PROM BLASTER

Apparat, Inc.

Apparat Prom Blasting System

ADDENDUM

This addendum is intended to introduce some minor changes in the A.P.B. software. We will also take this opportunity to answer some of the more common questions that we have received about the system.

SOFTWARE

The A.P.B. software has now been rewritten to operate in the 16 sector format as well as the original 13 sector format. The software package now consists of two disks. They are identical, except that one operates under DOS 3.2 and the other under DOS 3.3. If you have 3.3 ROMs in your disk controller, use the 16 sector A.P.B. software. Otherwise, use the 13 sector software.

The new software operates in the same manner as the old software described in the manual. Since the software is now available in the 16 sector format, it is now possible to read binary files from DOS 3.3 disks.

Several minor changes have been made to the software:

1) Three bugs in the binary read routines were fixed.

2) A version number is now printed when the software is started. The version number looks something like this:

VERSION: 2T13

The version number indicates three things. The first digit is the actual version number which is changed each time the software is updated. The character will be "T" or "A" to indicate whether the software is the Apex or the Apple DOS turnkey version. The last two digits indicate 13 or 16 sector versions.

3) The software prints out the starting address of the working array. This allows programs external to the A.P.B. software to access the working array. For example, you might want to disassemble part of the working array using the disassembler in Apple machine language monitor.

To conserve memory, the working array doesn't necessarily begin on an even page boundry. The working array is a full 8K in size.

Remember that the A.P.B. software has built in routines to edit, copy and preset the working array. It is only necessary to access the working array under special circumstances.

If you hit reset while the A.P.B. software is running, you will be taken to the machine language monitor not to BASIC. The starting address of the A.P.B. software is $900 hex. To restart the A.P.B. software after you have reset the Apple type:

900G

followed by Return.

WHAT IS THAT CLICKING SOUND?

We have had several people ask about the clicking sound that issues from the Apple when the A.P.B. software is running. This sound is no cause for alarm. The software "clicks" the speaker every time a disk sector is written or read. The sound is meant to be audio feed back. As an example, a bad sector on the disk is immediately apparent because of the disruption in the rhythm of the clicks.

ROM TYPES AND COMPATABILITY

All of the EPROMs that are programmable on the A.P.B. system are part of the same family of ROMs. All of these ROMs are erased with ultraviolet light. All of the ROMs are programmed with a 26 volt programming voltage, and all have roughly the same pinout. The only differences exist on pins 24, 21, 20, 19, and 18.

There are two main categories of ROMs within this family: the three voltage ROM and the five volt only ROM. The three voltage ROM requires +12, +5 and -5 volts to power the device. The five volt only requires +5 volts for power. Here is a list of ROMs in each category:

THREE VOLTAGE TYPE	FIVE VOLT ONLY
2704	Intel 2716
2708	2732
TI 2716	2508
	2516
	2532

I2716 VERSUS TI2716

As you can see from the above chart, there two versions of EPROM with the number 2716. One is a five volt only ROM; the other requires three voltages. The five volt only version was developed by Intel and the three voltage type by Texas Instruments. These two types of ROMs are NOT at all compatible. Before using a 2716 make sure that you have the correct type for your application.

CUSTOM APPLE ROMS

The Apple II uses 2K masked programmed ROMs to hold all of its system software. These ROMs are very similiar to the 5 volt only 2716, but the chip selects are of a different polarity, and they cannot be plugged directly into the Apple board. They can however, be used with an inexpensive adapter.

In order that the EPROM operate properly in the Apple, pin 18 must be grounded and pin 21 connected to +5 volts. This can be accomplished by clipping the pins 18 and 21 short and soldering short pieces of wire wrap wire to the appropriate voltage. (pin 18 to 12 and pin 21 to 24). This is not the most practical solution, since once the pins have been cut, the ROM cannot be reprogrammed. A more practical approach is to use a header that can be plugged into the ROM socket and which, in turn, can have a ROM plugged into it. Then the pins of the header can be clipped and soldered. This way the ROM pins are undamaged and the ROM can be removed from the header at any time for reprogramming.

