reproduced below as it appears on a terminal display. A summary of the seven options follows. Each menu option is then described in detail.

Call the EPU utility by typing the following at the keyboard.

```
      EPU<cr>
```

When the program signs on it offers the following "main menu" of available options.

```
      Main Options

      1) Copy from disk COM file to EPROM
      2) Copy from disk HEX file to EPROM
      3) Copy from memory to EPROM
      4) Copy from EPROM to EPROM
      5) Copy from EPROM to memory
      6) Copy from EPROM to Disk
      7) Exit Program

       Choose (1-7) -->
```

Note that the Main Option Menu contains four options for burning files into a target EPROM, three options for reading the contents of a programmed EPROM and transferring that code, and the Exit Program option that returns control to the operating system. The options are described briefly below. A detailed description of each option follows.

Summary of Main Option Menu --

1) Copy from disk COM file to EPROM

Option 1 is selected if the file to be programmed in the EPROM is a COM file located on a disk. Note that only legal COM files can be entered.

2) Copy from disk HEX file to EPROM

Option 2 is selected if the file to be burned into EPROM is a HEX file. Multiple HEX files, up to buffer size, are allowed. The Programmer Utility will provide the necessary prompts to you such that you can specify the offset to zero the buffer.

3) Copy from memory to EPROM

Option 3 is selected if the code to be burned into EPROM is located in system memory. Figure 6 above indicates the area of system memory in which the HEX code can reside.

4) Copy from EPROM to EPROM

Using this option, you can copy the contents of one EPROM into a blank EPROM. The EPROMs can be different types. Insert the programmed EPROM in the ZIF socket, write the PROM code into a buffer, and remove the PROM. You then insert a blank PROM into the ZIF socket and burn the code from the buffer into the blank EPROM.

5) Copy from EPROM to memory

Option 5 allows you to read a programmed PROM, and write the code into a specified location in system memory. See Figure 6.

6) Copy from EPROM to Disk

Using option 6, code from a ROM (or programmed PROM) inserted in the board burn socket can be written out to a disk file.

7) Exit Program

Returns To CP/M operating system.

Options 1 through 4 above are used to program blank EPROMs. Options 4 through 6 provide a procedure for copying the contents of an EPROM to either a blank EPROM, system memory, or a disk file, respectively.

Once the option number has been typed, (followed by a <cr>,) the Utility program will provide the appropriate series of prompts to take you through a complete procedure. The description of each option follows.

Detailed Description of Main Menu Options --

This section describes the specific prompts that the utility will generate to collect the information it requires to accomplish a menu option.

Option 1 - Copy from disk COM file to EPROM --

If option 1 is selected from the Main Option Menu by entering

```
      1<cr>
```

the following set of prompts is provided.

```
Low address of EPROM burn socket -->
```

> Enter "8000" (assuming that the burn EPROM base address was set at 8000H during board setup - section 2.2: SW3). This prevents the program from overwriting the EPROM burn address space.

```
Copy from disk COM file to EPROM

Make sure disk is in drive

Filename -->
```

> Enter the legal CP/M COM filename of the file you are copying from disk to the EPROM. If the file resides on a disk other than the currently logged "default" disk drive, you must designate which drive you are reading form. For example, B:filename.COM <cr> indicates that the file resides on disk drive B in the system.

```
Continuing with transfer...

Only 8192 bytes of the file fit in the buffer.
 8192 bytes of the file have been transferred so far.
```

> This prompt appears only if the COM file is larger than 8 kbytes (the capacity of the 8 kbyte buffer). If the COM file being transferred is greater than 8 kbytes, the program will transfer the first 8 kbytes into the buffer, guide you through the procedure of programming the buffer data into an EPROM, and then return to transfer the rest (or the next 8 kbytes) of the COM file from the disk drive to the buffer.

PRELIMINARY

Option 2: Copy from disk HEX file(s) to EPROM

Low address of EPROM burn socket -->

Enter '8000' (assuming that the EPROM burn socket address was set at 8000H during board setup.) This prevents the program from overwriting the EPROM burn socket address space.

Copy from disk HEX file/s to EPROM
Enter lowest org for all the HEX files -->

The "org" is the address at which the file is assembled. This is usually 100H.

You should enter the address at which the first items of code should appear within the EPROM being programmed. For example, in many cases you may want the HEX files to start at address 100H of the EPROM. In this case you would enter 0100. For a second example, take the case in which you would want the HEX files to start at an address 200H above the base address of the EPROM. In this case you would enter 0200.

Zeroing buffer...

This means that zeros are written to all memory locations in the buffer.

Make sure disk is in drive.

Filename -->

Enter the legal CP/M HEX filename of the file you are going to program into ERPOM. If the file resides on a disk other than the currently logged "default" disk, you must designate the disk. For example, if the HEX file is located on a disk in disk drive B, you should enter

```
B:filename.HEX<cr>
```

At this point you have transferred the contents of the first HEX file into the 8 kbyte buffer. The utility will now provide the opportunity to specify additional HEX files.

At this time the Buffer Menu is displayed. The Buffer Menu will provide a series of options that allow you to: 1) proceed with programming an EPROM with the code in the buffer, 2) display the contents of the buffer, 3) compare the contents of the buffer to the contents of an EPROM, or 4) return to the Main Option Menu. You can refer to the "Buffer Menu" section below for an expanded description of these options.

```
Option 3: Copy from memory to EPROM

 Low address of EPROM burn socket -->
```

Enter '8000' (assuming that the EPROM burn socket address was set at 8000H during board setup). This prevents the program from overwriting the EPROM burn address space.

```
Copy from memory to EPROM

Memory block # 1
   Low address -->
   High address -->
```

Enter the low and high address boundaries of the memory block being transferred.

```
Moving information into buffer...Finished
Do you have another memory block?  (y/n) -->
```

These prompts inform you that the first block transfer is complete and allows you to elect the transfer of additional blocks of memory. Enter "y<cr>" or <cr> if additional code in system memory is to be transferred. If "y" is selected, the above prompts for high and low address boundaries will be repeated.

The blocks are transferred sequentially, and without any gaps, into the buffer.

Enter "n" when all blocks of memory have been transferred. When "n" is selected, the following prompt will be displayed.

```
Transfer of information from memory block into buffer is
finished.
You are free to remove your disk if you wish.
```

At this time the Buffer Menu is displayed. The menu gives you the following options: 1) to program an EPROM with the contents of the buffer, 2) to display the code in the buffer, 3) to compare the contents of the buffer with the contents of an EPROM, or 4) to return to the Main Option Menu. Refer to the "Buffer Menu" section below for a complete description of the menu options, and examples.

PRELIMINARY

Option 4: Copy from EPROM to EPROM

This option allows you to copy the contents of one or more EPROMs into a blank EPROM. The EPROMs can be different types. Remember, though, that you may have to change personality modules. If you are reading from one EPROM type, and programming a second EPROM type, you must change personality modules. The personality module must match the EPROM in the ZIF socket, for both the read and the program procedures.

You will insert the programmed EPROM in the ZIF socket, write the EPROM code into a buffer, and remove the EPROM. You can then insert a blank EPROM into the ZIF socket and burn the code from the buffer into the blank EPROM.

After entering

```
4<cr>
```

as a response to the Main Option Menu, the following prompt will be displayed:

```
Low address of EPROM burn socket -->
```

Enter 8000 (assuming that the EPROM burn socket address was set at 8000H during board setup). This prevents the program from overwriting the EPROM burn address space.

```
Copy from EPROM to EPROM

EPROM # 1
Enter EPROM type #
```

Enter the EPROM type. For example, 2716<cr>, 2732<cr>, etc.

```
  Low address:  8000 -->
  High address:  XXXX -->
```

Based upon the low address of EPROM burn socket (entered above) and the EPROM type, the low and high addresses will be computed. If you wish to transfer only a specific block of code from the EPROM, enter the desired low and high address parameters.

Set SW1 to READ ONLY. RED lamp should be OFF

. SW2 to SAFE TO REMOVE. GREEN lamp should be ON

```
  Please insert the EPROM.
     (NOTE:  24-pin chips must be LEFT justified in the socket)

After EPROM is in place...

Set SW1 to READ ONLY.  RED lamp should be OFF

Set SW2 to ON.  GREEN lamp should be OFF

  Type <cr> when EPROM is inserted and ready to be read.
```

When you type the carriage return, the data will be transferred into the buffer.

```
  moving data into buffer...  Finished.
Do you have another EPROM? ( y/n )
```

If you desire to transfer data from additional EPROMs, enter "y<cr>" or "<cr>". You can continue to transfer data from EPROMs until the message

```
      Buffer is full
```

is displayed.

If not, enter "n<cr>.

```
Transfer of information from EPROM into buffer is finished.
```

The Buffer Menu will now be displayed providing a series of options that allow you to: 1) proceed with programming an EPROM with the in the buffer, 2) display the contents of the buffer, 3) compare contents of the buffer to the contents of an EPROM, or 4) return the Main Option Menu.
You can refer to the "Buffer Menu" section below for an expanded description of these options.

PRELIMINARY

Option 5: Copy from EPROM to memory

```
5) Copy from EPROM to memory
```

Option 5 allows you to read a programmed PROM, and write the code into a specified location in system memory. See Figure 6 for an illustration of the memory area that is available to write code to.
EPROM # 1 Enter EPROM type #

Enter the EPROM type. For example, 2716<cr>, 2732<cr>, etc.

```
  Low address:  8000 -->
  High address:  XXXX -->
```

Based upon the low address of EPROM burn socket (entered above) and the EPROM type, the low and high addresses will be computed. If you wish to transfer only a specific block of code from the EPROM, enter the desired low and high address parameters.

```
Set SW1 to READ ONLY.  RED lamp should be OFF

SET SW2 to SAFE TO REMOVE.  GREEN lamp should be ON

  Please insert the EPROM.
    (NOTE:  24-pin chips must be LEFT justified in the socket)

After EPROM is in place...

Set SW1 to READ ONLY.  RED lamp should be OFF

Set SW2 to ON.  GREEN lamp should be OFF

  Type <cr> when EPROM is inserted and ready to be read.
```

When you type the carriage return, the data will be transferred into the buffer.

```
  moving data into buffer...  Finished.
Do you have another EPROM? ( y/n )
```

If you desire to transfer data from additional EPROMs, enter "y<cr>" or "<cr>". You can continue to transfer data from EPROMs until the message

```
Buffer is full
```

is displayed.

If not, enter "n<cr>.

Transfer of information from EPROM into buffer is finished.

The Buffer Menu will be displayed next.
providing a series of options
that allow you to: 1) proceed with transferring the code in the bu
to a designated area (block) of system memory,
2) display the contents of the buffer, 3) compare the
contents of the buffer to the contents of an EPROM, or 4) return t
the Main Option Menu.
You can refer to the "Buffer Menu" section below for an
expanded description of these options.

```
Option 6: Copy from EPROM to disk

Low address of EPROM burn socket -->
```

Enter '8000'. The burn EPROM socket address was set at 8000H during board setup.

```
Copy from EPROM to disk

EPROM #1
Enter IC # of EPROM -->
```

Enter the EPROM type (i.e., 2732, 2716, etc.). This prompt is an item from the Miscellaneous Information Menu. The utility will compare the entered IC# to its record of IC types to obtain required parameters. If the IC# entered does not exist in the utility record, you will be asked a series of questions about the EPROM type. Based on the the EPROM type specified, and the low address of the burn socket specified above, the utility will generate the following display.

```
Low address: 8000 -->
High address: XXXX -->
```

The "high address" is determined by the utility, based on which EPROM type is entered in the step above, together with the specified "low address". You can enter your own low and high address limits if you wish to only read a portion of the EPROM space.

```
Set SW1 to the READ ONLY position. The RED light should be OFF.

Set SW2 to the SAFE TO REMOVE position. The GREEN lamp should be

Please insert the EPROM.
   (NOTE: 24 pin chips should be LEFT justified in the socket)

After EPROM is in place...

Set SW1 to the READ ONLY position. The RED light should be ON.

SET SW2 to the SAFE TO REMOVE POSITION. The GREEN light should be

Type <cr> when EPROM is inserted and ready to be read.
```

Enter carriage return.

```
Moving information to buffer...Finished.
```

If the information to be transferred from the EPROM is enough to fill the buffer the following message will be displayed.

```
The buffer is full.

Transfer of information from EPROM is finished.
You are free to remove your disk if you wish.
```

At this time the Buffer Menu is displayed. The Buffer Menu will provide a series of options that allow you to: 1) proceed with transferring the code in the buffer to a disk file, 2) display the contents of the buffer, 3) compare the contents of the buffer to the contents of an EPROM, or 4) return to the Main Option Menu. You can refer to the "Buffer Menu" section below for an expanded description of these options.

