# PROM BLASTER

Apparat, Inc.

# The A.P.B.

## APPARAT PROM BLASTING SYSTEM

### INTRODUCTION

The A.P.B. system is the most versatile and cost effective EPROM programmer on the market. The system will program virtually all of the common 24 pin EPROMs. Each EPROM type is selected by a special personality module that adapts the programmer to the EPROM. The following EPROMs can be programmed:

2704<br>
2716<br>
2732<br>
2516<br>
2758<br>
2708<br>
2716 (5 volt)<br>
2508<br>
2532

The EPROM programmer includes powerful and easy to use software and a complete set of personality modules. With the unique combination of personality modules and sophisticated software, the EPROM programming system will perform many operations impossible with other systems. A list of operations that can be performed follows:

Verify EPROM is erased
Read EPROMs
Copy EPROMs
Copy between different EPROM types
Program EPROMS
Verify that programming is correct
Read or save EPROM data on disk
Program directly from the assembler
Program directly from computer memory
Examine and/or modify memory
Preset memory

The EPROM programmer package consists of an interface card, and a complete set of personality modules, software on disk and instruction manual. The software will run under New DOS or the TRS-DOS operating system for Models I and III.

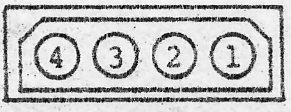
The TRS-80 PROM Burner is designed to be used with Apparat's Bus Extender or Mini Bus Extender. However, if you do not have a Bus Extender, you may connect the EPROM programmer to your computer with a cable and a user-supplied power supply.

If you have a Bus Extender, cut off the power supply cable on APB cord and follow the instructions supplied with the Bus Extender on how to install your EPROM Burner. If you do not have a Bus Extender, you will need a cable to connect the EPROM Burner to your computer. The Model I cable should be a 40 pin ribbon cable with a 40 pin edge connector on one end, and a 50 pin edge connector on the card end. Pins 41 through 50 should not be connected on the 50 pin connector.

For the Model III, the cable should be a 50 pin ribbon cable with a 50 pin edge connector on each end. Pins 42, 44, 46, 48 and 50 should be broken on one of the edge connectors.

Connect the 50 pin edge connector to EPROM Programmer card edge connector. (Note there are two edge connectors on the EPROM Programmer card labeled Model I and Model III. Make sure that you connect the cable to the connector labeled for the computer you are using.) Connect the other end of the cable to the computer: the Screen Printer Port for the Model I, and the I/O Port for the Model III.

Power Supply: If you are not using the TRS-80 PROM Burner with a Bus Extender, you will need to provide a power supply to operate the EPROM Burner. The EPROM Burner comes with a connector to connect the board to a disk drive power supply. If you do not want to use a drive power supply, the EPROM Burner requires +12 and +5 volt supply. These supplies may be connected to J1. Figure 1 shows the pin out of J1.



| Pin | Supply Voltage |
|---|---|
| 1 | +12v dc |
| 2 | Return ( +12v dc) |
| 3 | Return ( +5v dc) |
| 4 | +5v dc |

Figure 1

## USING THE A.P.B.

Before you proceed with the operation of the A.P.B., it is advisable to make a copy of the disk and set it aside. You may use a DOS disk to make a backup. It is also advisable to make some practice runs through the software to familiarize yourself with its operation.

## GENERAL

The TRS-80 PROM Blaster package consists of three items: the Programmer Card, a set of Personality Modules, and disk-based software. The personality modules are 16 pin plug-in adapters that reconfigure the programming socket to match the type of EPROM being programmed. The Personality Modules plug into the Programming Card and adapt it to the characteristics of each EPROM type. EPROMs are programmed by inserting the EPROM into the card, selecting the proper personality module and running the software.

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

!!! WARNING !!!

EPROMs can be damaged by static electricity.

NEVER install or remove EPROM with the power on.

NEVER remove or install TRS-80 PROM Blaster card with the power on.

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

## SOFTWARE PACKAGE

All of the programming operations of the TRS-80 EPROM Programmer are performed by a powerful and flexible software package. The software is included on a 5-inch disk as a part of the A.P.B. package. The software will run under the TRS-DOS operation systems for both the Model I and Model III. Instructions for using the software are provided later in this manual.

## WORKING MEMORY

Many of the operations of the TRS-80 EPROM Programmer package are performed between the TRS-80 EPROM Programmer card and an area of memory starting at 7000h. The working memory can be read from disk, written to disk, read from memory, read from EPROM, written to EPROM and edited.

Programming an EPROM consists of five steps:

1) Erase EPROM.

2) Set up programmer.

3) Load data to be blasted.

4) Verify that the EPROM is erased. Do actual blast and verify correct blast.

5) Remove EPROM.

## ERASE EPROM

All of the EPROMs programmed by the TRS-80 PROM Blaster System are erased using ultraviolet light. Each EPROM has a small glass window. Shining ultraviolet light through this window erases the EPROM. In the erased state, every byte in the EPROM is set to hex FF.

Ultraviolet EPROM erasers are available from many computer stores. Erasing an EPROM takes from a half hour to an hour, depending upon the intensity of the eraser. To ensure complete erasure, follow the EPROM manufacturer's instructions.

## SETTING UP TRS-80 PROM BLASTER TO PROGRAM

With the control lever in the open position, install the EPROM to be programmed in the zero insertion force socket. Be sure that Pin 1 of the EPROM is aligned with the index mark on the socket. Lock the EPROM in the socket by moving the control lever to the closed position.

Select the proper personality module from the chart below and install it in Socket U1. Be sure that the index mark lines up with Pin 1.

### PERSONALITY MODULE TABLE

| EPROM TYPE | MODULE TO USE | DRIVER FILE SPEC |
|---|---|---|
| 2704,2708 | 2708 | I2708 |
| TI 2716 (3 voltage) | 2716 (3 voltage) | TI2716 |
| 2508 | I2716 (5 volt) | I2716 |
| 2516 | I2716 (5 volt) | I2716 |
| Intel 2716 (5 volt) | I2716 (5 volt) | I2116 |
| 2532 | 2532 | TI2532 |
| 2732 | 2732 | I2732 |

Use the appropriate software driver routine to run the APB software. First the user enters the system DOS, then types the program.

EPROM, then ENTER.

(The software driver file spec may be entered after the program name, and the driver will be automatically loaded.) 'APPARAT PROM BURNER' should appear on the screen. Press 'Enter' to continue. 'INPUT FILE SPEC' should now appear on the screen. Now enter EPROM type's file spec from the preceding table.

(File Spec: Drive Number)

If no driver is needed, press 'Enter' to continue.

At this point the main menu should be displayed.

```
Main Menu     Main Menu

A:   LOAD ASCII FILE
B:   CHECK TO SEE IF PROM BLANK
D:   SAVE ASCII FILE
E:   EDIT MEMORY
I:   LOAD PROM DRIVER
L:   LOAD BINARY FILE
R:   READ PROM
S:   BINARY SAVE
W:   WRITE PROM
Z:   CALL USER ROUTINE
Q:   QUIT AND JUMP TO DOS
```

The user should type the letter function he wishes to use.

## FUNCTION DESCRIPTION

### EPROM I/O

ENTER as the first character of any input will return the user to the main menu.

"Byte Count" (BC): This is merely the number of bytes which will be read, written or verified.

"Relative PROM Address" (RPA): This is the byte in which the read/write/verify will begin. It can also be thought of as an offset from the beginning of an EPROM. RPA number of bytes will be skipped before the operation begins.

"Computer Memory Address Start" (CPA): This is only used in 'write'. It tells the computer the data source.

The first data byte is CPA; the next is CPA+1; the last is cPA+BC. This number is most often 7000h since that is where all other I/O is done.

NOTE: CPA+BC should not exceed the capacity of the EPROM. If it does, any remaining bytes to be read, written or verified will overlap to the beginning of the EPROM.

## I LOAD AN EPROM DRIVER

Before any EPROM (read, write or verify) operation can be used, an EPROM driver must be loaded. This EPROM driver can be either one of the Apparat supplied drivers or custom driver written by the user. If the driver is written by the user, it must follow the guidelines set down in Appendix I and include a read, a write, and a verify blank routine.

When you hit "I" from the main menu, the computer asks you to 'ENTER FILE SPEC'. The file spec you enter must be a legal file as defined by TRSDOS, include any file type extension (e.g., /obj,/cmd,/drv) and should also include a drive number to ensure that the driver is taken from the correct drive.

NOTE: The more flexibility any program gives the user, the more responsibility the user must take in using that flexibility, and the less "goof-proof" that program can be.

The only check that is made on the file chosen to be loaded is that it is binary. The user must make sure that it is indeed a driver and that it does not overlap the EPROM program. (See Appendix I)

WARNING: A driver can reside anywhere it memory. Always know exactly where it is. This avoids unintentionally destroying it through a conjugation, read, or load!

## R READ AN EPROM INTO MEMORY

The user must input the Byte Count (BC) and the Relative Prom Address (RPA). When using a driver that was supplied, or any driver that complies with Appendix I, the EPROM will be read into memory location 7000h through 7000h+BC. A blank EPROM will return all FFh; a blown EPROM might also return FFh, or 00.h. No card in system will return FF.

## B TEST TO SEE IF EPROM IS BLANK

The user must input the BC and the RPA. The routine checks to see that all bytes in the EPROM are OFFh (every bit is 1), starting at RPA, in the EPROM, to RPA+BC. If they are all FFh, then the routine tells you that the EPROM is blank; otherwise it reports the EPROM is not blank.

## W WRITE AN EPROM FROM MEMORY

The User must input the BC, the RPA, and the CPA. The driver then checks the EPROM to see if it is blank. If it is, the write continues. If the EPROM is not blank, the user is given the choice to continue or not.

The data at computer memory CPA through CPA+BC is transferred (written) to the EPROM starting at RPA and continuing through RPA+BC. The drivers supplied will verify that the EPROM was written accurately; if there is an error, it will be reported to the User and the write will terminate.

If the error happens at once, check the power, the personality module, and that the EPROM is inserted correctly. If all of these things check out, or if the error is not at once, suspect the EPROM itself.

## DISK I/O

"Input File Spec" (FS): For loads, input an existing file name (as defined by TRSDOS). For saves, it can (but need not) be an existing file name.

"Start Address" (SA): For saves, this is the computer memory address at which point the data to be saved originates. Since all EPROM I/O uses 7000h for its starting address, that is most often the response given to this inquiry.

"Ending Address" (EA): This is the last address to be saved (inclusive).

## L LOAD A BINARY FILE

The user must input the FS. The binary file FS is loaded into memory at 7000h. Although this routine loads all programs at 7000h, it does not change the program in any way--the purpose is only to allow the user to write to the EPROM a program that conflicts with ROM, DOS, or the EPROM burner software. (E.g., a program designed to run at 0000h can now be loaded and saved.) A period is printed after every block is loaded.

## S SAVE A BINARY FILE

The user must input the FS, SA and EA. Computer memory from SA through EA inclusive is saved to disk under file spec FS. This will be accomplished by jumping to DOS and executing the DUMP command. Any error that occurs will be reported on the screen in the normal DOS manner only. Under NEW DOS/80, hitting any key will return the user to the EPROM Burner software. Under TRSDOS the user will find himself back in DOS. If you wish to return to the EPROM Burner software, you must reload it.