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 ROM type is selected by a special personality module that adapts the programmer to the ROM. The following EPROMs can be programmed:

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

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 modules to buy.

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

Verify ROM is erased
Read ROMs
Copy ROMs
Copy between different ROM types
Program ROMs
Partial programming, even 2704 and 2708
Do partial copies
Verify that programming is correct
Read or save ROM data on disk
Read or save ROM data on cassette tape
Program directly from the APEX assembler
Program directly from computer memory
Examine and/or modify working memory
Preset working memory

The PROM programmer package consists of an interface card that plugs into one of the Apple II expansion slots, a

complete set of personality modules, software on disk and instruction manual. The software will run under under Apple DOS or the APEX operating system.

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 the Apple DOS disk copy to make the copy. 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 personality modules are 16 pin plug-in adapters that reconfigure 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 ROM type. ROMs are programmed by inserting the ROM into the card, selecting the proper personality module and running the software.

****

!!! WARNING !!!

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

****

Programming a PROM consists of eight steps:

1) Erase ROM.
2) Set up programmer.
3) Verify that the ROM is erased.
4) Load data to be blasted.
5) Do actual blast.
6) Verify correct blast.
7) Remove ROM.

ERASE ROM

All of the ROMs programmed by the APB system are erased using ultraviolet light. Each of ROM has a small glass window. Shining ultraviolet light through this window erases the ROM. In the erased state, every byte in the ROM is set to hex FF.

Ultraviolet ROM erasers are available from most computer stores. Erasing a PROM takes from a half hour to an hour, depending upon the intensity of the eraser. To insure complete erasure, follow the manufactures instructons.

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 ROM is aligned with the index mark on the socket. Lock the PROM 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 one.

PERSONALITY MODULE TABLE

ROM TYPE	MODULE TO USE
2704, 2708	2708
TI 2716 (3 voltage)	2716 (3 voltage)
2508	I2716 (5-volt)
2516	I2716 (5-volt)
Intel 2716 (5-volt)	I2716 (5-volt)
2532	2532
2732	2732

With the power off, install the A.P.B. card into an empty slot in the APPLE. You may use any slot but zero. If you are going to be programming several PROMs, install the programmer into a slot that gives clear access to the PROM socket.

VERIFY ROM IS ERASED

Turn the power on and boot the A.P.B. software disk. This is a 13-sector disk. If you have the Language Card or are using DOS 3.3, follow Apple's instructions for running 13-sector disks.

The PROM blasting software is automatically executed by the boot. The first thing the program will ask is into which slot you have placed the A.P.B. card:

ENTER SLOT NUMBER

Enter the slot number (should be between 1 and 7) followed by "RETURN".

The program will now print the "MAIN MENU" on the screen. Select option number 1, VERIFY ROM IS ERASED.

Next, the program will ask you for the ROM type. Enter the number that corresponds to the type of ROM that you are programming.

The program will now begin testing the ROM. If the ROM is fully erased, all bytes will be hex FF. If the program finds any bytes which are incorrect, the address and the incorrect byte will be printed on the screen.

If the erase is correct, go on to the next step. If the erase is incorrect, check the following:

Card in wrong slot.
ROM in socket backwards
Wrong personality module
Personality module in backwards
Try erasing for a longer period of time

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

LOAD DATA

The data to be written in the ROM must be loaded into a working array inside the program. To make things flexible, the data can be loaded in several different ways from several different sources:

From disk file
From another ROM (see Copy section)

Directly from memory
From the keyboard
From cassette tape
Directly from the APEX assembler
From Apple DOS binary file

Each of these techniques is described in detail in other sections of the manual. Choose the appropriate method, for your application.

DO ACTUAL BLAST

After you have loaded the data, return to the main menu. Select the "BLAST" option. The program again asks for the ROM type. The amount of time required to blast a ROM depends upon the ROM type. The 3 voltage ROMs (2704, 2708, and TI2716) require much more time than the other types. These ROMs require about one hundred passes to program. The software will indicate each pass as it's completed.