Option 7: Exit Program

This option will return you to CP/M.

3.4.2 Buffer Menu

After a programming procedure is selected from the Main Menu, and basic information has been supplied in response to the utility prompts, the Buffer Menu is displayed on the screen. In the case of main menu options 1 through 4, the following element of the Buffer Menu is displayed.

```
  Current Options

     1) Move buffer to EPROM
     2) Display buffer
     3) Compare buffer to an EPROM
     4) Return to main menu
```

In the case of main menu options 5 and 6, the first item of the Buffer Menu is different.

Main Menu option 5 will have a Buffer Menu option 1 of:

```
     1) Move buffer to memory
```

and Main Menu option 6 will have a Buffer Menu option 1 of:

1) Move buffer to disk

The four options of the Buffer Menu are decribed in the pages that follow. A complete description of the steps required for each item on the menu is provided. Because three separate versions of the first option are provided, depending on the destination for the contents of the buffer, each will be discussed fully before the second through fourth options of the Buffer Menu are described. Therefore, this section will be organized as follows:

First 1) Move buffer to EPROM (Main Menu options 1-4)
1) Move buffer to memory (Main Menu option 5)
1) Move buffer to disk (Main Menu option 6)
Then 2) Display buffer
3) Compare buffer to an EPROM
4) Return to main menu

1) Move buffer to EPROM --

If "1) Move buffer to EPROM" is selected (during Main Menu procedures 1 through 4) the utility program will guide you through the procedure of burning code located in the buffer into a specified EPROM. A series of prompts (part of the Miscellaneous Information Menu) will be provided to obtain necessary information about the EPROM being programmed.

The utility will allow you to create an odd/even split of the code, so that even-addressed bytes are burned into one EPROM and odd-addressed bytes are burned into a second EPROM.

The utility will automatically lead you through the steps required to burn code into a series of EPROMs if the code (disk file, EPROM contents, etc.) exceeds the capacity of the EPROM being used. For example, a 5 kbyte disk file might be burned into two consecutive 4 kbyte EPROMS; 4 kbytes would be burned into the first EPROM, and the remaining 1 kbyte of code would be burned into a second EPROM.

Once option 1 of the Buffer Menu is selected, the following prompt will be displayed:

```
Is this an odd/even split?  (y/n)
```

> The odd/even split feature allows you to burn a single file into two PROMs: the first PROM receives the even bytes (ZERO is defined as even) and the second PROM receives the odd bytes.
>
> Enter either 'y' or 'n'.
>
> If a 'y' is entered, the following prompt appears-

```
I will start with the even bytes...
```

> -reminding you that even bytes will be burned into the first blank EPROM. After even addressed bytes are burned into the first EPROM, the EPROM is removed, a second blank EPROM is inserted, and odd addressed bytes are burned into the second EPROM.
>
> If you enter "no<cr>", the program will proceed.

```
There are 8192 bytes of information remaining in the buffer.
```

> In this example the buffer is full. The number of bytes listed is an indication of the file length in

the buffer.

```
Enter IC# of EPROM -->
```

This prompt is an item from the Miscellaneous Information Menu. The utility will compare the entered IC# to its record of IC types to obtain required burn parameters. If the IC# entered does not exist in the utility record, you will be asked a series of questions about the EPROM-type to determine the relevent characteristics of the EPROM. For example, if you entered an EPROM type not recognized by the utility, the following prompt will be displayed.

```
More information is needed:
  Enter number of burns ( 1 to 200 ) -->
```

To which you should respond by entering the required number of burn iterations required by the EPROM.

```
  Enter total size of EPROM in bytes -->
```

Note that the size is to be stated in bytes. For example, an 8 kbyte EPROM would require the entry 8192. A 4 kbyte EPROM is equivelent to 4096 bytes. A 2 kbyte EPROM is equivelent to 2048 bytes. A 1 kbyte EPROM is equivelent to 1024 bytes.

```
  Low address: 8000 -->
  High address: XXXX -->
```

The low address value is a reiteration of the address you entered as the first step of the procedure. The high address is computed by the utility program.

If you have entered a standard EPROM type, the following message will be displayed.

```
Low address in this EPROM where burn will start: 8000 -->
High address where burn will end: XXXX -->
```

This information is provided by the utility program. The burn start address is derived from the "low address of EPROM burn socket" query earlier in the procedure. The burn end address is calculated by the utility program.

```
EPROM too small for all info - you will need a total of
```

PRELIMINARY

more bytes of EPROM to burn the rest of the information.

You will see this message only if the file (block of code, etc.) is too long to be burned into one EPROM of the type designated. The programmer utility will provide the correct prompt sequence to insert and burn additional EPROMs at the appropriate time.

Set SW1 to READ ONLY. The RED lamp should be OFF.

Set SW2 to SAFE TO REMOVE. The GREEN lamp should be ON.

Please insert the EPROM.
(NOTE: 24-pin chips must be left justified in the socket)

The above steps make the board safe for installing an EPROM in the ZIF socket.

Type <cr> when EPROM is inserted in burn socket...

Set SW1 to PROGRAM POWER. The RED lamp should be ON

Set SW2 to ON. The GREEN lamp should be OFF

Type <cr> when ready to burn...

The above steps prepare the board to program the EPROM. The utility will now automatically check to ensure that the inserted EPROM is completely erased (all FFs).

Entering EPROM check-

The EPROM is checked to ensure that it is completely erased (i.e., all FFs).

Area of burn is all FF's

This message indicated that the EPROM is completely erased.

Any byte that is not FF is identified as shown.

```
Entering EPROM check -
Error at address: 8000  byte =F5
Error at address: 8001  byte =2A
Continue display? ( y/n )
```

If y is selected, additional errors will be displayed. If no is selected, the program provides the following prompt:

```
Enter 1) Continue with burn; 2)Insert new EPROM; 3)Main Men
```

If option 1) is selected, the burn procedure will continue in spite of the fact that the check has indicated the EPROM is not completely erased. This option might be selected, for example, if the code being burned into EPROM was not going to conflict with the code to be transferred.

If option 2) is selected, the correct procedure for replacing the EPROM with another blank will be provided. The instructions are similar to those already described so they will not be repeated here.

If option 3) is selected, the Main Menu will be displayed. This will allow you to start over or exit the EPROM Programmer Utility.

If the EPROM check indicates that all EPROM address locations are erased or the "continue with burn" option is selected, the next prompt will be:

```
Entering EPROM burn -
EPROM size is  XXXX
Burn start address is 8000
Burn end address is XXXX
Number of bytes to be burned is XXXX
```

Where the values are provided as a review of information entered for earlier prompts.

```
Set SW1 to PROGRAM POWER.  RED lamp should be ON

Set SW2 to ON.  GREEN lamp should be OFF

 Ready to burn? ( y/n )
```

Enter either "y<cr>" or "no<cr>".

If you have selected "y", the program will display the word "programming", followed by a series of periods. The periods are an indication to you that the utility is still involved in the programming process. There will be a variable length delay, depending upon the particular EPROM-type programming requirements.

```
pgramming..............................

Entering EPROM verify-
Burn successful on try # 1
Do you want to repeat this burn? ( y/n )
```

This set of prompts indicates the completetion of the burn cycle, and that the burn was succesful. If the burn was successful, you must select whether or not to repeat the burn. If you enter "y<cr>" you can make another copy of the same EPROM without inputing the parameters again. If you enter "n<cr>", the following series of prompts will be displayed.

```
Set SW1 to READ ONLY.  RED lamp should be OFF

Set SW2 to SAFE TO REMOVE.  GREEN lamp should be ON

  You may now remove the EPROM from the board.
```

The Main Option Menu will be displayed.

If the first burn is unsuccesful the utility will automatically try the burn a second time. If the second attempt to burn an EPROM is also unsuccessful the following message will be displayed.

```
Burn has failed twice - try another EPROM
```

You can attempt programming another EPROM, or you can attempt to reprogram the first EPROM. If the verify finds a byte that does not agree with the code in the buffer, an error message will be displayed.

PRELIMINARY

1) Move buffer to memory --

Note that this description of the Buffer Menu option 1 refers only to Main Menu option 5 (copy from EPROM to memory). For the explanation for this system of organization refer to section 3.4.1 above. If you are looking for the Buffer Menu option 1 for Main Menu options 1 through 4, refer to the preceding pages. If you are looking for instructions for moving the buffer to a disk, it follows this discussion.

This description assumes that in response to the prompt

```
Current options:
  1) Move buffer to memory
  2) Display buffer
  3) Compare buffer to an EPROM
  4) Return to main menu

  Choose option ( 1-4 ) -->
```

that you have selected option 1. The next prompt you will see is:

```
Low address of destination memory -->
```

You should enter the low address of the memory block to which you wish to move the buffer contents. Assuming that the address you select is appropriate, the following prompt will be displayed.

```
programming.............
  ..... Finished.
```

At this point, the contents of the buffer have been transferred to the assigned area of memory. The Main Menu will be displayed, giving you the option to continue, or exit the program.

1) Move buffer to disk --

Note that this description of the Buffer Menu option 1 applies to Main Menu uption 6 only (copy from EPROM to disk). For the explanation for this system of organization refer to section 3.4.3 above.
This description assumes that in response to the prompt

```
Current options:
  1) Move buffer to disk
  2) Display buffer
  3) Compare buffer to an EPROM
  4) Return to main menu

  Choose option ( 1-4 ) -->
```

that you have selected option 1. The next prompt you will see is:

```
Filename -->
```

Enter the desired legal filename. You can specify disks other than the default disk if desired. For example, you could transfer the contents of the buffer to a file called SAMPLE.COM on disk drive B with the instruction:

```
B:SAMPLE.COM<cr>
```

Following the creation of the file the Main Menu will be displayed, allowing you to either continue using the utility program or to exit the program.