## A LOAD ASCII FILE

The user must input the FS. The file FS is loaded into memory starting at 7000h. The file must be ASCII since loader information, if it exists, will be loaded as data. A period (.) will be printed for every line feed, carriage return, or control Z.

## D SAVE ASCII FILE

The user must input the FS, SA and EA. This differs from a binary save in that it is all done by the EPROM Burner software and not by DOS. It also differs in the fact that there is no loader code information, just continuous data.

## E EDIT MEMORY

At the main menu level, hitting the "E" key will put you in full screen edit mode. The top left position will be 7000h, the bottom right will be 70FFh. Any hex character the user now types will replace the character under the big cursor. Controls in edit mode are as follows:

Arrows: Arrows move the cursor in their respective directions.

Clear Key: When you hit the Clear Key, you will switch modes between hex and ASCII. The first time this is done, the big cursor on top of the hex characters will switch with the little cursor on the ASCII characters. Any character the user now types, except one of the EPROM Burner's control characters, will be put directly into memory. To get a control character into memory, you must switch back to hex and type in the appropriate hex value.

#: Conjugate memory. The user is asked the beginning and ending addresses to conjugate and also the conjugate byte. The conjugate byte is then written throughout the memory range inclusively.

/: The computer asks the user where in memory to jump. If the user just hits an ENTER, the computer stays in the same place.

!: Leave edit mode and jump back into main memory.

## DO ACTUAL BLAST

After you have loaded the data, return it to the main menu. Select the 'Write' EPROM option. The amount of time required to blast an EPROM depends upon the EPROM type. The 3 voltage EPROMs (2704, 2708, and TI2716) require much more time than the other types. These EPROMs require about one hundred passes to program.

The single voltage EPROMs only require a single pass, so they program much faster.

If there are errors, check the following:

EPROM in socket backwards
Wrong personality module
Personality module in backwards
Try erasing for a longer period of time
Power off

If these fail to correct the problem, the EPROM is probably defective; try another EPROM.

## REMOVE EPROM (Turn Off the Computer!!!)

Remove the EPROM from the card. If you are going to program several EPROMs, you can leave the card in the slot. If this is the only EPROM you are going to program, you may remove the card before removing the EPROM. To prolong the life of the slot edge connector, you should minimize insertions and removals from the slot.

## COPYING EPROMS

Copying an EPROM is similar to the programming process described above. The process involves reading the EPROM to be copied into memory, then, if desired, saving the data on disk. Once it is saved, one or more copies can be made as described above.

Here are the steps:

1) Set up the card and install the EPROM to be copied. This is described in detail in the programming sequence above.

2) Boot the disk, enter EPROM and the EPROM File Spec. Enter the main menu, as described above.

3) Select the 'READ EPROM' option. This will read the EPROM into memory starting at 7000h.

4) Now, if you want to save a binary file or an ASCII file, select the appropriate option. Save the data on disk according to your application. Reading and saving data is described in detail in the software section of this manual.

5) Remove the EPROM as described in the programming instructions above.

6) You now make one or more copies of the data using the normal programming sequence described above.

## HARDWARE NOTES

The A.P.B. board has been designed to blast all of the common 24 pin EPROMs. Most of the pins on these EPROMs are identical. The only differences exist on pins 18, 19, 20, 21 and 24. These pins require various voltages and logic levels depending upon the EPROM type. These pins are connected to the personality module socket on the card. Other pins in the personality module socket provide special logic levels and voltages needed by the different EPROM types. By installing headers with appropriate jumpers, the proper logic levels can be connected to the proper EPROM pins for each EPROM type.

The following special signals are available at the personality module socket:

| | |
|---|---|
| A11 | Address bit 11, provides most significant bit for 4K EPROMS |
| A10 | Address bit 10, provides most significant bit for 2K EPROMS |

| | |
|---|---|
| PPH | High Program Pulse, can be switched between 0, 5, 26 volts. |
| PPL | Low Level Program Pulse, can be switched between 0, 5, and 12 volts. |
| TTL1 | TTL level pulse, used for chip select, output enable, etc. |
| -5V | Minus 5 volt power supply line |
| +12V | Plus 12 volt power supply line |
| +5V | Plus 5 volt power supply line |

## EPROM ADDRESSING

The programming and reading addresses for the EPROM are provided by a 12 bit binary register. This takes the EPROM off the computer address bus and allows the programmer to program large EPROMs without interfering with the EPROM space. The register is loaded with a byte address under software control.

The EPROM Blaster uses 4 port based registers to control EPROM programming. The registers are defined as follows:

| Port Address | Register |
|---|---|
| 3Ch | Control |
| 3Dh | EPROM Address (Low) |
| 3Eh | EPROM Address (High) |
| 3Fh | EPROM Data |

Control Register: The control register is used to set the logic states and to control the voltages to the personality modules.

Control Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Assignment | Not Used | Not Used | TTL 1 | Data Reg Enable | PPL D | PPL C | PPH B | PPH A |

PPH

| B A | Voltage |
|---|---|
| 0 0 | 0 |
| 0 1 | 5 volts |
| 1 0 | 26 volts |
| 1 1 | Undef. |

PPL

| D C | Voltage |
|---|---|
| 0 0 | 0 |
| 0 1 | 0 |
| 1 0 | 5 volts |
| 1 1 | 12 volts |

Address Registers: Ports 3DH and 3EH are used for the EPROM Address Register. Port 3DH is used for the lower 8 bit of the EPROM Address. Port 3EH is used for the high 4 bits of the EPROM Address. Writing to 3DH and 3EH will establish the address of the byte to be programmed by the EPROM Blaster.

The Address Register will hold the address information until written to again. The Address Register cannot be read by the CPU.

Data Register: The Data Register holds the data that is being programmed into the EPROM. The Data Register is enabled to the EPROM by setting bit 4 of the control register. The Data Register can be written to by writing to Port 3FH and can be read only when bit 4 of the control register is set. The Data Register can be read on Ports 3CH and 3FH.

## Procedure for Writing Data to EPROM

First the low order address byte should be written to the Address Register. Then the high order address bits are written to the Address Register. The data to be programmed into the EPROM is written to the Data Register. Then bit 4 of the control register is set and the data will be placed on the EPROM Data bus. Next, the rest of the bits in the control register are set or reset as necessary to program the byte of data.

## Read Data

To read data from an EPROM, first the low and high order address bits are written to their respective registers. Bit 4 of the control register is reset to disable the Data Register. Next set the control register so the EPROM is in a read mode. Then input the data through Port 3CH.

## Charge Pump

Some EPROMs require -5 volt and all EPROMs require a +26 volt programming voltage. However, only +5 and +12 are supplied to the Programmer. The board must generate the other voltages. A charge pump triples the 12 volt supply to provide approximately 36 volts, which is then regulated down to 26 volts. The charge pump also inverts the 12 volt to provide -12, which is then regulated down to -5 volts.

## APPENDIX I

### Writing a Specialized Driver:

For most applications the drivers supplied are adequate. Occasionally, however, the user has a highly specialized application that requires special interaction with the outside world; in this case the APB software allows a custom driver to be loaded and used.

The custom driver has 4 links in which it can communicate with the APB PROM software. These are: Write, Read, Blank, and Extra. With Write, Read, and Blank, parameters are passed back and forth. Extra calls your routine. To get to Extra, just type 'Z' from the Main Menu.

The parameters passed are as follows:

```
       Write:   BC  = byte count
                HL  = relative PROM address
                DE  = start of memory to be written
                Return:
                       z = no errors in verify
                      nz = error in verify

       Read:    BC  = byte count
                HL  = relative PROM address
                Return:

                      BC bytes written into memory starting
                      at 7000h

       Blank:   BC  = byte count
                HL  = relative PROM address
                Return:

                       z = blank
                      nz = not blank
```

To hook your routine into APB, you must place the address of the start of your routines into memory starting at 5201h (see below). To return to APB, simply execute a Return.

```
     100     DEFB     OC3H
     200     DEFW     WRITE  ; Your write routine w/ label WRITE
     300     DEFB     OC3H
     400     DEFW     READ
     500     DEFB     OC3H
     600     DEFW     BLANK
     700     DEFB     OC3H
     800     DEFW     EXTRA  ; Your routine
     900; The reason for the Oc3h (jp) and not an Ocdh (call) is
     999; because the program calls these addresses; e.g., call 5200h
```

To talk to the PROM, the port #s are as follows:

| | |
|---|---|
| 60 | PROM type value/data |
| 61 | PROM relative low value |
| 62 | PROM relative high value |
| 63 | Data to write |

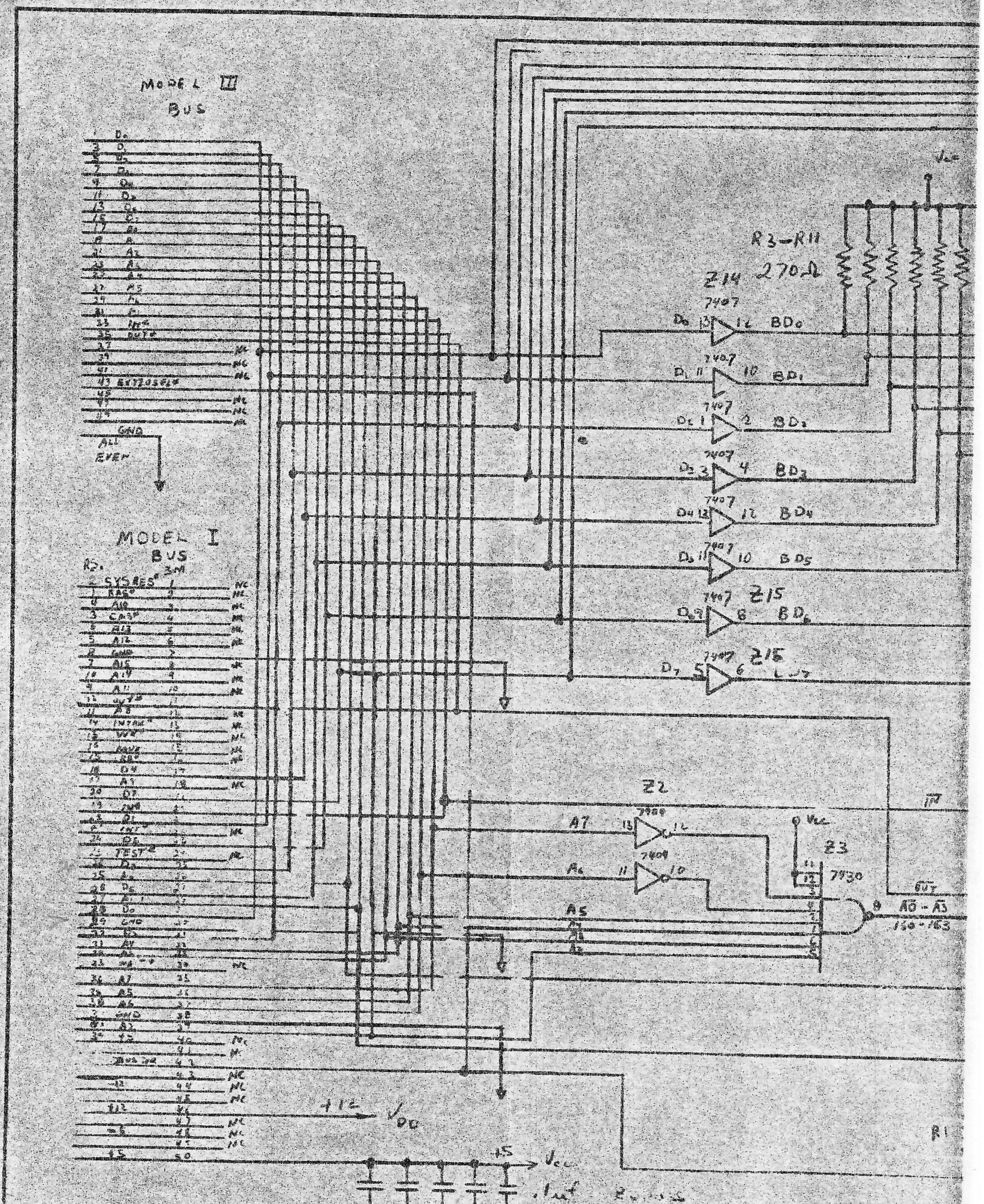
| PROM type values: | X | Y | Z |
|---|---|---|---|
| 2708 | x | y | z |
| 2716 | x | a | f |

```
Before doing anything:      OUT  (C=60),X
Before doing a Write:       OUT  (C=60),Y ; this turns Write on, so
                            OUT  (C=60),Z ; have everything already
                            OUT  (C=60),0 ; done after Write
```