The single voltage ROMs only require a single pass, so they program much faster. No pass information is printed. The following are approximate programming times for each ROM type:

2704	2 min	45 sec
2708	5 min	20 sec
2508		53 sec
TI2716	10 min	40 sec
2516 or Intel 2716	1 min	46 sec
2532 or 2732	3 min	32 sec

VERIFY CORRECT BLAST

After the programming is complete, you should verify that all of the data has been correctly programmed into the ROM. To do this select the "TEST FOR CORRECT BLAST" option. The programmer will read every byte in the ROM and compare it with the data in the working array. Any difference will be printed on the screen.

If there are errors check the following:

Card in wrong slot.
ROM in socket backwards
Wrong personality module
Personality module in backwards
Try erasing for a longer period of time

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

REMOVE ROM

Turn Off The Apple !!!!

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

COPYING ROMS

Copying a ROM is similiar to the programming process described above. The process involves reading the ROM to be copied into memory, then saving the data on disk or tape. 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 ROM to be copied. This is described in detail in step two of the programming sequence above.

2) Boot the disk, enter the slot number and enter the main menu as described in step three above.

3) Select the "READ ROM INTO MEMORY" option. This will read the ROM into the working memory of the program.

4) Now select the "READ OR SAVE DATA" option. Save the data on disk or cassette tape, according to your application. Reading and saving data is described in detail in the software section of this manual.

5) Remove the ROM as described in step seven of the programming instructions above.

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

SOFTWARE PACKAGE

All of the programming operations of the A.P.B system are performed by a powerful and flexible software package. The software is included on a 5-inch disk as a part A.P.B. package. The software will run under both Apple DOS and the APEX operating system. Instructions for using the software under APEX are provided later in this manual.

THE WORKING ARRAY

Many of the operations of the A.P.B. package are performed between the A.P.B. card and an area of memory called the working array. The working array is simply memory space set aside by the program to hold program data. The working array can be read from disk, written to disk, read from memory, read from ROM, written to ROM and edited.

USING THE SOFTWARE

To use the software under Apple DOS, the disk is booted in the normal manner. The disk provided is a 13-sector disk. If you have the Language Card or are using DOS 3.3, follow Apple's instructions for running 13-sector disks.

The PROM blasting software is automatically executed by the boot. The first thing the program will ask is where you have placed the A.P.B. card:

ENTER SLOT NUMBER

Enter the slot number (should be between 1 and 7) followed by "RETURN".

The program will now print the "MAIN menu" on the screen:

```
********MAIN MENU********

1) VERIFY ROM IS ERASED
2) READ ROM INTO ARRAY
3) BLAST ROM FROM ARRAY
4) TEST FOR CORRECT BLAST
5) READ OR SAVE DATA
6) COPY MEMORY INTO ARRAY
7) EXAMINE OR EDIT ARRAY
8) PRESET WHOLE ARRAY
SELECT ONE:
```

VERIFY ROM IS ERASED

This operation verifies that the ROM has been completely erased. In it's erased state, every byte of the ROM is hex FF. The verify operation checks each byte and prints any byte that is not fully erased:

ERROR-ADDRESS:213 EXPECTED FF FOUND: 12

The address is the relative address from the beginning of the ROM.

Most erase errors are due to under erasure. When a ROM is under erased, the data will be more temperature sensitive. In some situations, the data will change as the temperature changes.

READ ROM INTO ARRAY

This operation reads the ROM into the working array. The data can then be edited, saved to disk etc.

BLAST ROM FROM

This option writes the data in the working array into the ROM. The amount of time required to blast a ROM depends upon the ROM type. The 3 voltage ROMs (2704, 2708, and TI2716) require much more time than the other types. These ROMs require about one hundred passes to program. The software will indicate each pass as it's completed.