2) Display buffer --

If option "2" is selected from the Buffer Menu, (for any of the six Main Menu functions) the contents of each address location of the buffer is displayed. The contents will be presented in display-screen length segments, 20 lines each. Each line of the display will have the address of the first byte of the line followed by sixteen pairs of Hex characters. Each display will provide the option of continuing to display the buffer or quit. For example, a sample display might look like this:

```
  1: 2A 06 00 2B F9 C3 08 01 0E 19 CD 05 00 2A 06 00
 17: F3 08 20 A3 FF 01 19 06 32 72 D3 04 E7 D2 31 27
 33: AD 03 20 00 FF C3 2A 06 2A 06 C5 CC 33 CD DA 19
 49: FF 01 19 06 32 F3 08 20 A3 72 D3 04 E7 D2 31 27
 65: D4 42 3A D3 0E 65 63 21 F1 69 53 65 6C 18 21 DD
 81: 6E 67 4D 20 D2 7A 20 4E 66 70 74 A0 19 FD 06 02
 97: CD 03 20 00 FF C3 2A 06 2A 06 C5 CC 33 CD DA 19
113: F7 F7 6C 53 74 61 E5 E0 01 CD 68 72 72 66 F7 8A
129: AD 03 20 00 2A 06 FF C3 2A 06 C5 CC 33 CD DA 19
145: F3 A3 FF 01 08 20 19 06 32 72 D3 04 E7 D2 31 27
161: AD 03 20 00 FF C3 2A 06 2A 06 C5 CC 33 CD DA 19
177: F3 08 20 A3 FF 01 19 06 32 72 D3 04 E7 D2 31 27
193: D4 42 3A D3 0E 65 63 21 F1 69 53 65 6C 18 21 DD
209: 08 F3 F3 08 20 F3 08 20 FF 6E 67 C3 08 20 A3 20
225: 32 42 2F 3A 04 00 20 C3 DD 6E D3 67 C3 A3 08 02
241: FF 20 00 FF C3 2A 06 2A AD 03 20 00 2A 06 08 20
257: D4 3A 42 0E 63 21 D3 65 F1 69 65 6C 53 18 21 DD
273: FF C3 AD 03 20 00 2A 06 2A 06 C5 DA CC 33 CD 19
289: AD 03 06 C5 CC 33 CD DA 20 00 FF C3 2A 06 2A 19
305: D4 42 3A D3 0E 65 63 21 F1 69 53 65 6C 18 21 DD


Continue display ( y/n )?
```

If you enter "y<cr>", the next section of the buffer will be displayed. If you enter "n<cr>" the next display will be:

```
Current Options:
  1) Move buffer to EPROM
  2) Display buffer
  3) Compare buffer to an EPROM
  4) Return to the main menu
```

The Buffer Menu will be displayed.

3) Compare buffer to an EPROM --

If 3) is selected, a series of prompts will be provided to allow the buffer to be compared to the selected EPROM, starting at the burn-EPROM base address location.

If the buffer contents match the EPROM contents the message

```
EPROM is the exact match of buffer
```

will be displayed.

If the buffer contents do not match the EPROM contents, a message will be displayed. For example, if the first two bytes of the buffer do not match the first two bytes of the EPROM, the first three lines of the message would look like this:

```
Entering EPROM verify-
Address = 8000 EPROM = F5 Buffer = 2A
Address = 8001 EPROM = AF Buffer = FF
```

> In this example, the contents of EPROM addresses 8000H and 80001H do not match the contents of the buffer.

4) Return to main menu --

If 4) is selected, the routine for returning to the main menu will be started. See section 3.4.1.

Note that the contents of the buffer are lost if 4) is selected.

4.0 PROGRAMMING WITH USER-GENERATED SOFTWARE

Burn software may reside anywhere in RAM. The software can be elaborate or simple, but probably should accomplish the following:

1) Check to see if the EPROM is "clear": if all bits are unwritten. A 2708, for instance, should read all FF Hex.

2) Read the original data and write it to the burn EPROM, byte by byte, either once, or repetitively, depending upon the EPROM type.

3) After the appropriate number of read/write cycles, check the burn EPROM against the source data, and verify that there is a match.

The second function above, the most essential, can be performed by many simple monitors that include a block move instruction if the EPROM type does not require a repetitive sweep. 2704, 2708, and the TI2716JL EPROMs require repetitive sweeps; that is, the program should read and write through the whole data block, byte by byte, and then repeat the same process over again, for a certain number of times. The actual number of times required depends upon the manufacturer's specification, but 2708s, for instance, are properly programmed after 200 iterations of a block move using the InterSystems EPROM Programmer utility. The 2758, 2716 (not manufactured by Texas Instruments), 2732, TMS2532, or 2764 variety EPROMs require only a single sweep.

All other critical timing parameters such as setup and hold times, programming voltage dwell time, and wait states are provided for by hardware.

The original data may conveniently reside in a disk file which can be loaded into memory before the burn software is executed.

PRELIMINARY

5.0 FRONT PANEL PROGRAMMING

If the microcomputer system includes a front panel (such as the InterSystems Front Panel), the very simplest kind of EPROM programming may be conducted simply by executing front panel deposits of the desired data into the burn EPROM address space for EPROM types that require a single write sequence for each byte to be burned. The front panel approach will frequently be useful when only a few bytes need to be written or altered in an EPROM. A bit can be changed from 1 to 0 without erasing the EPROM (though erasure of the EPROM is neccesary to change a bit from 0 to 1). Such a scheme is not practical with the 2704, 2708, and TI2716JL EPROMs, since these types require repetitive sweeps of data.

1) Set SW3 and SW4 as necessary (extended, and burn EPROM address, respectively) before inserting the board into the bus.

2) With system power set to OFF, install the EPROM Programmer board.

3) Insert the appropriate personality module for the type of EPROM to be programmed. See section 1.3.

4) Set SW2, the SAFE TO REMOVE/ON switch, to SAFE TO REMOVE, and insert the EPROM to be burned.

5) Set the SAFE TO REMOVE/ON switch, SW2 to ON.

6) Using the front panel functions, examine the memory location (within the burn address space) to be programmed. Displayed data should be FF, indicating erasure.

7) Set the front panel data switches for the data byte to be programmed.

8) Set SW1, the READ ONLY/PROGRAM POWER switch, to PROGRAM POWER.

9) Press the front panel DEPOSIT switch.

10) To program the subsequent memory location, set the data switches for the new byte, and press the front panel DEPOSIT NEXT switch. To program other locations within the burn address space, repeat steps 5, 6, and 7.

PRELIMINARY

11) To stop programming, set SW1 to READ ONLY, and SW2 to SAFE TO REMOVE. Remove the EPROM.

2-26-82

TO: W. Stark, E. Pankowski, C. Fritz, T. Bond, J. Faller
FROM: J. Carson
RE: Eprom Programmer Personality Module Assembly Dwg/B.O.M.

This document describes the correct assembly of the various personality modules in the set available for the eprom programmer. Revision level is first production release. Where applicable, a drawing is furnished. In most cases, a jumper or component is called out on the B.O.M. The installation of the board connector and keying pin is assumed for each of the following types:

Board assemblies fabricated from #1606 (3-supply eproms):

* 2708: assy #001623
Install bare wire jumper #24 bent to 0.1" hole spacing at 3, 7, 10.
Install insulated #24 jumper wire bent to resistor spacing at D1.
Install 0.1 uF dipguard at C1, C2, C3
Install 0.001 uF dipguard at C4, C5, C6

* 2704: assy #001622 (no order anticipated for this type)
Same as 2708 except cut default jumper at "A"

* T.I. 2716: assy #001624 [illegible]
Install bare wire jumper at 2, 4, 5, 8, 9, 11
Cut default jumper at "B"
Install 0.1 uF dipguard at C1, C2, C3
Install 0.001 uF dipguard at C4, C5, C6
Install 2K 5% 1/4W carbon comp at R1
Install 1N4148 at D1

Following board assemblies fabricated using #1607 (all single +5 volt style):

* 2516- Intel 2716 (5V only)- 2758- 2508: assy #001633
(note that the 1631 assy # is therefore no longer needed)
Install bare wire jumper at 4, 13, 15, 21, 23, 27, 31, 36
Install 0.1 uF dipguard at C1

* TMS 2532: assy #001634
Install bare wire jumper at 4, 12, 13, 16, 21, 22, 29, 31, 36
Install 0.1 uF dipguard at C1

* 2732: assy #001635
Install bare wire jumper at 7, 8, 12, 13, 16, 24, 28, 31
Install 100 ohm 5% 1/4 W carbon comp at R1
Install 0.1 uF dipguard at C1
Install 100 pF ceramic disc at C3

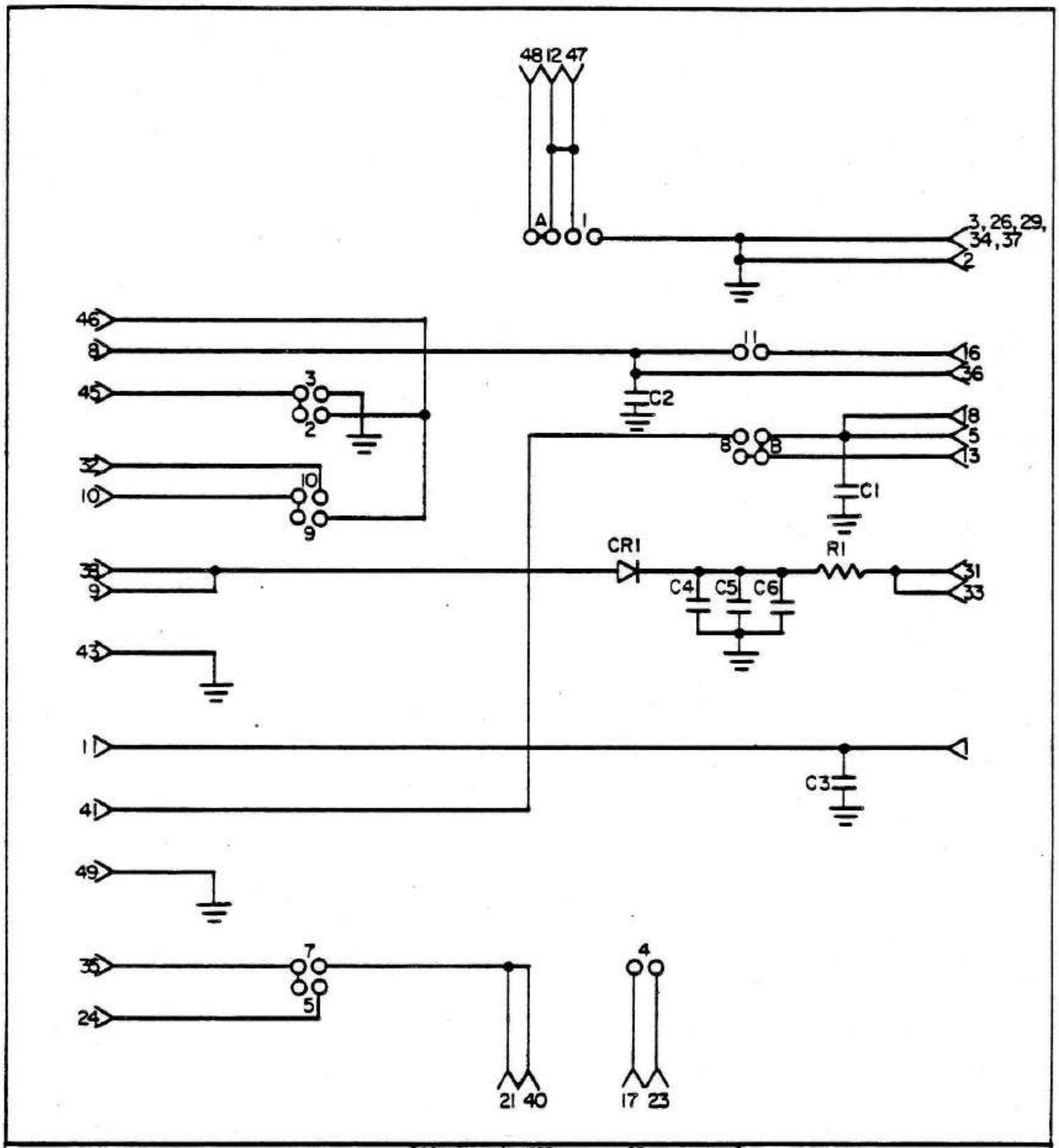
* 2732A: assy #001636 (this could be merged with type 2732 in next art rev if a shunt jumper is used for 21v)
Same as 2732 except add 95.3K 1% at R3

* 2764: assy #001637 (see assembly drawing)
parts required: 0.1 uF dipguard=C1, 100 pF disc ceramic as noted, 95.3K 1% at R3, 100 ohm 5% 1/4 W carbon comp at R1, bare wire jumper at 4, 14, 16, 23, 31, 36; three insulated wire jumpers as shown on print

* TMS 2564: assy #001638
Install 0.1 uF dipguard at C1
Install 100 ohm 5% 1/4W carbon comp at R2
Install bare wire jumpers at 1, 4, 12, 14, 16, 20, and 22, 31, 32, 35

* MCM68764: assy # 001639
Install 0.1 uF dipguard at C1
Install 100 ohm 5% 1/4W carbon comp at R2
Install bare wire jumpers at 6, 8, 12, 14, 16, 24, 29, 31

attachments: 2764 personality module assembly dwg. all others self-explanatory. Ed: this time I have enclosed copies of the silkscreen so that Mike can make sketches with colored pencil similar to what he does for other assemblies. For engineering configuration control purposes, I feel that a Bill Of Materials (B.O.M.) will suffice. If you wish more comprehensive documentation for the other modules, let me know.