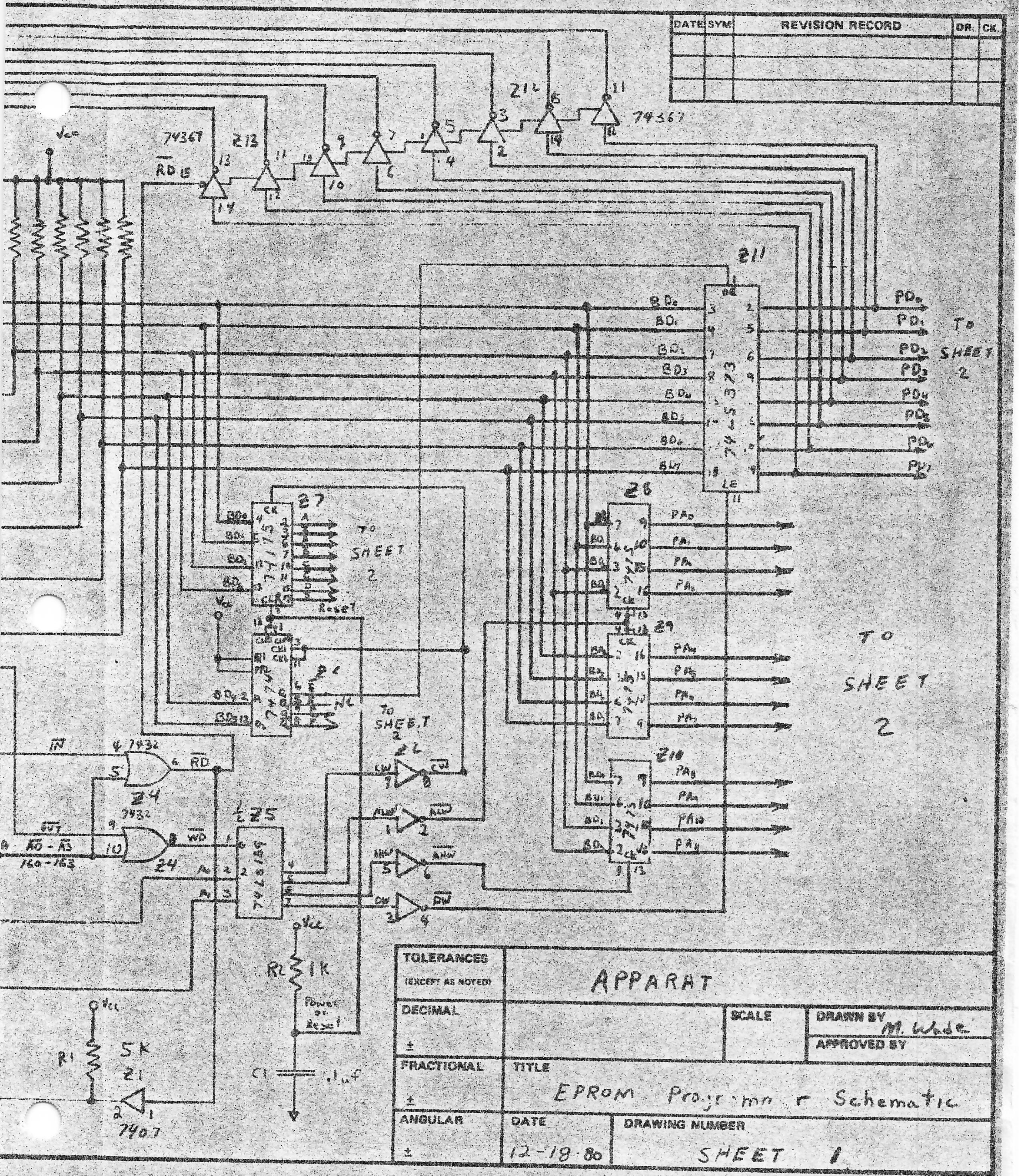
```
A few extras:
CALL 520Fh    ; save byte in A to memory.  First call save to
              ; 7000h, next at 7001h, etc.
(5212h) stores 16 bit counter for 520Fh.
(5214h-5614) is a 400h byte butter
```

Questions, comments and suggestions, send to:

Apparat, Inc.
4401 S. Tamarac Pkwy.
Denver, CO  80111

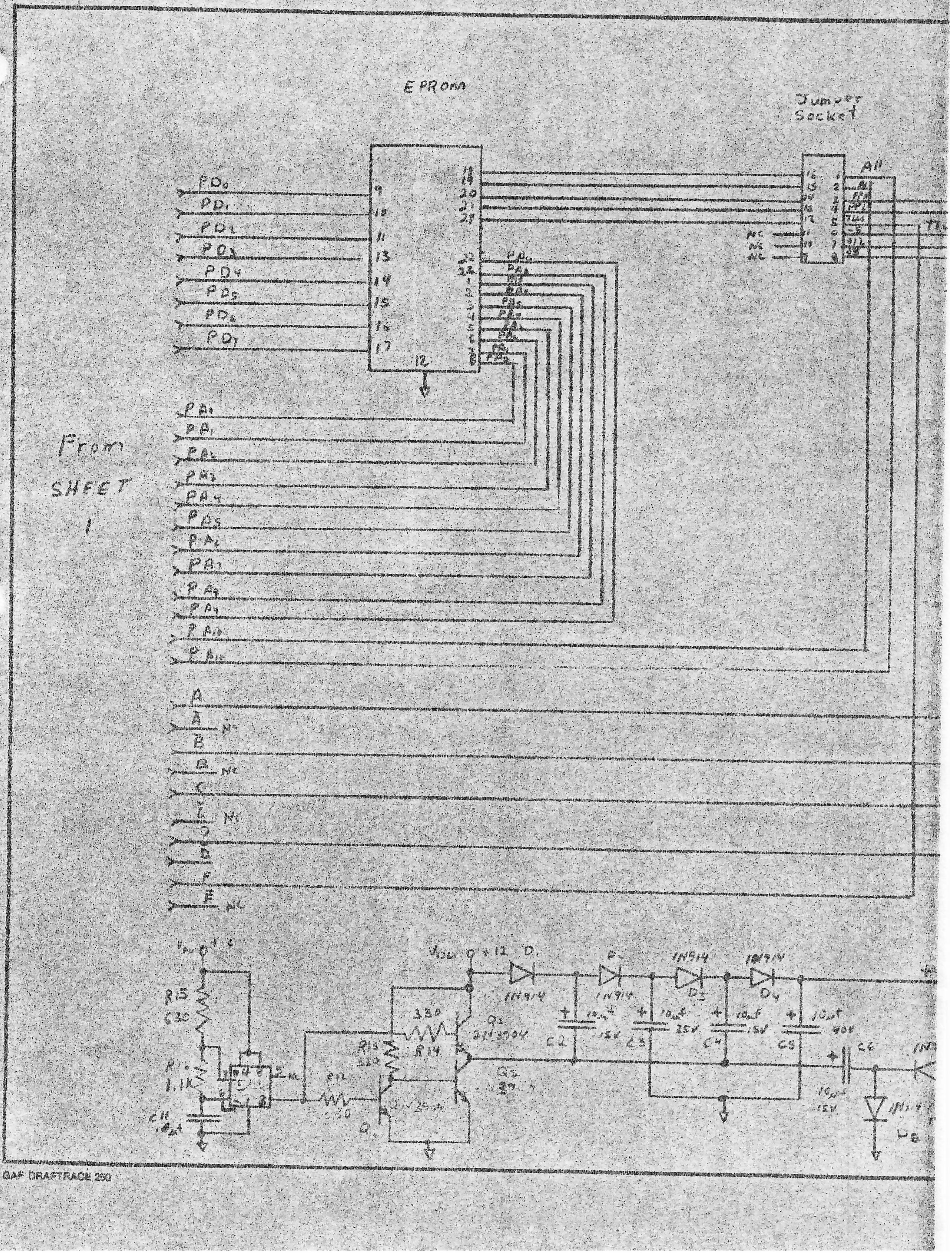


GAF DRAFTRACE 250



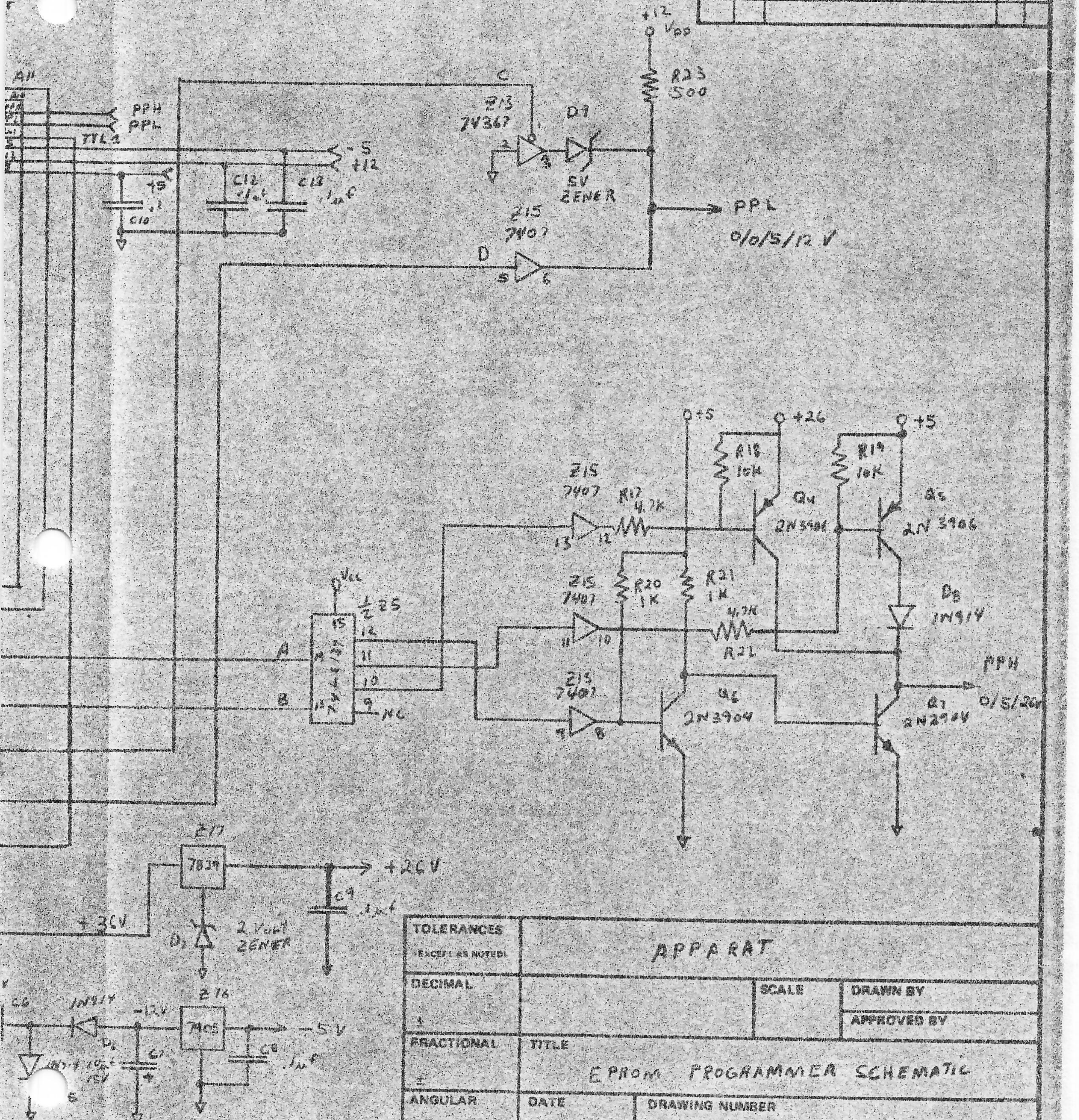
| DATE | SYM | REVISION RECORD | DR. | CK. |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |

| | | | |
|---|---|---|---|
| TOLERANCES<br>(EXCEPT AS NOTED) | APPARAT | | |
| DECIMAL<br>± | | SCALE | DRAWN BY M. Wade<br>APPROVED BY |
| FRACTIONAL<br>± | TITLE<br>EPROM Programmer Schematic | | |
| ANGULAR<br>± | DATE<br>12-18-80 | DRAWING NUMBER<br>SHEET 1 | |



GAF DRAFTRACE 250

| DATE | SYM | REVISION RECORD | DR. | CK. |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |



TOLERANCES (EXCEPT AS NOTED)

APPARAT

DECIMAL ±

SCALE

DRAWN BY

APPROVED BY

FRACTIONAL ±

TITLE EPROM PROGRAMMER SCHEMATIC

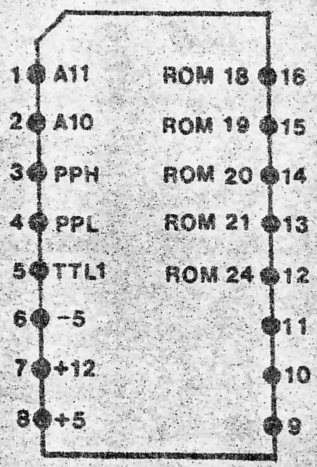
ANGULAR ±

DATE 12-18-80

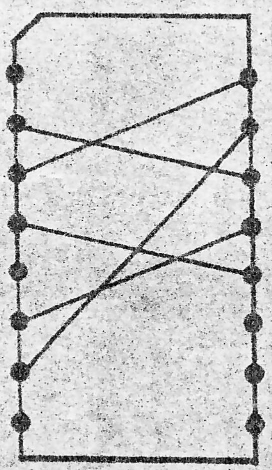
DRAWING NUMBER SHEET 2

# PARTS LIST

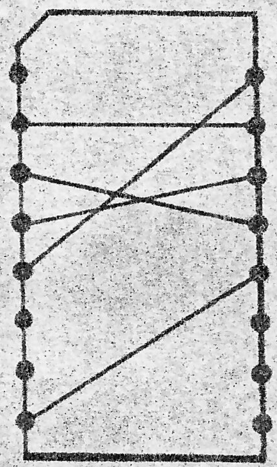
| | |
|---|---|
| Z1, Z14, Z15 | 7407 |
| Z2 | 7404 |
| Z3 | 7430 |
| Z4 | 7432 |
| Z5 | 74LS139 |
| Z6 | 7474 |
| Z7 | 74175 |
| Z8, Z9, Z10 | 7475 |
| Z11 | 74LS367 |
| Z12, Z13 | 74LS367 |
| Z16 | 555 |
| Q1, Q2, Q3, Q6, Q7 | 2N3904 |
| Q4, Q5 | 2N3906 |
| CR1, CR2, CR3, CR4, CR5, CR6, CR8 | 1N914 |
| CR7 | 3.3 Volt ½ Watt Zenner |
| CR9 | 5.1 Volt ½ Watt Zenner |
| R1 | 5K ¼ Watt |
| R2, R20, R21 | 1K ¼ Watt |
| R3, R4, R5, R6, R7, R8, R9, R10, R11 | 270 r ¼ Watt |
| R12, R13, R14 | 330 r ¼ Watt |
| R15 | 630 r ¼ Watt |
| R16 | 1.1K ¼ Watt |
| R17, R22 | 4.7K ¼ Watt |
| R18, R19 | 10K ¼ Watt |
| C1, C10, C11, C12, C13 C14, C15, C16, C17, C18, C19, C20, C21 | .1 Mfd 25V |
| C2, C4, C6, C7 | 47uf 16V |
| C3, C5 | 47uf 50V |
| C8, C9 | .1 Mfd 25V |



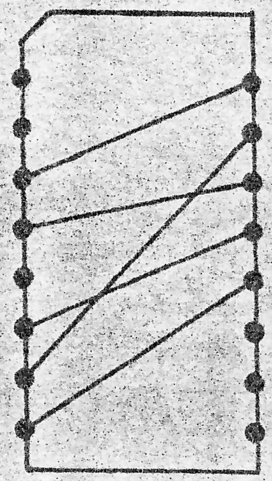
SIGNALS



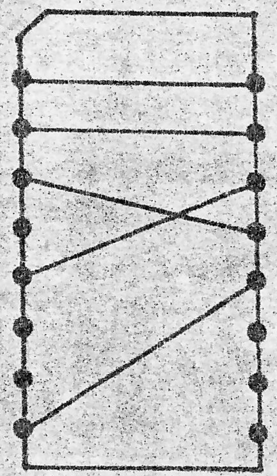
2716
3 Voltage



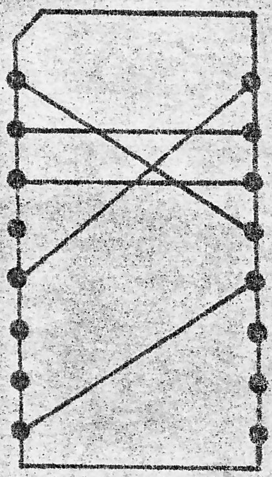
2716
5 Volt Only



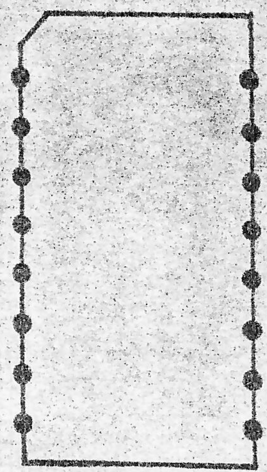
2708

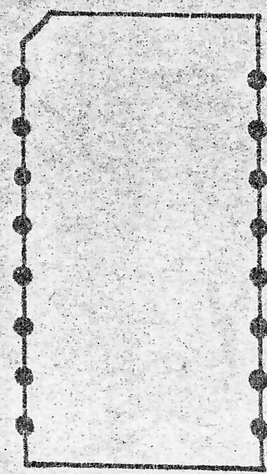


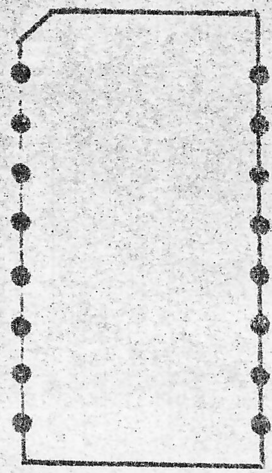
2532



2732







# THE A.P.B.

# APPARAT PROM BLASTING SYSTEM

## INTRODUCTION

The A.P.B system is the most versatile and cost effective PROM programmer on the market. The system will program virtually all of the common 24 pin EPROMs. Each PROM type is selected by a special Personality Module that adapts the programmer to the PROM. The following EPROMs can be programmed:

2508
2516
2532
2704
2708
2716 (5 volt)
2716 (3 voltage)
2732

The PROM programmer is a complete package. It includes powerful and easy to use software and a complete set of Personality Modules. There are no extra logic modules to buy.

With the unique combination of Persolality Modules and sophisticated software, the PROM programming system will perform many operations impossible with other systems. Following is a list of operations that can be performed:

Verify PROM is erased
Read PROMs
Copy PROMs
Copy between different PROM types
Program PROMs
Partial PROM programming, even 2704 and 2708
Do partial PROM copies
Verify that programming is correct
Read or save PROM content using disk
Program directly from EDTASM assembler object code
Program directly from computer memory
Examine and/or modify working memory
Preset content of working memory

The PROM programmer package consists of:

- A Programmer Card that attaches, by bus extender or ribbon cable, to a card edge connector of the Model I Expansion Interface, or Model III CPU.
- A complete set of PROM Personality Modules.
- Software on disk.
- This instruction manual.

The software provided will execute under all TRSDOS type operating systems.

## USING THE A.P.B.

Before you proceed with the operation of the A.P.B., it is advisable to make a copy of the disk provided and set it aside. You may use the DOS disk copy function for this purpose. It is also advisable to make some practice runs through the software to familiarize yourself with it's operation.

## GENERAL

The A.P.B. package consists of three items! the Programmer Card, a set of Personality Modules, and disk based software. The Persolality Modules are 16 pin plug-in adapters that configure the programming socket to match the type of PROM being programmed. The Personality Modules plug into the Programming Card and adapt it to the characteristics of each PROM type. PROMs are programmed by inserting the PROM into the card, selecting the proper Personality Module, and running the software.