PASS 37 OUT OF 104

The single voltage ROMs only require a single pass, so they program much faster. No pass information is printed. The following are approximate programming time for each ROM type:

2704	2 min	45 sec
2708	5 min	20 sec
2508		53 sec
TI2716	10 min	40 sec
2516 or Intel 2716	1 min	46 sec
2532 or 2732	3 min	32 sec

TEST FOR CORRECT BLAST

This option compares each byte in the ROM with each byte in the working array. This verifies that the data has been successfully written to the ROM. Any mismatch will be printed on the screen:

ERROR-ADDRESS:213 EXPECTED CD FOUND: 12

The address is the relative address from the beginning of the ROM. If all byte are correct, the program will respond with "GOOD BLAST".

READ OR SAVE DATA

This option allows you to read or write data from the disk or cassette tape. When you enter this option the program will print another menu indicating the choice of operations:

```
]]]-DISK MENU-[[[

1) EXAMINE ROM FILES
2) DELETE ROM FILES
3) WRITE TO DISK
4) READ FROM DISK
5) WRITE TO CASSSETTE
6) READ FROM CASSETTE
7) READ AN APPLE DOS FILE
0) QUIT
```

The ROM software maintains a set of files on the disk that are used to hold ROM data. Up to six files can be stored on a single disk at one time. Each file can be given a name up up to ten characters long. When you examine the ROM files on disk, the program will print the file list like this:

```
>>>>>>>FILES>

        1) GRAPHICS
        2) SINETABLE
        3)
        4)
        5)
        6)
```

In this example, files one and two contain active data, while the rest of the file spaces are vacant and can be used to store other ROM data.

The other commands allow you to read or write the working array to or from the disk. It also allows you to delete unwanted files.

READING APPLE DOS FILES

The A.P.B. software allows you to read files directly from Apple DOS disks. The file to be read must be a binary file.

To read a file from an Apple disk, select option number seven from the disk menu. Place the disk that you want to read into drive 1. You have two options: the first is to examine the catalog of the disk. Only the binary files on the disk will be displayed. You can examine the catalog of several different disks by switching disks and redisplaying the catalog.

The second option is to read one of the files into the working array. The file is selected by entering the number of the file to be read. The data is entered directly into the working array so that you can perform any of the normal operations including editing, blasting and saving the data in a standard ROM file. When you exit from the Apple disk read routine, be sure to put the ROM disk back into the drive.

CASSETTE TAPE

The software also provides the option of reading and writing ROM data to cassette tape. The operation is similiar to tape reads and saves from Applesoft. Here are some tips to insure good reads and writes:

> When writing, start the tape past the clear leader. Start the recording before starting the write.
>
> When reading, start the tape just past the start of the leader tone. Start the recorder running before starting the read.

COPY MEMORY INTO ARRAY

This option allows you to copy data from any part of memory directly into the working array. This allows you to make copies of any part of the Apple memory including the Apple ROM space. When you enter the option, the program will prompt you for the beginning and ending address of the memory you wish to copy. The addresses must be entered in hexadecimal. The maximum size of the working array is 8K, so the range of memory to be copied cannot exceed the array size. As an example:

START OF SOURCE E000
END OF SOURCE EFFF

In this example, the 2K block of memory from E000 to EFFF is copied into the working array. The copy always store the block starting at the bottom of the array.

In some instances, it is useful to copy data from one part of the working array to another; for instance, when you want to blast a large program into several small ROMs. Each segment can be copied to the bottom of the working array and blasted in turn.

To copy segments of the working array, just precede the address you specify with the letter "R". This tells the software that the address is "Relative" to the beginning of the working array. For example:

START OF SOURCE R1000
END OF SOURCE R1100

This example copies the data between 1000 and 1100 to the bottom of the array.

EXAMINE OR EDIT MEMORY

With this option, you can examine, change or enter data directly into the working array. When you enter the option, the program will prompt you for an address. This is the relative address from the beginning of the working array and must be in hexadecimal. After you have entered an address, the program will print eight bytes, starting with the address you entered:

```
0128 12 34 FF FA DD 76 90 99
```

If you hit "RETURN", the next 8 bytes will be printed. If you enter another address, 8 bytes starting with that address will be listed. If you wish to return to the main menu type Q.