REVISIONS			
DESCRIPTION	DATE	APPROVED	

SCALE — SHEET 1 OF 1

DRAWN cls DATE 1/18/82

CHECKED DATE

UNLESS OTHERWISE NOTED

FRACTIONS ± — .00 ± — .000 ± — ANGLES ± —

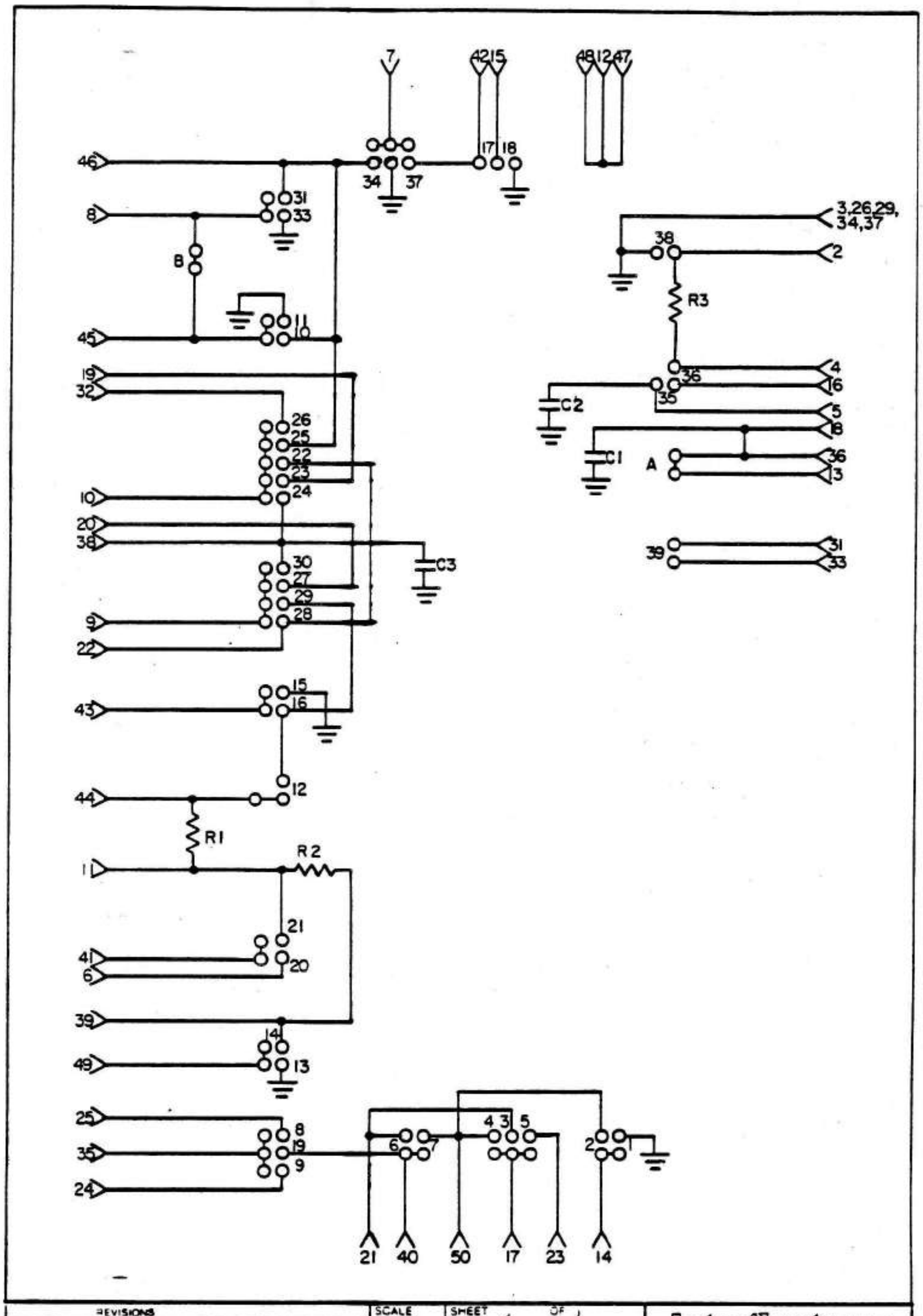
RADII — DRAFT — FINISH —

InterSystems

ITHACA, NEW YORK 14850

PROJECT EPROM BURNER PERSONALITY MODULE

DATE 1-18-82 NUMBER 1606



REVISIONS

LTR	DESCRIPTION	DATE	APPROVED

SCALE — SHEET 1 OF 1

DRAWN uk DATE 1/18/82

CHECKED DATE

UNLESS OTHERWISE NOTED

FRACTIONS ± — .00 ± — .000 ± — ANGLES ± —

RADII — DRAFT — FINISH —

InterSystems

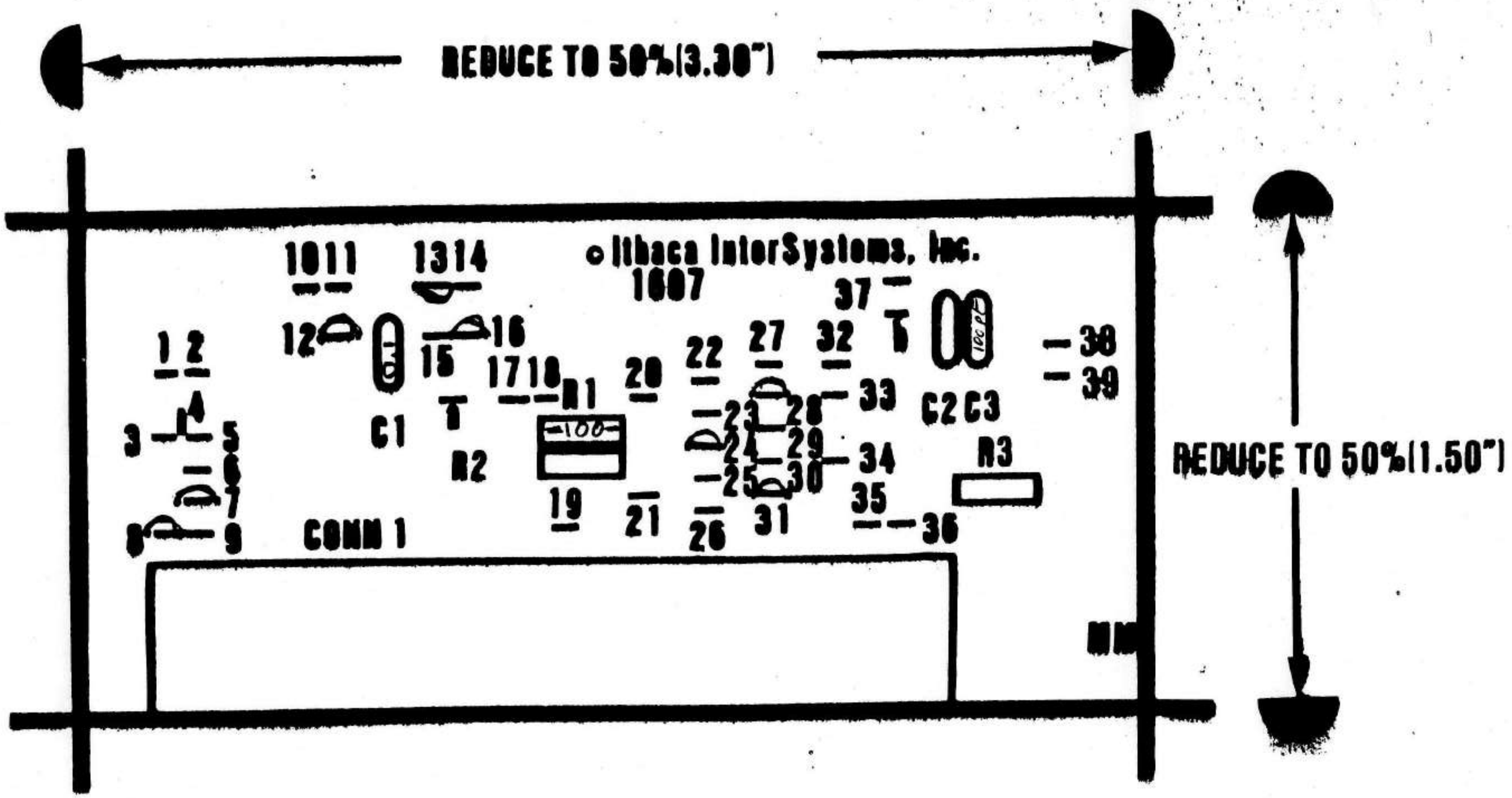