## !!!WARNING!!!

PROMs can be damaged by static electricity.
Never install or remove a PROM with the power ON.
Never install or remove the A.P.B. card with the power ON.

Programming a PROM consists of seven steps!

1) Erase the PROM.
2) Set up programmer.
3) Verify that the PROM is erased.
4) Load data to be written to the PROM (BLASTED).
5) Do actual "BLAST".
6) Verify correct "BLAST".
7) Remove PROM.

## ERASE PROM

All of the PROMs programmed with the A.P.B. system are erased using ultraviolet light. Each PROM has a small glass window. Directing ultraviolet light through this window erases the PROM. In the erased state, each byte of PROM storage is set to hexidecimal 'FF'.

Ultraviolet PROM erasers are available from most computer stores. Characteristically, PROM erasure takes from half an hour to an hour, depending upon the intensity of the eraser. To insure complete erasure, follow the manufacturers instructions.

## SET UP PROGRAMMER

With the control lever in the open position, install the PROM to be programmed in the zero insertion force socket. Be sure that pin one of the PROM is aligned with the index mark on the socket. Lock the PROM in the socket by moving the control lever to the locked position.

Select the Personality Module from the chart below and install it in the 16 pin socket located to the immediate right of the socket wherein the PROM is now mounted. Be sure that the index mark is aligned with pin one.

## LOADING THE SOFTWARE

Boot the computer system. When "DOS READY" is displayed on the monitor, enter "BLASTER" at the keyboard. The A.P.B. software will be loaded from disk. Once loaded, the software will require initialization. A sequence of menus and prompts will be displayed to dirct the initialization.

## CPU SPEED IDENTIFICATION

Various CPU "speed-up" kits for the TRS-80 have been marketed. Your machine may have one of these modifications installed. Further, there is CPU speed sensitive program logic within the A.P.B. software. For these reasons, identification of the internal speed of your CPU is necessary. The following prompt/menu is provided so that identification can be easily accomplished.

```
APPARAT PROM BLASTER

IDENTIFY CPU CLOCK SPEED

0 - 1.8 MHZ
1 - 2 MHZ
2 - 2.25 MHZ
3 - 2.5 MHZ
4 - 2.75 MHZ
5 - 3 MHZ
6 - 3.25 MHZ
7 - 3.5 MHZ
8 - 3.75 MHZ
9 - 4 MHZ
```

The numeric value to the left of the displayed speed that most closely approximates the internal speed of your CPU should be entered. (1.8 MHZ represents the unmodified Model I CPU speed. An unmodified Model III operates at 2 MHZ)

## PROM IDENTIFICATION

Unique software logic must be exercised for each PROM type that can be programmed using the A.P.B. package. Consequently, the PROM type must be identified to the software. The following prompt/menu is provided to simplify PROM identification.

```
IDENTIFY PROM TYPE

0 - 2508
1 - 2516
2 - 2532
3 - 2704
4 - 2708
5 - 2716 - 5 VOLTS
6 - 2716 - 3 VOLTAGES
7 - 2732
```

The numeric value to the left of the appropriate displayed PROM identifier must be entered.

## ADDRESS ENTRY REQUESTS

Several of the user selectable A.P.B. functions require user specification (keyboard entry) of an operand address or both "STARTING" and "ENDING" operand addresses. All required addresses are requested by individual prompt. All but one required address (the main memory "START" address of the "FROM" operand in the "COPY MEMORY INTO ARRAY" function) are range checked relative to the size of the active PROM.

All entered addresses are of a singular form, that being four hexidecimal characters followed by depression of the "ENTER" key. Leading zeros must be provided. Depression of the backspace character (left arrow) will blank all previously entered address characters, enabling re-entry of the entire address.

An entered address failing the PROM size range check will result in the diagnostic:

ADDRESS EXCEEDS SPECIFIED PROM SIZE - RE-ENTER

Where both "STARTING" and "ENDING" addresses are requested, both addresses are range checked. Further, the addresses are tested to determine that the "ENDING" address is not less than the "STARTING" address. A smaller specified "ENDING" address will result in the diagnostic:

ADDRESS ENTRY SEQUENCE ERROR. RE-ENTER? <Y/N>

A keyboard response is required, where:

Y = Return to request entry of the "STARTING" address.
N = Return to display the MAIN MENU.

## Function 2 - BLAST ROM FROM ARRAY

In processing this function, "STARTING" and "ENDING" PROM addresses are requested. The content of the ARRAY, within the specified address range, is written to the active PROM. On completion of the "BLAST", the user is apprised of the condition and queried regarding further action. The advisory/query message is:

PROGRAMMING COMPLETE. CONTINUE? <Y/N>

A keyboard response is required, where:

Y = Return to the MAIN MENU.
N = Exit to DOS.

As differing PROM's may be of differing sizes, partial PROM "BLASTS" may be specified, and hexidecimal 'FF' characters found in the ARRAY are never written to the PROM, differing timings may be experienced using this function. As a close approximate, 20 bytes/second may be used to predict the duration of a PROM "BLAST". Maximum timings for the various PROMs are:

| PROM | DURATION |
|---|---|
| 2704 | 26 Seconds |
| 2508/2708 | 52 Seconds |
| 2516/2716 | 103 Seconds (1 minute, 43 seconds) |
| 2532/2732 | 205 Seconds (3 minutes, 25 seconds) |

As several minutes can seem a very long time to watch a machine do an apparent nothing, a one second ON, one second OFF "blip" is displayed on the monitor during PROM "BLASTS".

A "BLAST" can be aborted by holding the (X) key depressed until the MAIN MENU is displayed.

## Function 3 - VERIFY CORRECT BLAST

This function requires no initial keyboard input as PROM/ARRAY addresses are assumed from either:

- The specified addresses of a preceeding "BLAST".
- The active PROM's size if no address range was previously specified.

In either case, the content of the ARRAY is moved to a secondary work area and the appropriate area of the PROM is read into the evacuated ARRAY. The content of the two memory areas are compared and disparate content is identified to the user, as with Function 0 - VERIFY PROM IS ERASED. Function completion conditions are identical to those of Function 0 as well.

## Function 6 - COPY MEMORY INTO ARRAY

This function allows main memory content, internal or external to the ARRAY, to be moved into the ARRAY to a length derived from the user specified "STARTING" and "ENDING" ARRAY addresses. Areas may overlap in any way.

On activation, the user is prompted to:

ENTER "TO" START ADDRESS
ENTER "TO" END ADDRESS
ENTER "FROM" START ADDRESS

Address checks and diagnostics generation are as detailed in the section "ADDRESS ENTRY REQUESTS". "FROM" is an absolute main memory address.

With satisfactory addresses, the main memory-to-ARRAY move is performed. Termination of the function is identical to that of Function 1 - READ ROM INTO ARRAY.

## Function 7 - MOVE ARRAY INTERNAL DATA

This function is identical to that of FUNCTION 6 - COPY MEMORY INTO ARRAY, except that both "TO" and "FROM" areas must be wholly located internal to the ARRAY, as scaled to the active PROM's size. Both "TO" and "FROM" addresses are ARRAY relative.

## Function 8 - EXAMINE/EDIT ARRAY

On activation, the user is prompted to :

ENTER STARTING ADDRESS

The entered address is validated, as detailed in the section "ADDRESS ENTRY REQUESTS". A valid address is rounded down to the nearest hexidecimal 100. A hexidecimal 100 bytes of ARRAY content, from the rounded address, are displayed on the monitor. Address references are displayed to the left of the ARRAY content.

Once display generation is complete, the "EDIT" mode is entered. Editing is accomplished by moving the blinking cursor to the desired location, internal to the display, and typing any valid hexidecimal value(s) over the display content. When editing is complete, the modified content of the display can be committed to the corresponding area of the ARRAY by depressing the "ENTER" key and responding to the resulting query:

PERMANENT? <Y/N>

A keyboard response is required, where:

Y = Commit display content to the ARRAY. Return to "EDIT" mode.
N = Return to "EDIT" mode.

(Only the hexidecimal 100 bytes currently displayed are committed to the ARRAY when "Y" is depressed)

## EXTENDED FUNCTIONS

Combinations of previously detailed keyboard selectable functions may be exercised to accomplish an extended set of PROM programming actions. Some of those actions are:

- Copy ROMs
- Copy between differing ROM types (same size)
- Copy two small PROMs to one large PROM
- Copy one large PROM to two small PROMs
- Do partial copies
- Program from computer (main) memory
- Program from the assembler

These actions might be accomplished as detailed following:

### Copy ROMs

This action requires that the source ROM be read to the ARRAY, in its entirety, using the "READ ROM INTO ARRAY" function. The "SAVE ARRAY ON DISK" function can then be exercised to preserve the ROM content. After dismounting the source ROM and mounting the destination ROM (observing the disk dismount, power OFF, ROM change, power ON, disk mount, system boot, A.P.B. initialization sequence), the "LOAD MEMORY FROM DISK" function can be exercised to bring the "SAVED" program back into the ARRAY. The "BLAST ROM FROM ARRAY" function can be exercised to complete the action.

### Copy Between Different ROM Types (same size)

This action is identical to the preceeding, excepting the reinitialization of the A.P.B. software (subsequent to the destination PROM mount sequence), where the differing PROM type must be identified. The "LOAD MEMORY FROM DISK" function will reload the previously "SAVED" program, to the programs length, irrespective of the destination PROM's size.

### Copy Two Small PROMs To One Large PROM

The events necessary to accomplish this action are as follow:

1. Save the content of both source PROMs on disk (using unique FILESPECs).

2. Mount the destination PROM.

3. Using the "LOAD MEMORY FROM DISK" function, load the "SAVED" content of the source PROM that is to be the content of the upper half of the destination PROM.

4. Using the "MOVE ARRAY INTERNAL DATA" function, move the content of the lower half of the ARRAY to the upper half.

5. With the "LOAD MEMORY FROM DISK" function, load the "SAVED" content of the source PROM that is to be the content of the lower half of the destination PROM.

6. If more than one destination PROM is to be "BLASTED", the combined content of the ARRAY should be saved on disk.

7. The destination PROM can be "BLASTED".

## HARDWARE NOTES

The A.P.B. has been designed to "BLAST" all of the common 24 pin PROMs. Most of the pins on these PROMs are identical. Differences that do exist are with pins 18, 19, 20, 21, and 24. These pins require variable voltages and logic levels determined by the PROM type. These pins are connected to the Personality Module socket on the Programmer Card. Other pins in the Personality Module socket provide special logic levels and voltages required by the individual PROM types. By installing headers with appropriate jumpers, proper levels can be provided to the appropriate PROM pins for each individual PROM type.

The following signals are available at the Personality Module socket:

- A11 Address Bit 11 - Provides most significant bit for 4K PROMs.
- A10 Address Bit 10 - Provides most significant bit for 2K PROMs.
- PPH High Program Pulse - Can be switched between 0, 5, 26 volts.
- PPL Low Level Program Pulse - Can be switched between 0, 5, 12 volts.
- TTL1 TTL Level Pulse - Used for Chip Select, Output Enable, etc.
- -5V Minus 5 volt power supply line.
- +12V Plus 12 volt power supply line.
- +5V Plus 5 volt power supply line.