If you wish to modify or enter bytes into the array, type "M". You will now be able to modify memory beginning with the last the address listed.

```
0128 12 34 FF FA DD 76 90 99
M
0128 12 34
0129 34 35
012A FF
012B FA
```

The program will display the address and the byte that you are about to modify. Typing a hex number will change the contents of that memory location to the number you have entered. Typing "RETURN" will pass to the next location without modifying the memory location. Typing minus (-) will backup one address without modifying the memory location. Typing "Q" will cause you to exit to examine mode. Here is a summary of commands:

EXAMINE MODE

Hex address ..... Examine this address
RETURN ..... Print next 8 bytes
Q ..... Return to main menu
M ..... Go to modify mode

MODIFY MODE

Hex byte ..... Enter this value
RETURN ..... Skip to next address
- ..... Back up one address
Q ..... Exit to examine mode

PRESET WHOLE ARRAY

This operation allows you to preset every byte of the working array to a value or a series of values. Enter each value followed by RETURN. When you have completed the string of values to be written type Q and a RETURN. Some examples:

```
FF
Q
```

```
12
34
56
78
9A
BB
CC
Q
```

The first example sets each byte of the array to FF. The second fills the array with the string of values shown.

This feature is usefull in two ways. First, you can blast a specific pattern of information into the ROM to test for pattern sensitive defective ROMs. Second, setting the array to all FF allows you to program small segments of 2704 or 2708. This is something most other blasters cannot do. More on this later in the manual.

SPECIAL KEYS

Several keyboard keys have special functions under the A.P.B. Here is a list the keys and their function.

CONTROL-C ..... Abort current operation and restart the program

CONTROL-S ..... Pauses the printing of text on the screen.

ANY KEY ..... Restarts the printing of text on the screen.

COPYING DISKS

You can make other working copies of the A.P.B. disk using either the Apple DOS disk copy or the APEX DUPDSK. Since Apple DOS is slower than the APEX operating system, copies made with Apple DOS will run a little slower than your original disk.

Once you have created a copy, you can delete all of the ROM files on the copy disk and have a completely fresh A.P.B. disk. You can also leave the ROM files intact, thus creating duplicates of all the ROM files.

COPYING FILES

ROM files can also be copies from one A.P.B. disk to another. To copy files, load the file you wish to copy into the working array. Then switch disks and save the working array. If you want to copy all of the files on disk, it is probably easier to do the full disk copy and leave the files in tact.

USING A.P.B UNDER APEX

(The following information only applies to those who are using the APEX operating system. If you using Apple DOS only, skip this section.)

If you have a copy of the APEX operating system you can use

the A.P.B. software under APEX. The A.P.B disk contain three important files:

SYSTEM.SYS
BLAST.SAV
ROM.DAT

SYSTEM.SYS normally contains the operating system on an APEX disk. On the A.P.B. disk SYSTEM.SYS contains a copy of the ROM blasting software. In this way, a user without APEX can boot the disk and automatically run the ROM software.

BLAST.SAV is a copy of the ROM software that is specially designed to run under APEX.

ROM.DAT is the master file containing the directory and data for all of the ROM data stored on disk. It must be 193 blocks long.

To use the A.P.B. software under APEX, boot the APEX system in the normal manner. Now copy BLAST.SAV to another APEX system disk. You may also copy ROM.DAT to the system disk at the same time. ROM.DAT can also be created directly using the make command:

```
MAKE ROM.DAT=193
```

If you create ROM.DAT this way, you will have to clear the A.P.B. file list because it will initially have garbage in it.

The APEX version of the A.P.B. has two extra commands under the "READ OR SAVE DATA" option:

8) WRITE BINARY
9) READ BINARY

These two commands allow you to read binary files from the APEX assembler.

After a program has been assembled on the APEX assembler, a binary file is generated. The file will have the extention ".BIN". The binary file can be loaded into the working array and written into the ROM by typing the following:

```
BLAST <SINETAB.BIN
```

This will start the "BLAST" program with the specified file as the input file. In this example, we have setup a file

SINETAB.BIN as an input file to the A.P.B. program. Now we can choose the "READ BINARY" option, and read the file into the working array.