ITHACA, NEW YORK 14850

PROJECT EPROM BURNER PERSONALITY MODULE

DATE 1-18-82 NUMBER 1607

INTEL 2732
2732

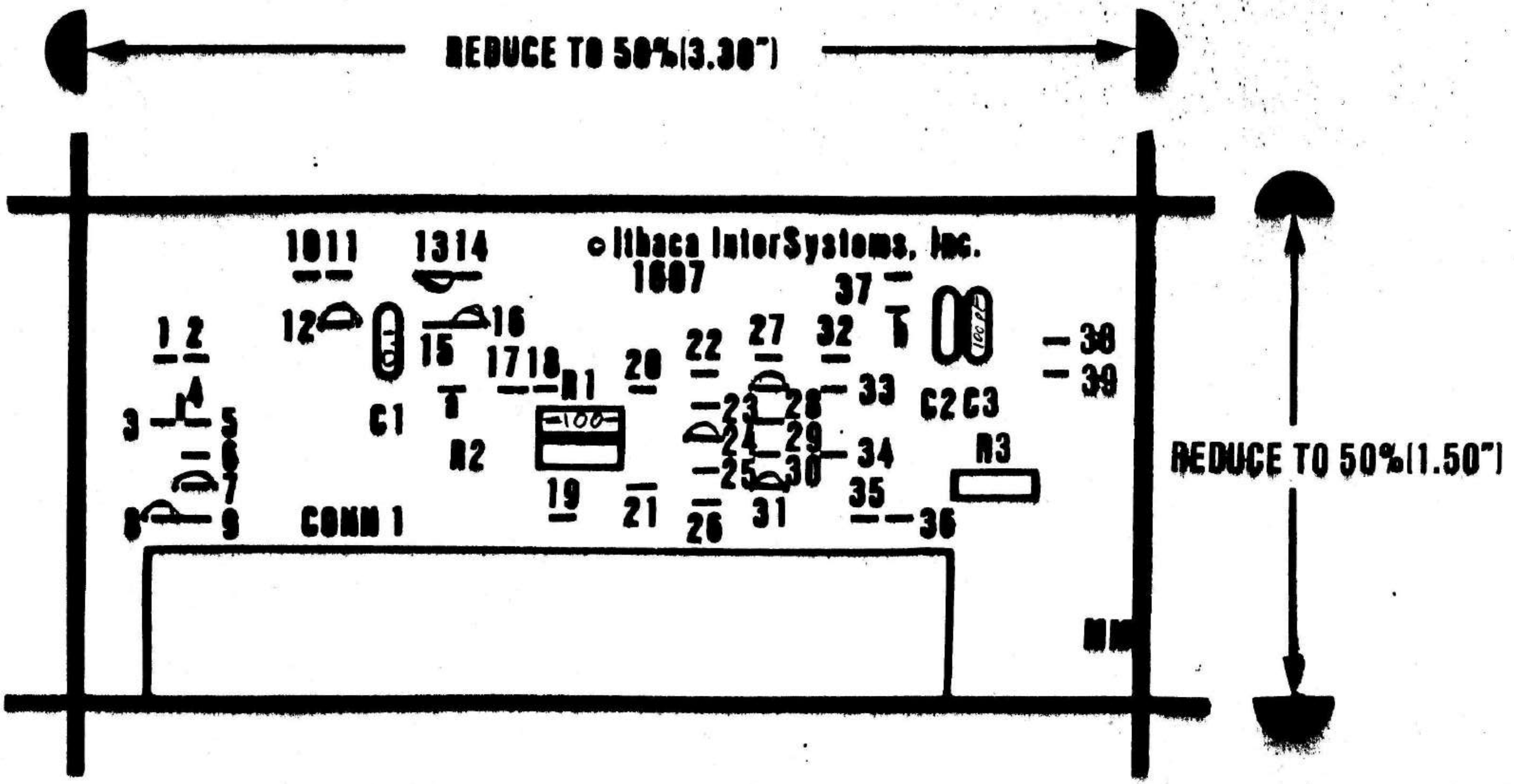
6/16/82
MN



ITHACA INTERSYSTEMS EPPM T2 SILKSCREEN

INTEL 2732
2732

6/16/82
MN



ITHACA INTERSYSTEMS EPPM T2 SILKSCREEN

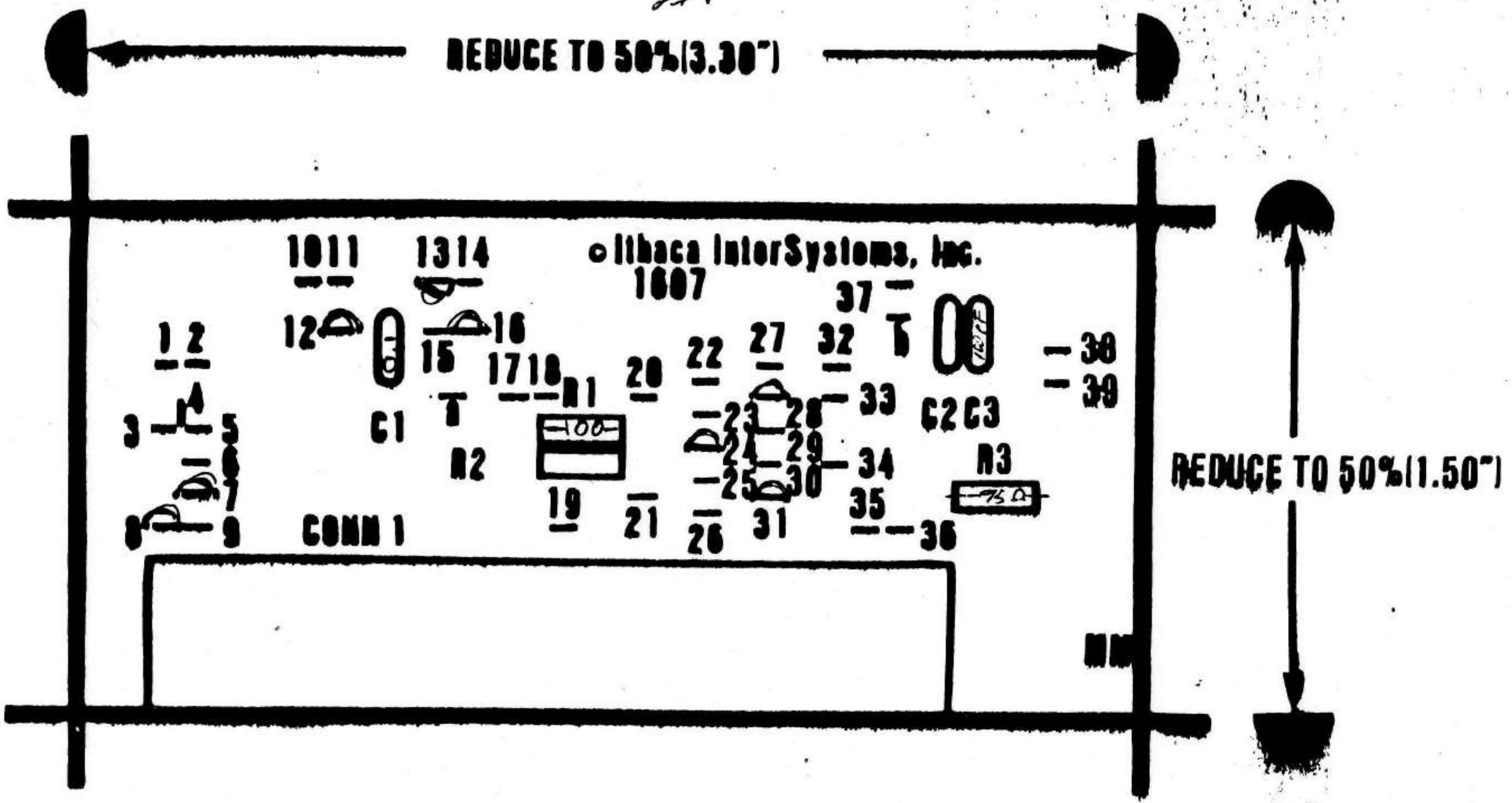
IN[illegible]

2732A

21V

6/16/82

MN



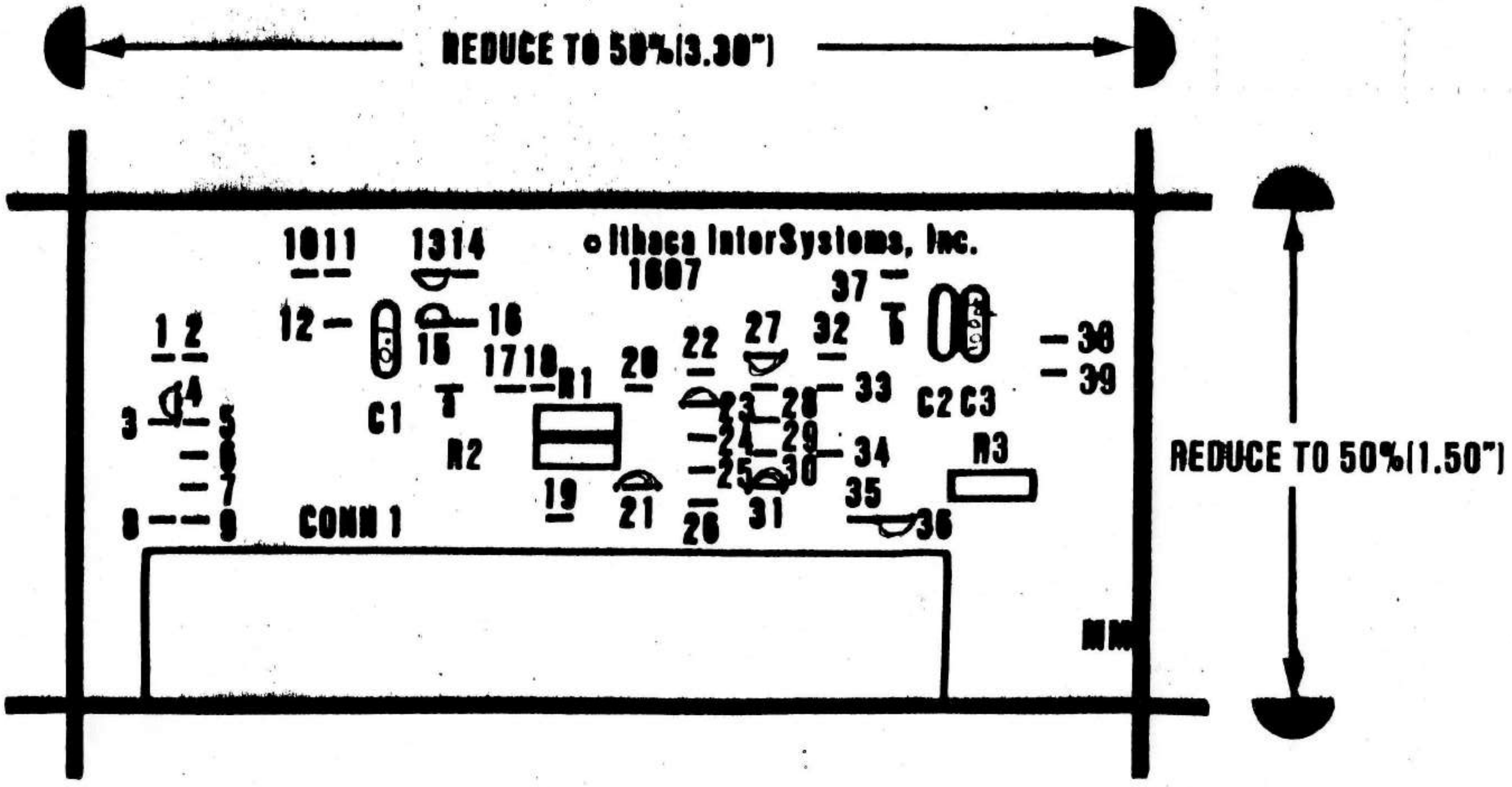
REDUCE TO 50%(3.30")

REDUCE TO 50%(1.50")

ITHACA INTERSYSTEMS EPPM T2 SILKSCREEN

INTEL 2716
2716

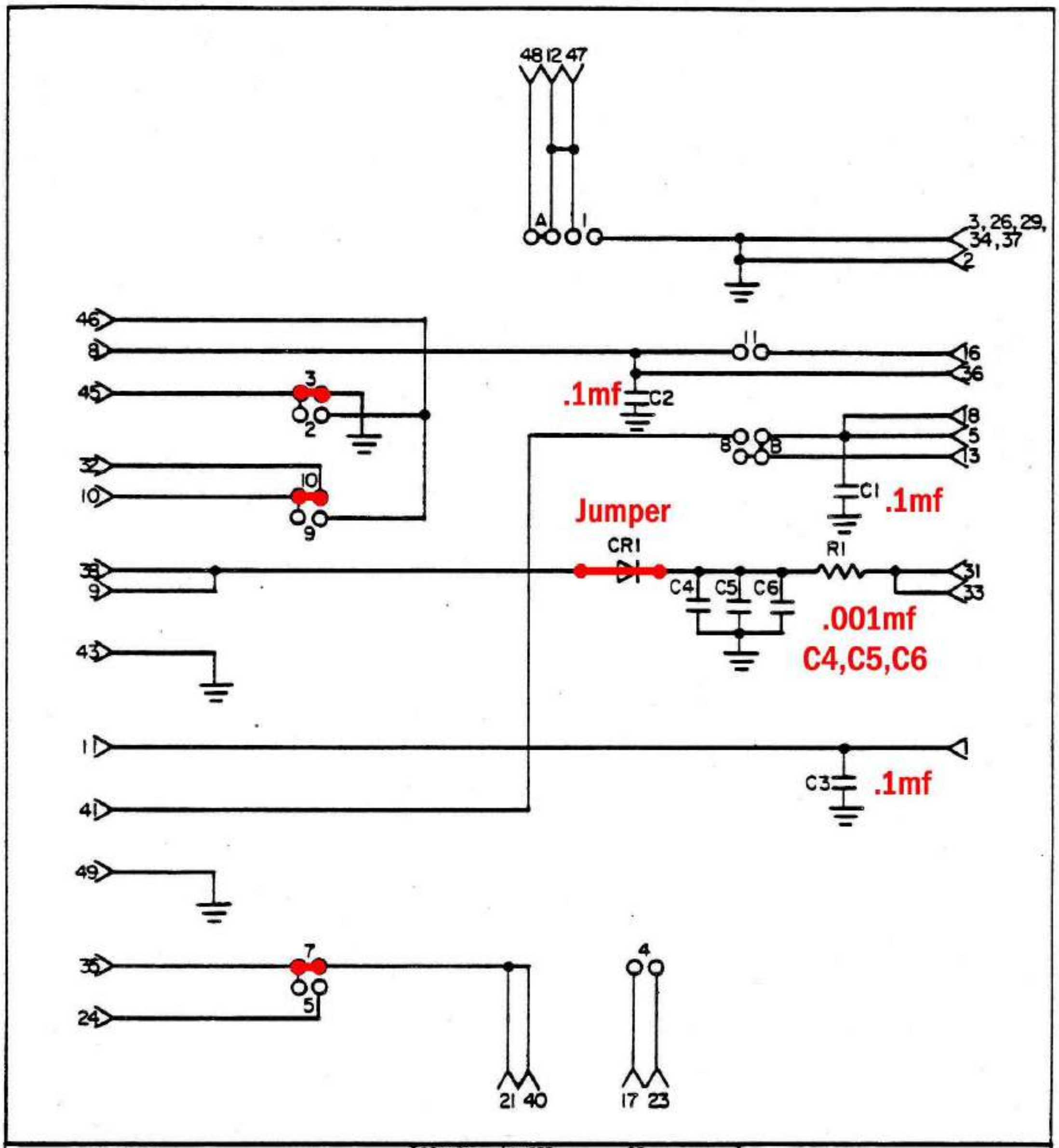
6/16/82
MN


REDUCE TO 50%(3.30")