## PROM ADDRESSING

Addresses for programming and reading PROMs are provided by a 12 bit binary register on the Programmer Card. This takes the active PROM off the TRS-80's address bus and allows the user to reference PROM addresses without interfering with the CPU's ROM area. The register is loaded with a two byte address under software control.

The Programmer hardware uses four port accessed registers to control PROM programming and reading. The registers are accessed as follows:

| PORT ADDRESS | REGISTER FUNCTION |
|---|---|
| 3C | Control |
| 3D | PROM Address (low) |
| 3E | PROM Address (high) |
| 3F | PROM Data |

## APPENDIX I

Writing a Specialized Driver:

For most applications the drivers supplied are adequate. Occasionally, however, the user has a highly specialized application that requires special interaction with the outside world; in this case the APB software allows a custom driver to be loaded and used.

The custom driver has 4 links in which it can communicate with the APB PROM software. These are: Write, Read, Blank, and Extra. With Write, Read, and Blank, parameters are passed back and forth. Extra calls your routine. To get to Extra, just type 'Z' from the Main Menu.

The parameters passed are as follows:

```
Write:   BC  = byte count
         HL  = relative PROM address
         DE  = start of memory to be written
         Return:
                z = no errors in verify
               nz = error in verify

Read:    BC  = byte count
         HL  = relative PROM address
         Return:

               BC bytes written into memory starting
               at 7000h

Blank:   BC  = byte count
         HL  = relative PROM address
         Return:

                z = blank
               nz = not blank
```

To hook your routine into APB, you must place the address of the start of your routines into memory starting at 5201h (see below). To return to APB, simply execute a Return.

```
100     DEFB     OC3H
200     DEFW     WRITE  ; Your write routine w/ label WRITE
300     DEFB     OC3H
400     DEFW     READ
500     DEFB     OC3H
600     DEFW     BLANK
700     DEFB     OC3H
800     DEFW     EXTRA  ; Your routine
900; The reason for the Oc3h (jp) and not an Ocdh (call) is
999; because the program calls these addresses; e.g., call 5200h
```

To talk to the PROM, the port #s are as follows:

| | |
|---|---|
| 60 | PROM type value/data |
| 61 | PROM relative low value |
| 62 | PROM relative high value |
| 63 | Data to write |

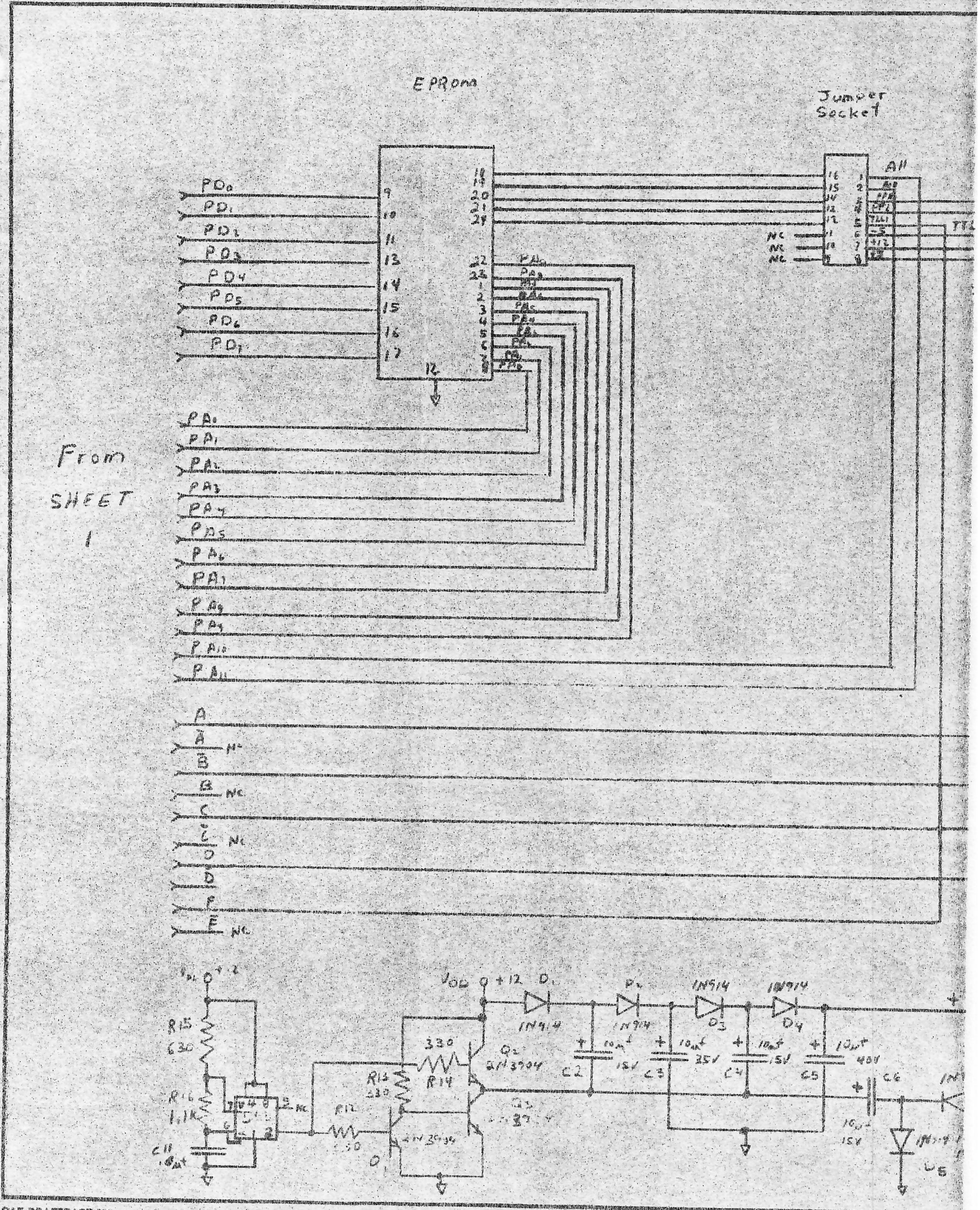
| PROM type values: | X | Y | Z |
|---|---|---|---|
| 2708 | x | y | z |
| 2716 | x | a | f |

```
Before doing anything:      OUT  (C=60),X
Before doing a Write:       OUT  (C=60),Y ; this turns Write on, so
                            OUT  (C=60),Z ; have everything already
                            OUT  (C=60),0 ; done after Write
```

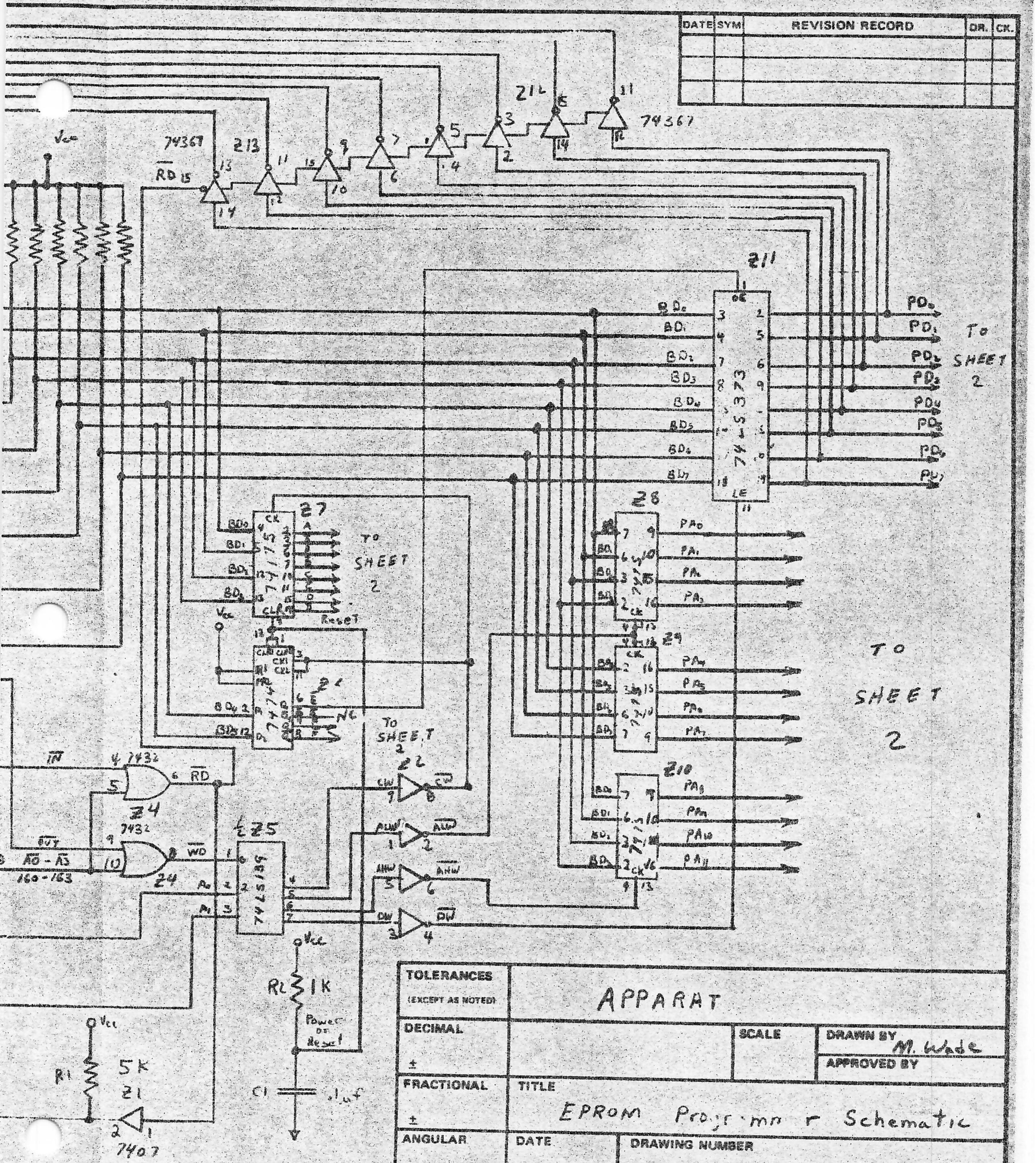
```
A few extras:
CALL 520Fh    ; save byte in A to memory.  First call save to
              ; 7000h, next at 7001h, etc.
(5212h) stores 16 bit counter for 520Fh.
(5214h-5614) is a 400h byte butter
```

Questions, comments and suggestions, send to:

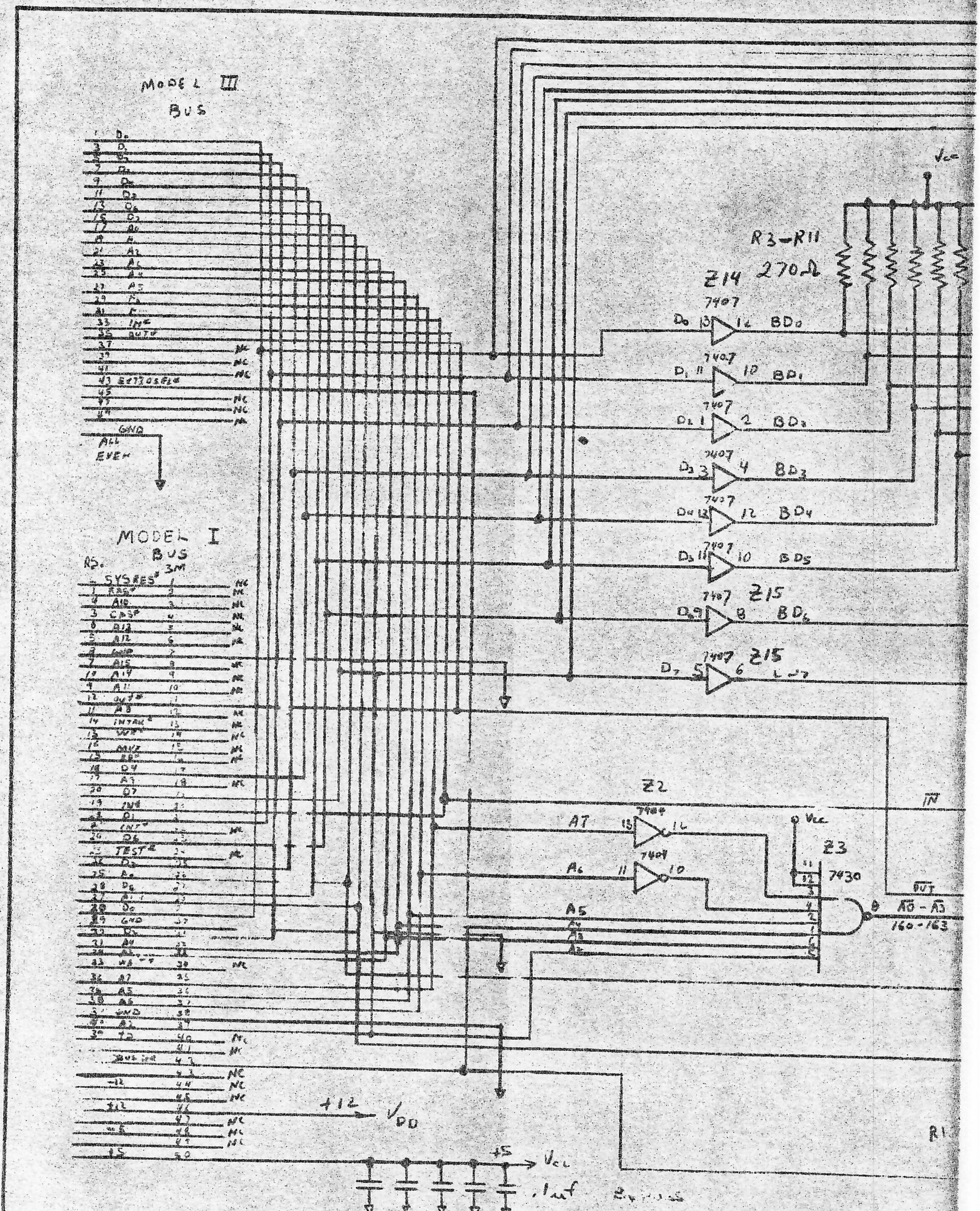
Apparat, Inc.
4401 S. Tamarac Pkwy.
Denver, CO  80111


EPROM
Jumper Socket
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
9
10
11
13
14
15
16
17
18
19
20
21
24
22
23
1
2
3
4
5
6
7
8
12
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
From SHEET 1
16
15
14
13
12
NC
NC
NC
A11
TTL
-5
+12
A
A NC
B
B NC
C
C NC
D
D
F
E NC
VCC +12
R15 630
R16 1.1K
C11 .01µf
2 NC
R12
Q1 2N3904
R13 330
330 R14
Q2 2N3904
Q3
VDD +12
D1 1N914
D2 1N914
1N914 D3
1N914 D4
C2 10µf 15V
C3 10µf 35V
C4 10µf 15V
C5 10µf 40V
C6 10µf 15V
1N914
U5

| DATE | SYM | REVISION RECORD | DR. | CK. |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |

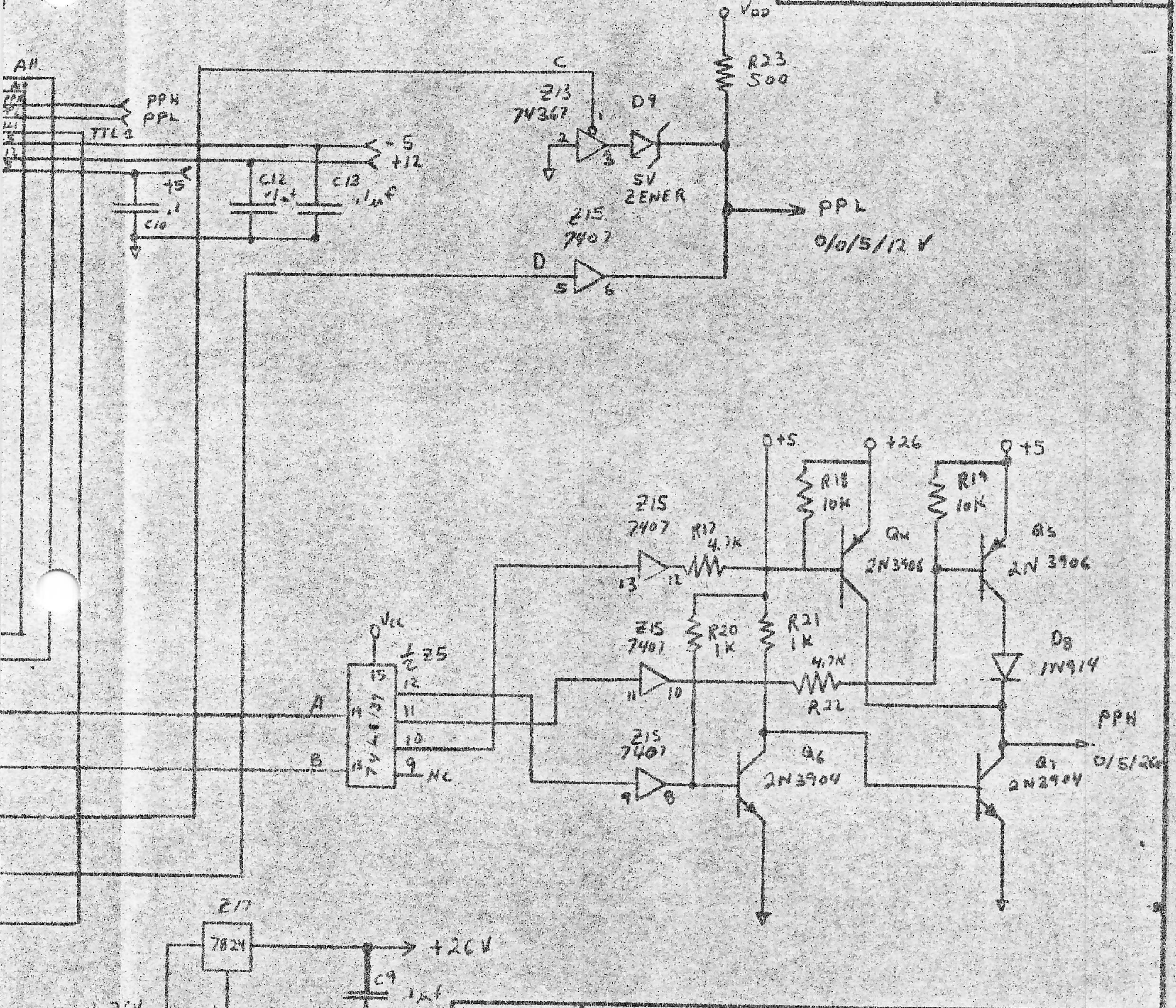


| TOLERANCES (EXCEPT AS NOTED) | APPARAT | | |
|---|---|---|---|
| DECIMAL ± | | SCALE | DRAWN BY M. Wade<br>APPROVED BY |
| FRACTIONAL ± | TITLE EPROM Progr·mn r Schematic | | |
| ANGULAR ± | DATE 12-18-80 | DRAWING NUMBER SHEET 1 | |



GAF DRAFTRACE 250

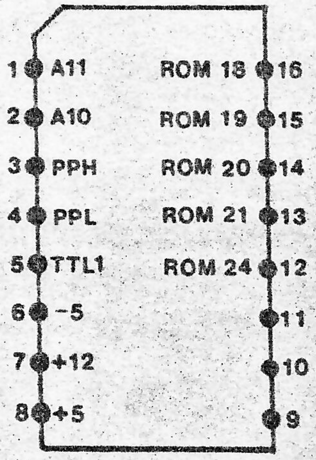
| DATE | SYM | REVISION RECORD | DR. | CK. |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |



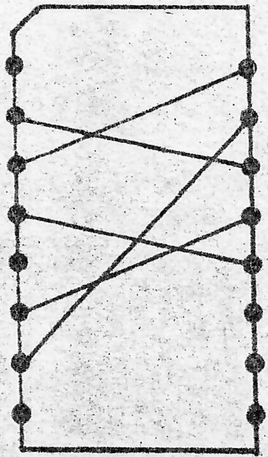
| TOLERANCES (EXCEPT AS NOTED) | APPARAT | | |
|---|---|---|---|
| DECIMAL<br>± | | SCALE | DRAWN BY<br>APPROVED BY |
| FRACTIONAL<br>± | TITLE EPROM PROGRAMMER SCHEMATIC | | |
| ANGULAR<br>± | DATE 12-18-80 | DRAWING NUMBER SHEET 2 | |

# PARTS LIST

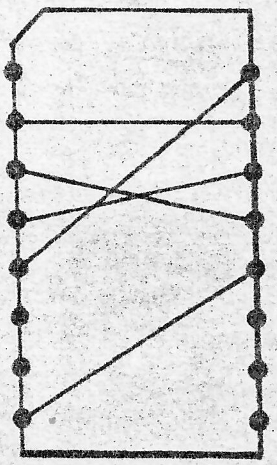
| | |
|---|---|
| Z1, Z14, Z15 | 7407 |
| Z2 | 7404 |
| Z3 | 7430 |
| Z4 | 7432 |
| Z5 | 74LS139 |
| Z6 | 7474 |
| Z7 | 74175 |
| Z8, Z9, Z10 | 7475 |
| Z11 | 74LS367 |
| Z12, Z13 | 74LS367 |
| Z16 | 555 |
| Q1, Q2, Q3, Q6, Q7 | 2N3904 |
| Q4, Q5 | 2N3906 |
| CR1, CR2, CR3, CR4, CR5, CR6, CR8 | 1N914 |
| CR7 | 3.3 Volt ½ Watt Zenner |
| CR9 | 5.1 Volt ½ Watt Zenner |
| R1 | 5K ¼ Watt |
| R2, R20, R21 | 1K ¼ Watt |
| R3, R4, R5, R6, R7, R8, R9, R10, R11 | 270 r ¼ Watt |
| R12, R13, R14 | 330 r ¼ Watt |
| R15 | 630 r ¼ Watt |
| R16 | 1.1K ¼ Watt |
| R17, R22 | 4.7K ¼ Watt |
| R18, R19 | 10K ¼ Watt |
| C1, C10, C11, C12, C13 C14, C15, C16, C17, C18, C19, C20, C21 | .1 Mfd 25V |
| C2, C4, C6, C7 | 47uf 16V |
| C3, C5 | 47uf 50V |
| C8, C9 | .1 Mfd 25V |