WRITING THE PROGRAM

There are several things you will need to consider when you write an assembly language program to be written to ROM. The A.P.B. will take the first .LOC psuedo-op as the base address of the ROM. Any subsequent .LOC code will be taken as relative to the base address. For example:

```
        .LOC    $1000

START:  LDA#    $00
        JMP     FROG

        .LOC    $1200

MARK:   INY
        CLC
        RTS
```

In this example, $1000 is the base address, and this part of the program will be stored in the working array starting at address $0000. The second .LOC has an address of $1200, which is $200 greater than the base address, so that this part of the program is stored in the working array starting at address $200.

The working array can also be saved to disk in assembler binary form. The starting location counter will set at $8000 for consistency. To write a binary file, start from the APEX system and type:

```
BLAST FILENAM.BIN<
```

This will open the named file as an output file. Now select the "WRITE BINARY" command, and the file will be written to disk. This is a slow process, since each byte is converted to hex before it is written and the whole working array is being saved.

HARDWARE NOTES

The A.P.B. board has been designed to blast all of the common 24 pin EPROMs. Most of the pinouts on these ROMs 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 ROM 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 ROM types. By installing headers with appropriate jumpers, the proper logic levels can be connected to the proper ROM pins for each ROM type.

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

A11	Address bit 11, provides most significant bit for 4K ROMS
A10	Address bit 10, provides most significant bit for 2K ROMS
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.
-5 V	Minus 5 volt power supply line.
+12 V	Plus 12 volt power supply line.
+5 V	Plus 5 volt power supply line.

ROM ADDRESSING

The programming and reading addresses for the ROM are provided by a 12 bit binary counter. This takes the ROM off the Apple's address bus and allows the programmer to program large ROMs without interfering with the Apple's ROM space. The counter can reset, incremented and preset from software.

THE PIA

Interfacing of the A.P.B. card to the Apple bus is provided by a 6821 PIA. The PIA registers are addressed as follows:

N=SLOT NUMBER

Data Direction Register-A ..... $C0(N+8)0
Data Port-A ..... $C0(N+8)0
Control Register-A ..... $C0(N+8)1
Data Direction Register-B ..... $C0(N+8)2
Data Port-B ..... $C0(N+8)2
Control Register-B ..... $C0(N+8)3

Port-B of the PIA is connected directly to data bus of the ROM. Since the port can be programmed as either an input or an output, data can be either read or written to the ROM.

Port-A is used to control the address counter and the special personality module logic lines. Each bit of Port-A is set up as an output and controls the following functions:

Bit	7	6	5	4	3	2	1	0
	A	B	C	D	TTL	-	RES	CLK

The lines labeled A and B are used to control the PPH line.

A	B	PPH
0	0	GND
0	1	+5
1	0	+26
1	1	Tristate

Lines C and D control the PPL line:

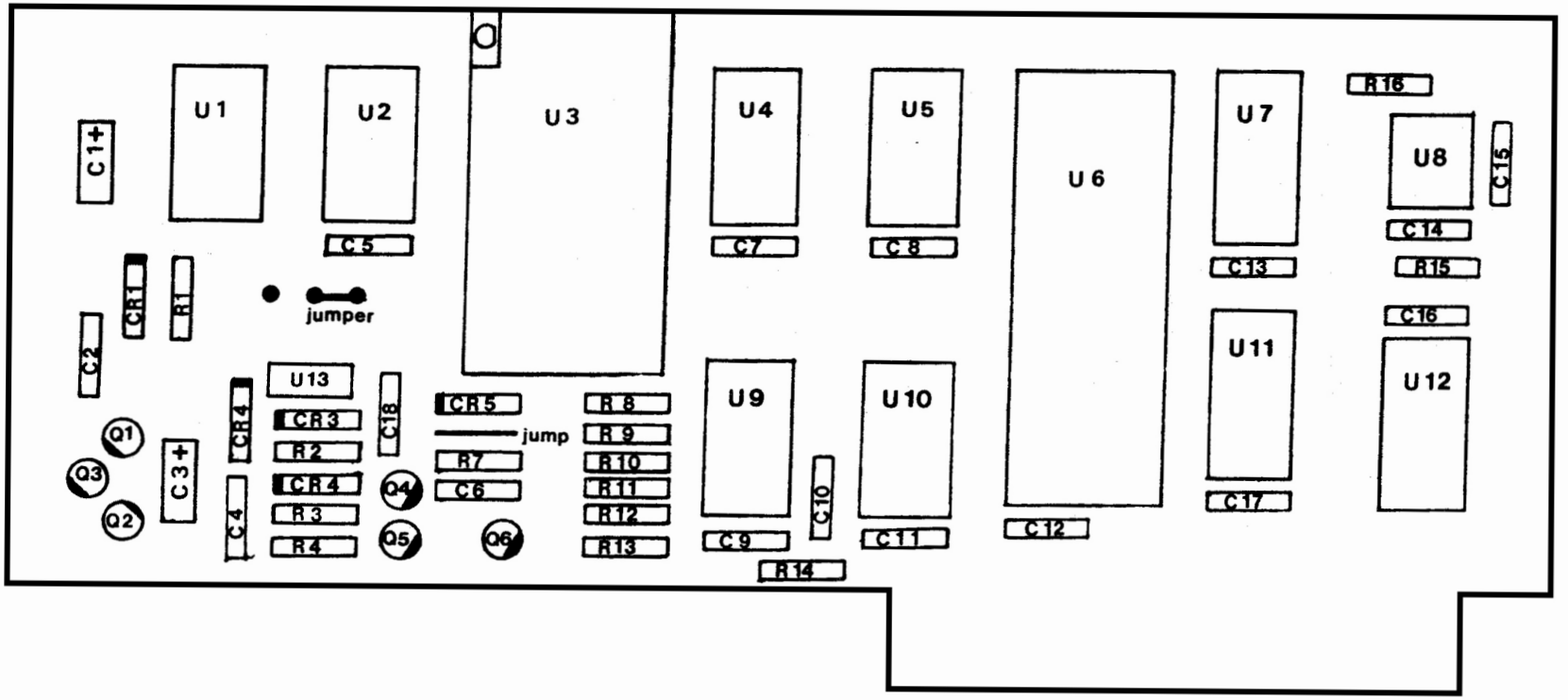
C	D	PPL
0	0	GND
0	1	GND
1	0	+12
1	1	+5

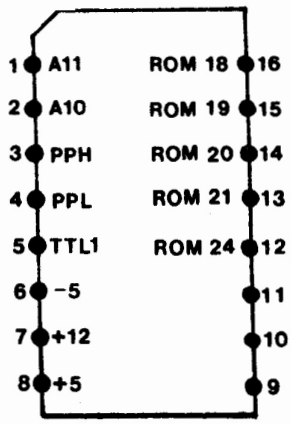
Bit-3 controls the TTL1 line. A high pulse on bit-1 resets the address counter. A high pulse on bit-0 increments the address counter.

The 26-volt programming voltage is provided by a charge pump which converts plus and minus 12-volts into +36-volts. The 36-volt supply is regulated down to 26-volts and fed to the pulse control logic.