REDUCE TO 50%(1.50")

ITHACA INTERSYSTEMS EPPM T2 SILKSCREEN

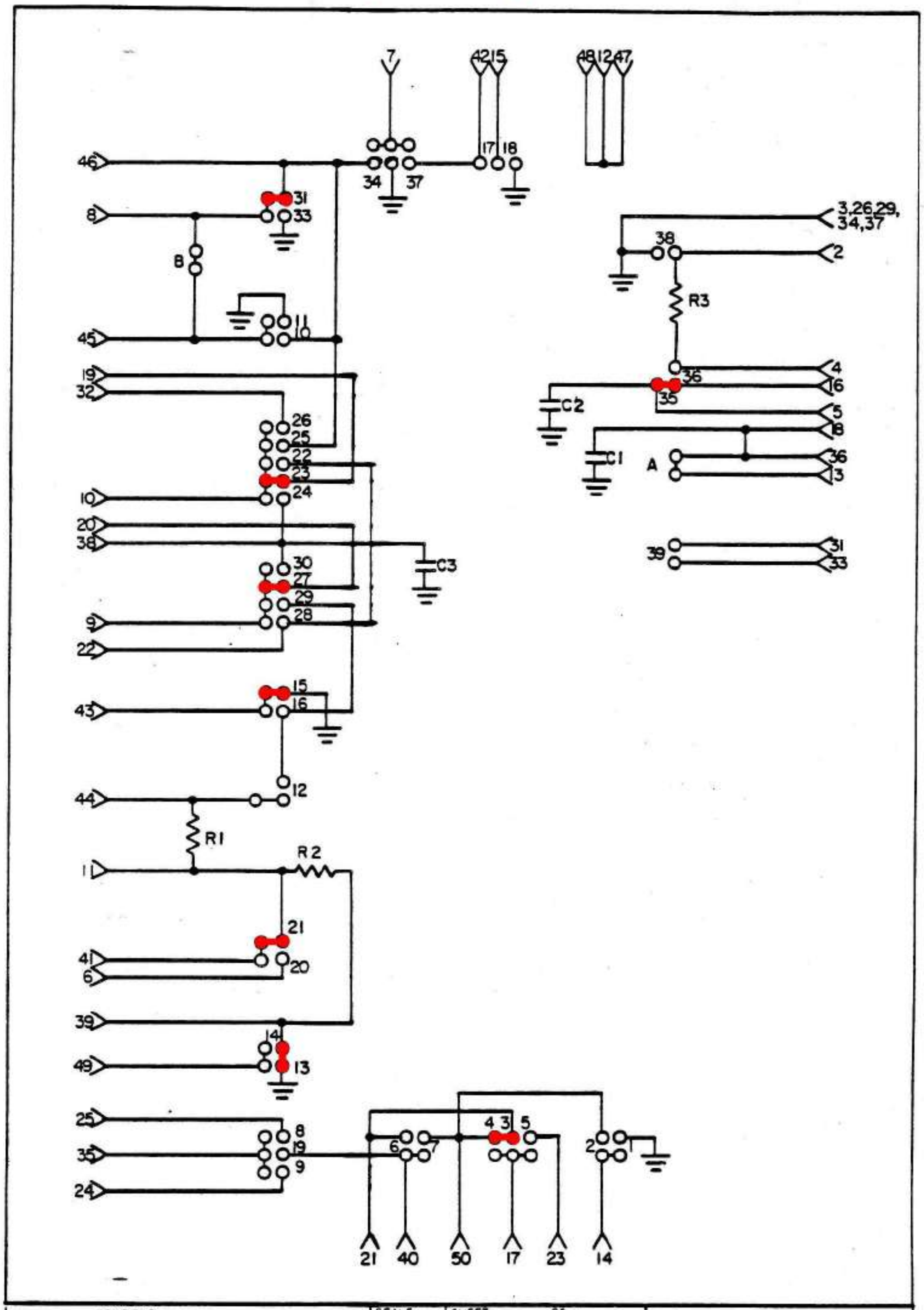
2708



REVISIONS				SCALE	SHEET 1 OF 1	InterSystems
LTR	DESCRIPTION	DATE	APPROVED	DRAWN	DATE 1/18/82	ITHACA, NEW YORK 14850
				CHECKED	DATE	
				UNLESS OTHERWISE NOTED		PROJECT EPROM BURNER PERSONALITY MODULE
				FRACTIONS ±	.00 ± .000 ± ANGLES ±	DATE 1-18-82 NUMBER 1606
				RADII	DRAFT FINISH	

2716

(not TMS2716)



REVISIONS			
NO	DESCRIPTION	DATE	APPROVED

SCALE — SHEET OF

DRAWN ux DATE 1/18/82

CHECKED DATE

UNLESS OTHERWISE NOTED

FRACTIONS ± — .00 ± — .000 ± — ANGLES ± —

RADII — DRAFT — FINISH —

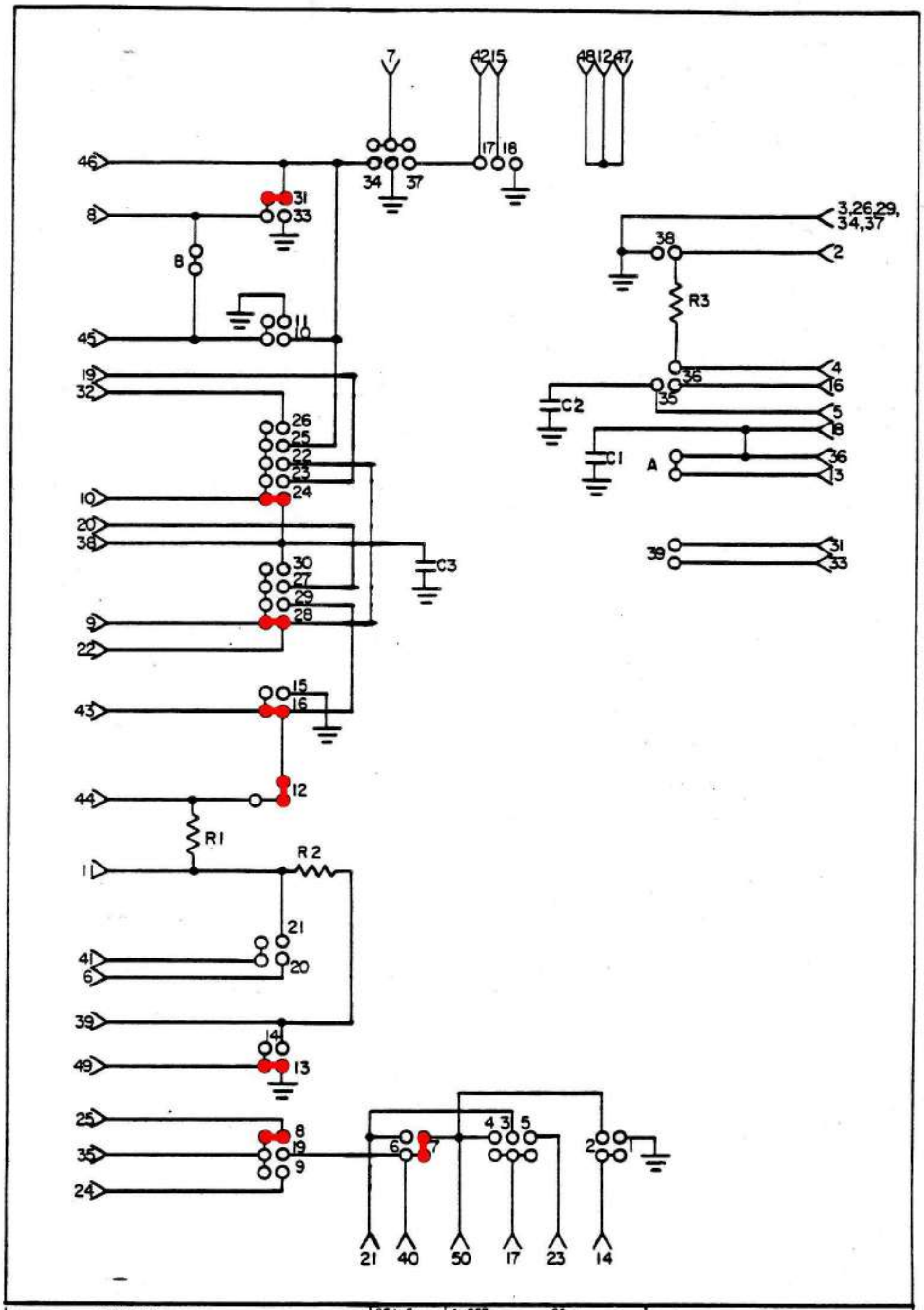
InterSystems

ITHACA, NEW YORK 14850

PROJECT EPROM BURNER PERSONALITY MODULE

DATE 1-18-82 NUMBER 1607

2732



REVISIONS			
	DESCRIPTION	DATE	APPROVED

SCALE — SHEET , OF)