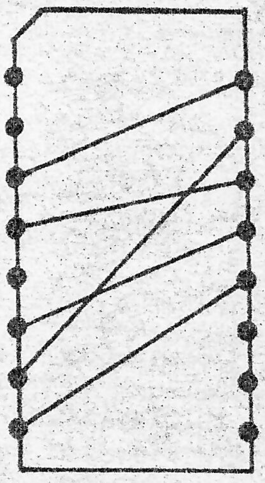
SIGNALS



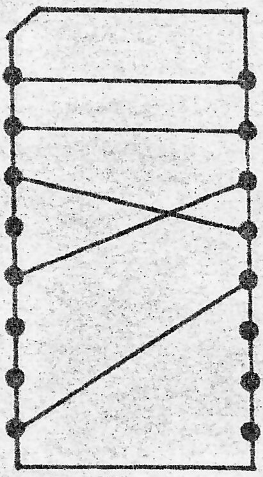
2716
3 Voltage



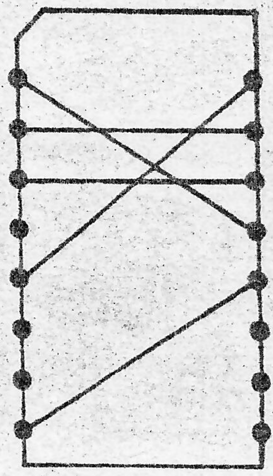
2716
5 Volt Only



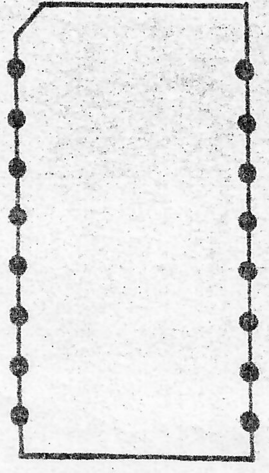
2708

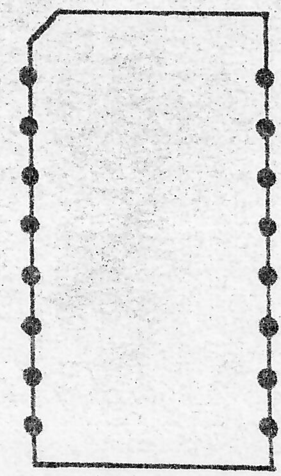


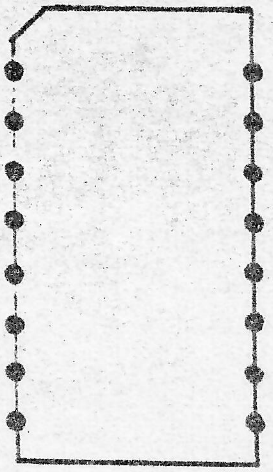
2532

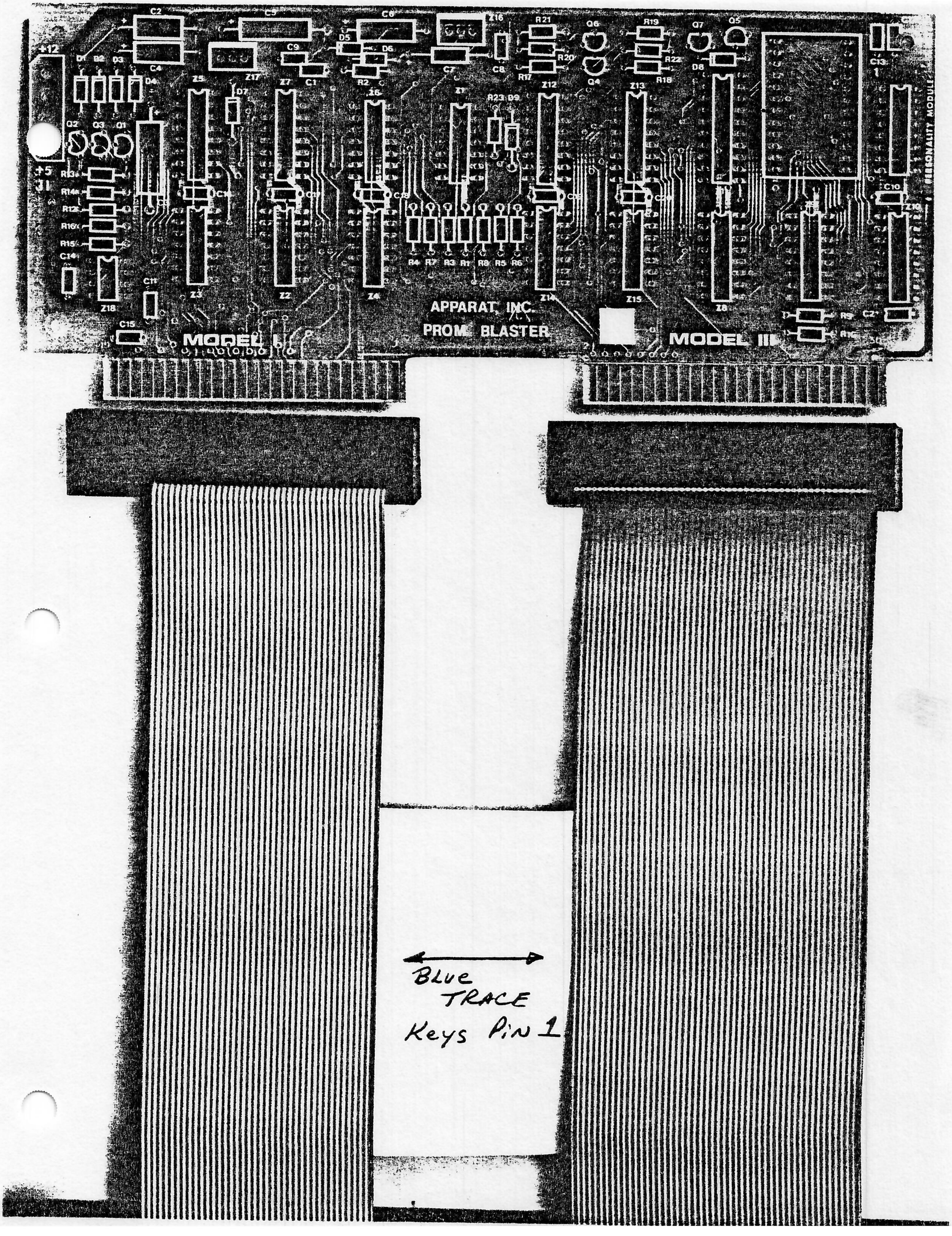


2732

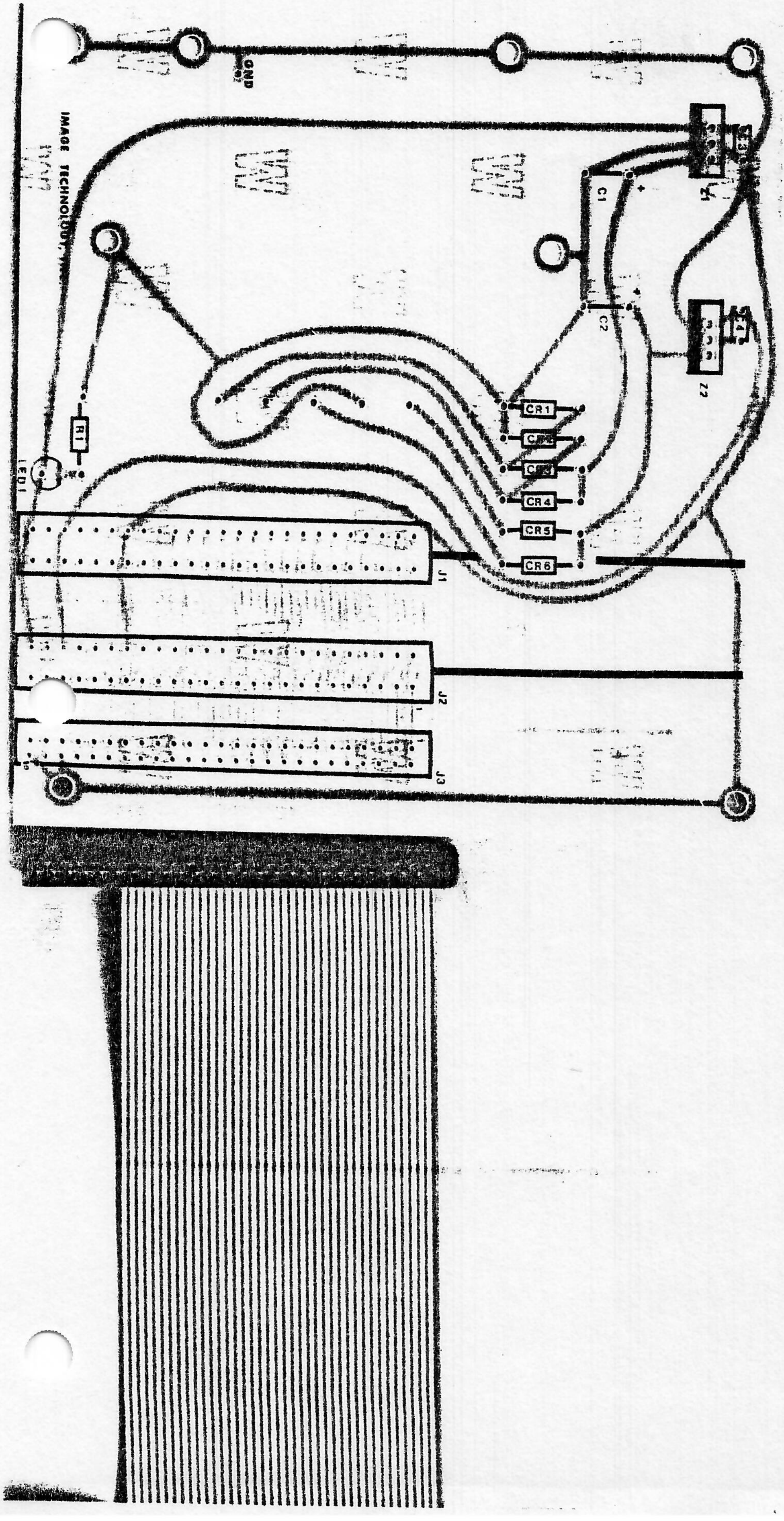








BLUE TRACE Keys Pin 1


GND
IMAGE TECHNOLOGY, INC
C1
C2
+
Z1
Z2
R1
LED1
CR1
CR2
CR3
CR4
CR5
CR6
J1
J2
J3