DRAWN uk DATE 1/18/86

CHECKED DATE

UNLESS OTHERWISE NOTED

FRACTIONS ± — .00 ± — .000 ± — ANGLES ± —

RADII — DRAFT — FINISH —

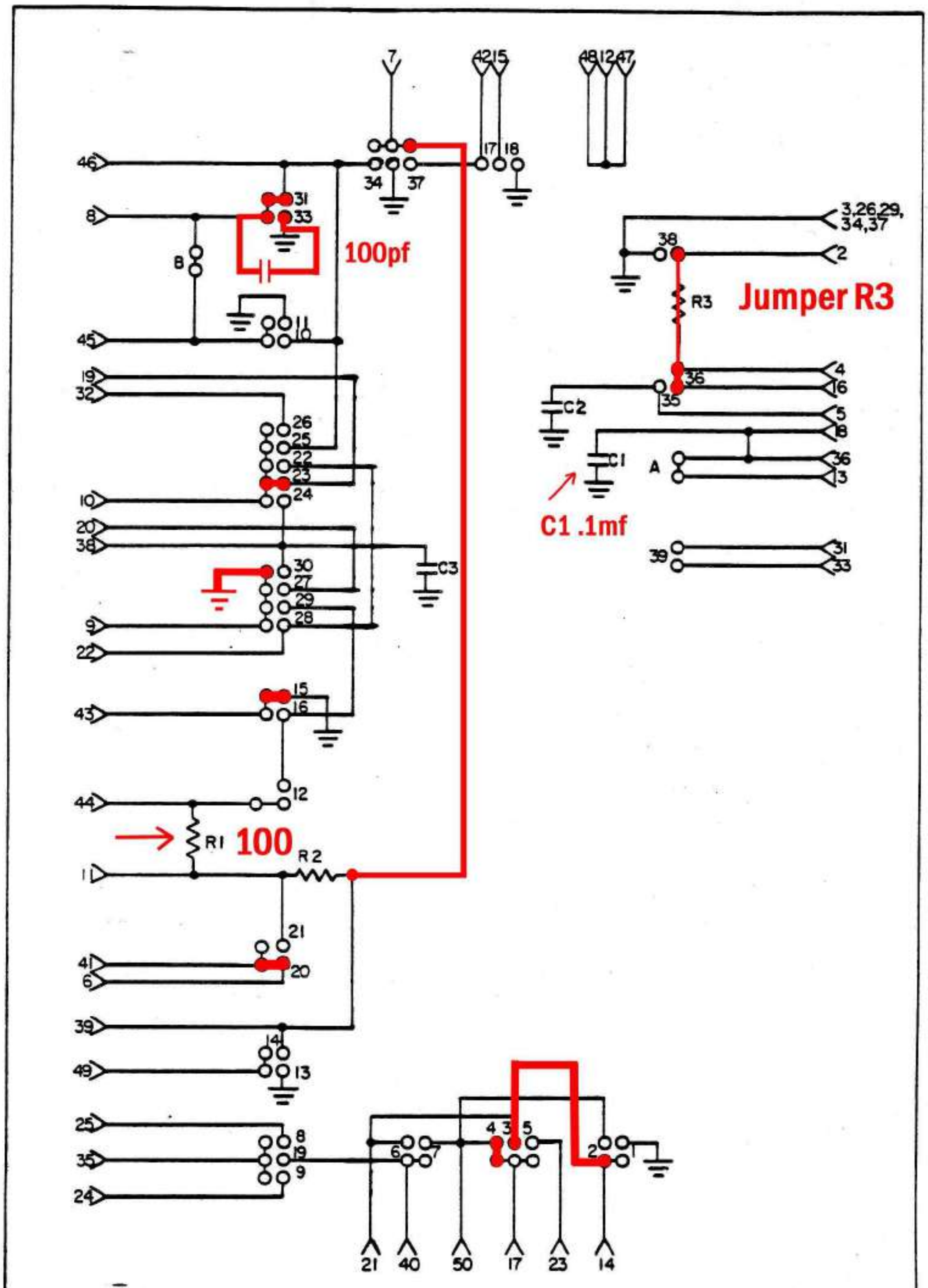
InterSystems

ITHACA, NEW YORK 14850

PROJECT EPROM BURNER PERSONALITY MODULE

DATE 1-18-82 NUMBER 1607

2764



100pf

Jumper R3

C1 .1mf

100

REVISIONS			
LTR	DESCRIPTION	DATE	APPROVED

SCALE — SHEET , OF ,

DRAWN ux DATE 1/18/82

CHECKED DATE

UNLESS OTHERWISE NOTED

FRACTIONS ± — .00 ± — .000 ± — ANGLES ± —

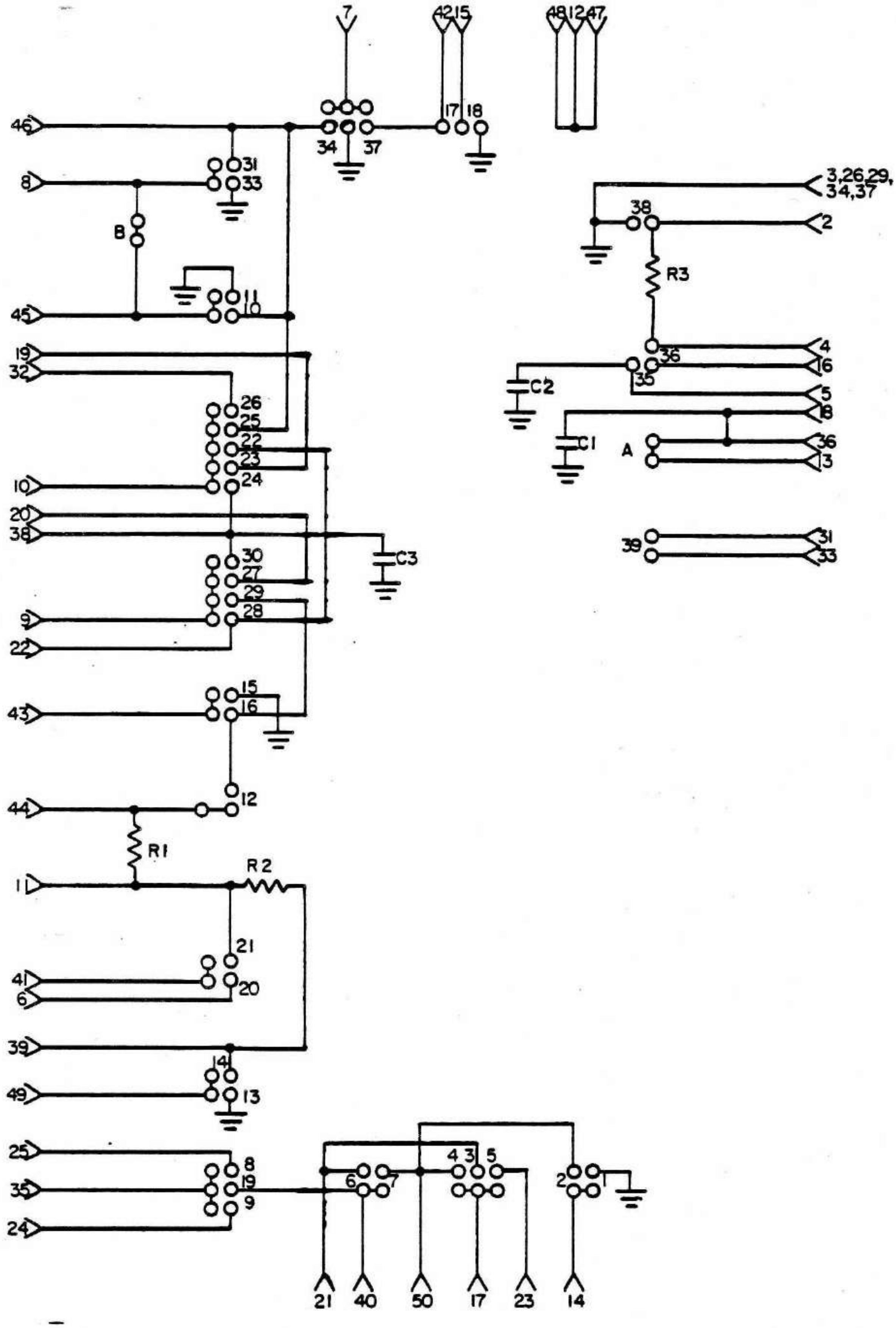
RADII — CRAFT — FINISH —

InterSystems

ITHACA, NEW YORK 14850

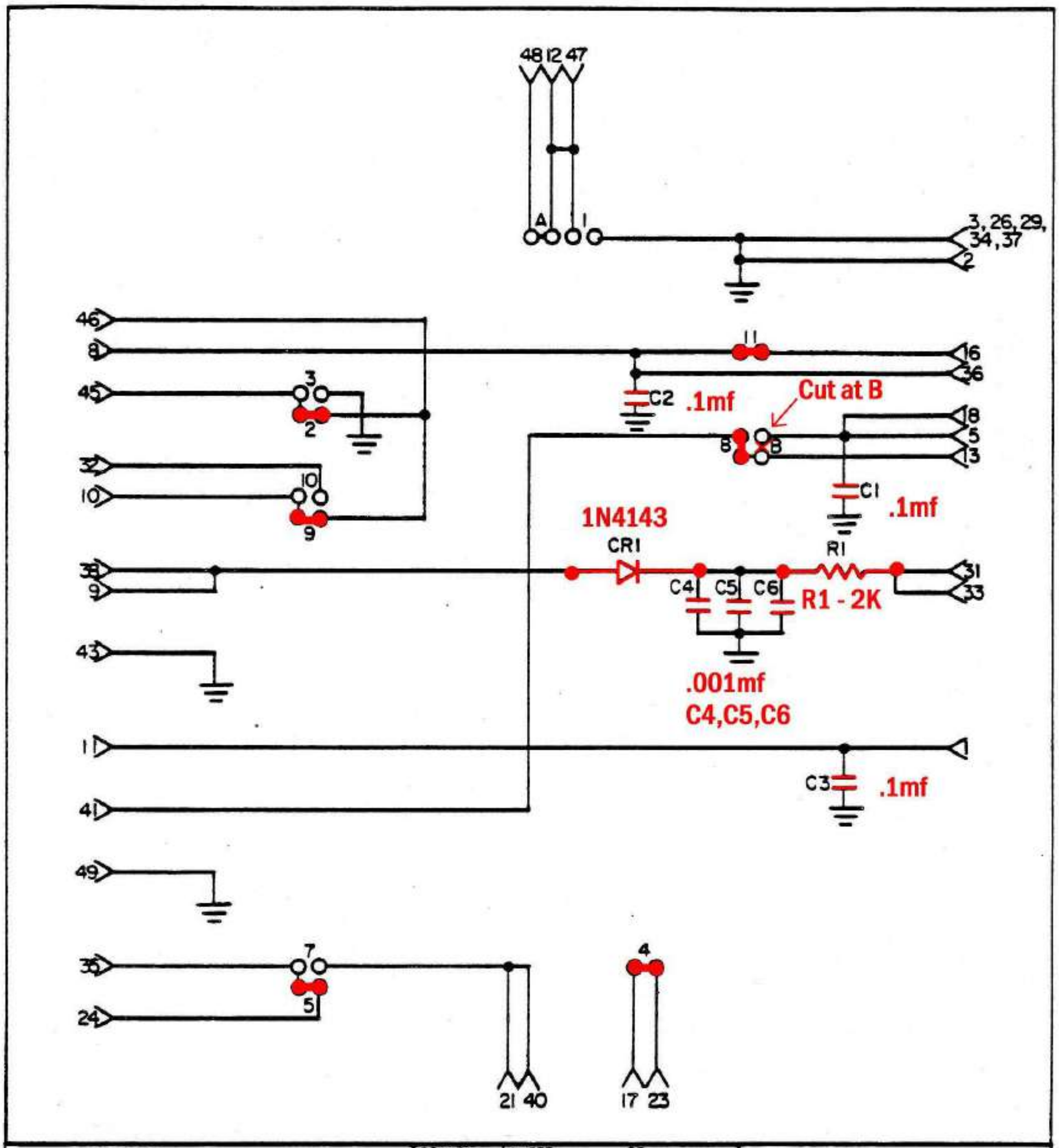
PROJECT EPROM BURNER PERSONALITY MODULE

DATE 1-18-82 NUMBER 1607



REVISIONS				SCALE	SHEET	OF	InterSystems
	DESCRIPTION	DATE	APPROVED	DRAWN dt		DATE 1/18/82	ITHACA, NEW YORK 14850
				CHECKED		DATE	
				UNLESS OTHERWISE NOTED			PROJECT EPROM BURNER PERSONALITY MODULE
				FRACTIONS ±	.00 ± .000 ±	ANGLES ±	DATE 1-18-82
				RADII	DRAFT	FINISH	NUMBER 1607

TMS2716



REVISIONS

DESCRIPTION	DATE	APPROVED

SCALE — SHEET / OF /

DRAWN DATE 1/18/82

CHECKED DATE

UNLESS OTHERWISE NOTED

FRACTIONS ± — .00 ± — .000 ± — ANGLES ± —

RADII — DRAFT — FINISH —

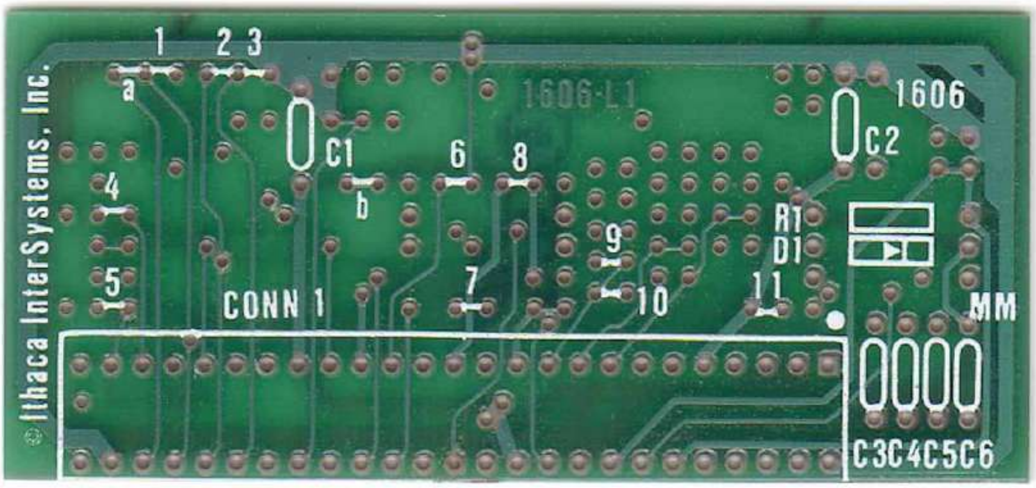
InterSystems

ITHACA, NEW YORK 14850

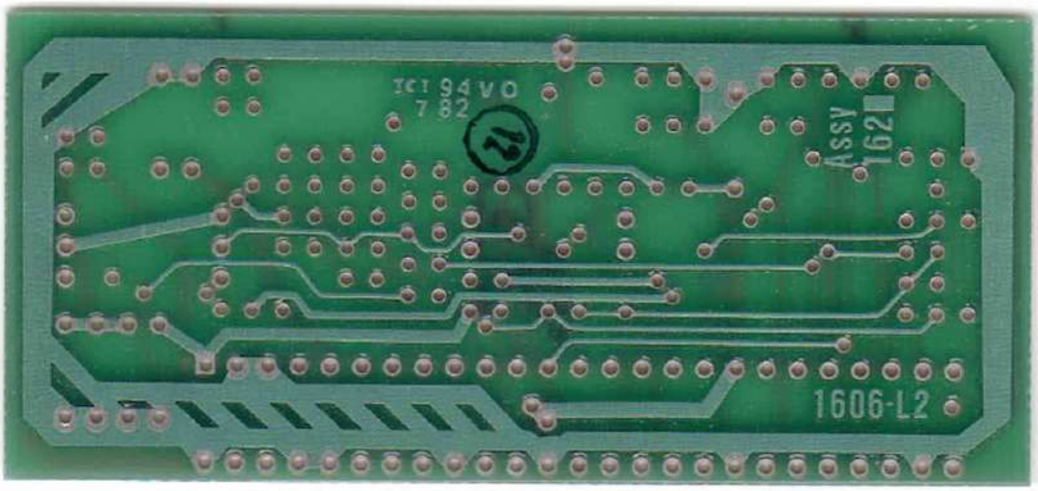
PROJECT EPROM BURNER
PERSONALITY MODULE

DATE 1-18-82

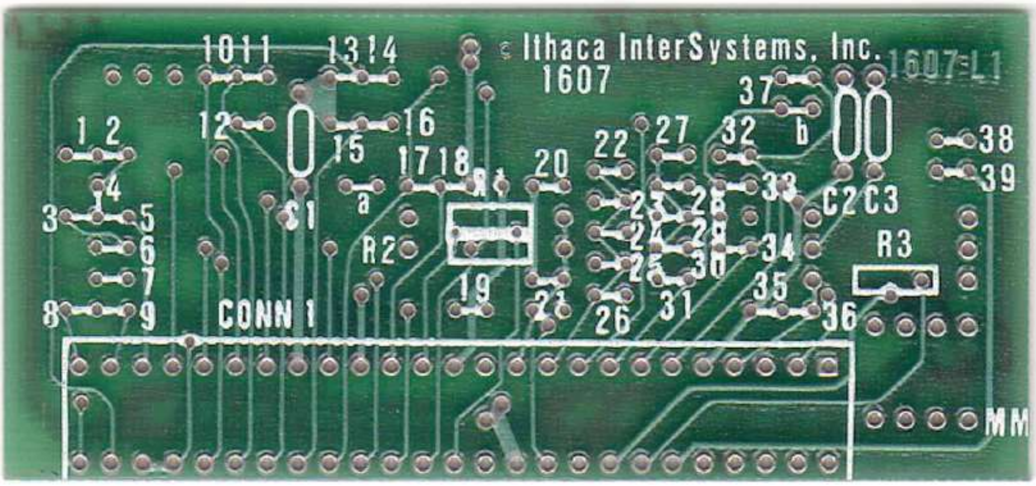
NUMBER 1606



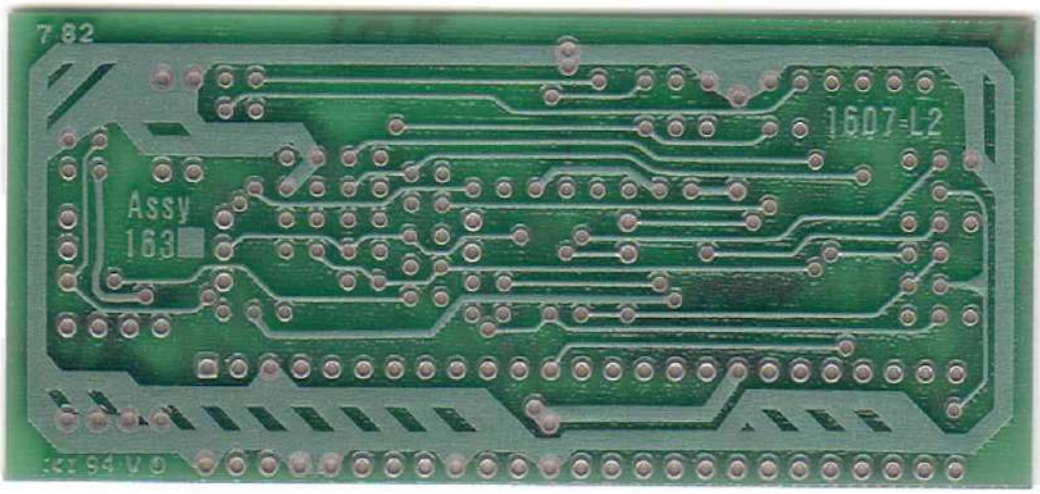
1606 front



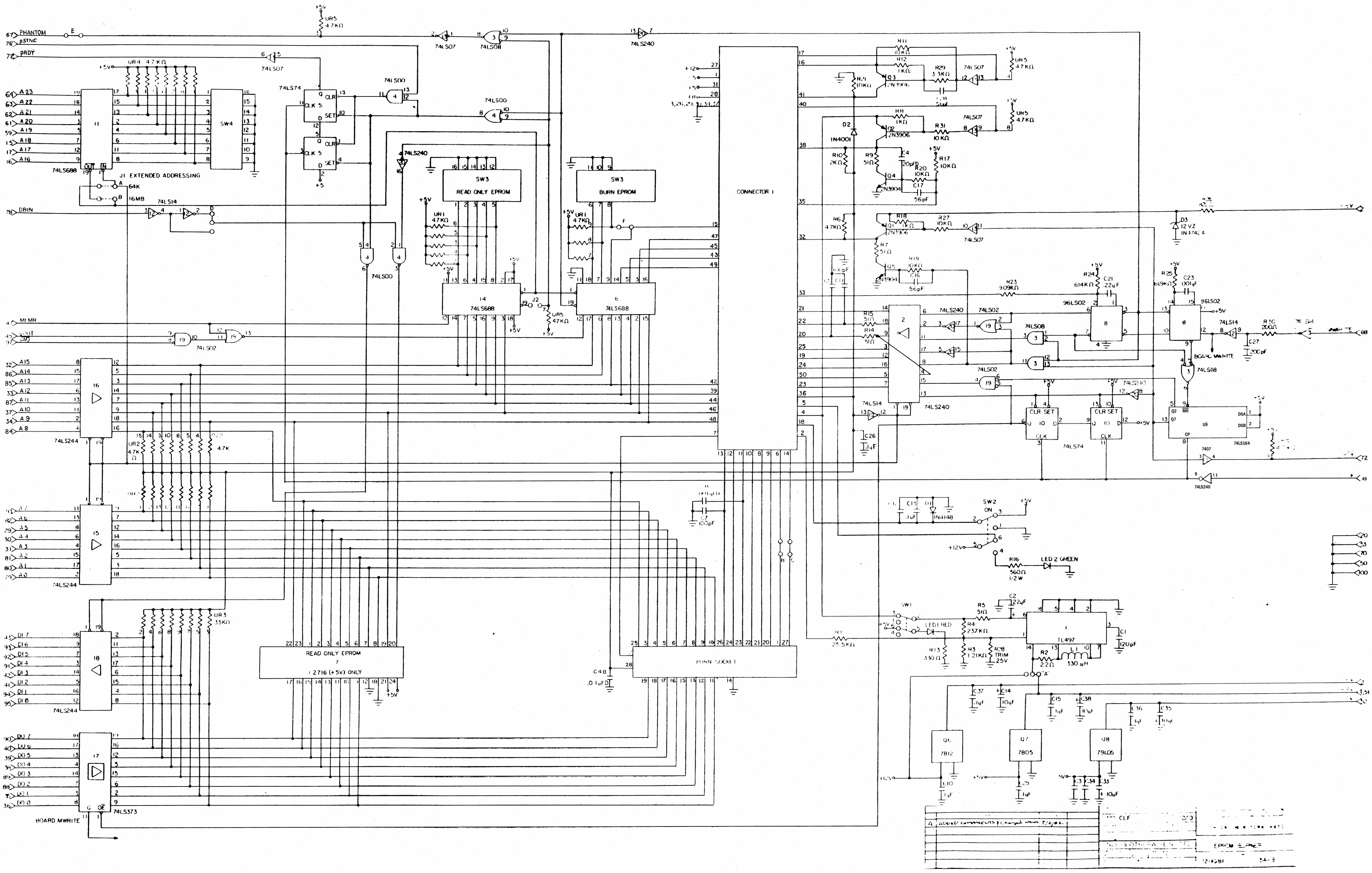
1606 back



1607 front



1607 back


PHANTOM
PSYNC
PRDY
A23
A22
A21
A20
A19
A18
A17
A16
74LS688
J1 EXTENDED ADDRESSING
64K
16MB
SW4
DBIN
74LS14
74LS07
74LS08
74LS74
74LS00
74LS240
SW3
READ ONLY EPROM
BURN EPROM
74LS688
CONNECTOR 1
MEMR
SOUT
SINP
74LS02
A15
A14
A13
A12
A11
A10
A9
A8
74LS244
A7
A6
A5
A4
A3
A2
A1
A0
DI 7
DI 6
DI 5
DI 4
DI 3
DI 2
DI 1
DI 0
DO 7
DO 6
DO 5
DO 4
DO 3
DO 2
DO 1
DO 0
74LS373
BOARD MWRITE
READ ONLY EPROM
2716 (+5V) ONLY
BURN SOCKET
2N3906
2N3904
IN4001
IN4148
12VZ
96LS02
74LS164
7407
BOARD MWRITE
SW2
ON
LED 2 GREEN
SW1
LED1 RED
1L497
7812
7805
79L05
A ADDED COMPONENTS & changed values 7/2/82
CLF
EPROM BURNER
12/10/81