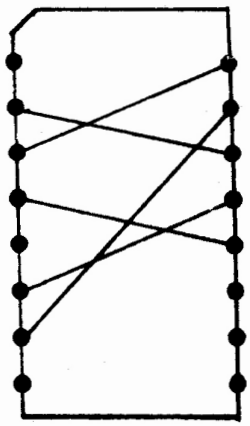
PARTS LIST

U1	Personality Select Socket
U2	4040
U3	PROM Socket (ZIF)
U4	74LS02
U5	74LS04
U6	6820 or 6821 or 6521
U7,U11,U12	74LS367
U8	555
U9	7407
U10	74LS00
U13	7824
C2,C5,C6,C7, C8,C9,C11,C12, C13,C14,C16, C17,C18	.1 Mfd.
C1,C3	3.3 Mfd.
C4	.06 Mfd.
C10	.22 Mfd.
C15	.001 Mfd.
R1,R15	1K 1/4 Watt
R2,R12,R13	10K 1/4 Watt
R3	10 Ohm 1/4 Watt
R4	50 Ohm 1/4 Watt
R7	390 Ohm 1/4 Watt
R8	6.8 K 1/4 Watt
R9	2.2 K 1/4 Watt
R10,R11	2.7 K 1/4 Watt
R14	1.5 Meg. 1/4 Watt
R16	47 K 1/4 Watt
CR1,CR2,CR4	1N4001
CR3	3.2 Volt Zener
CR5	5 Volt Zener
Q1,Q2,Q3,Q6	2N2222 NPN
Q4,Q5	2N2907 PNP

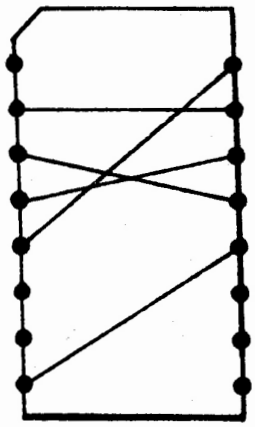

U1
U2
U3
U4
U5
U6
U7
U8
U9
U10
U11
U12
U13
C1+
C2
C3+
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
C14
C15
C16
C17
C18
CR1
CR3
CR4
CR5
R1
R2
R3
R4
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16
Q1
Q2
Q3
Q4
Q5
Q6
jumper
jump



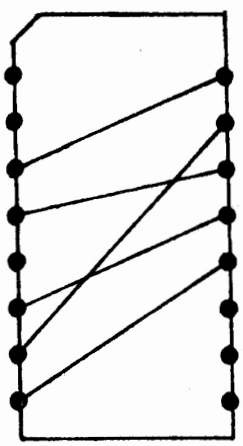
SIGNALS



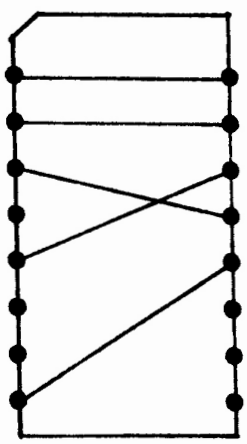
2716
3 Voltage



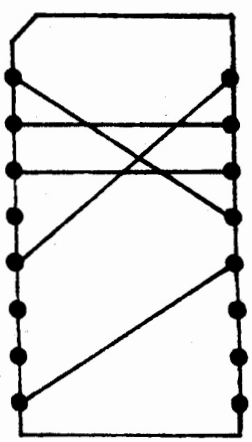
2716
5 Volt Only



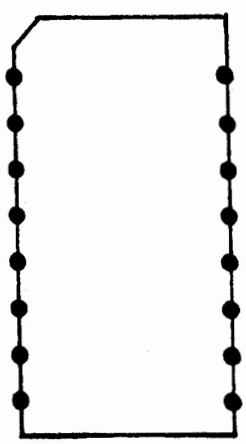
2708

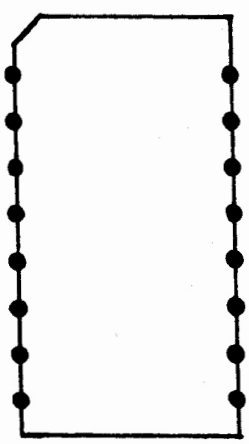


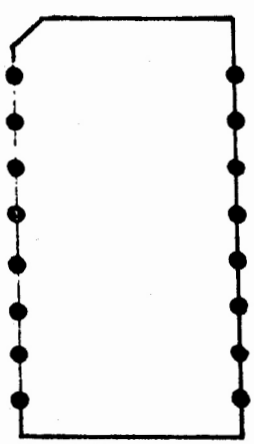
2532

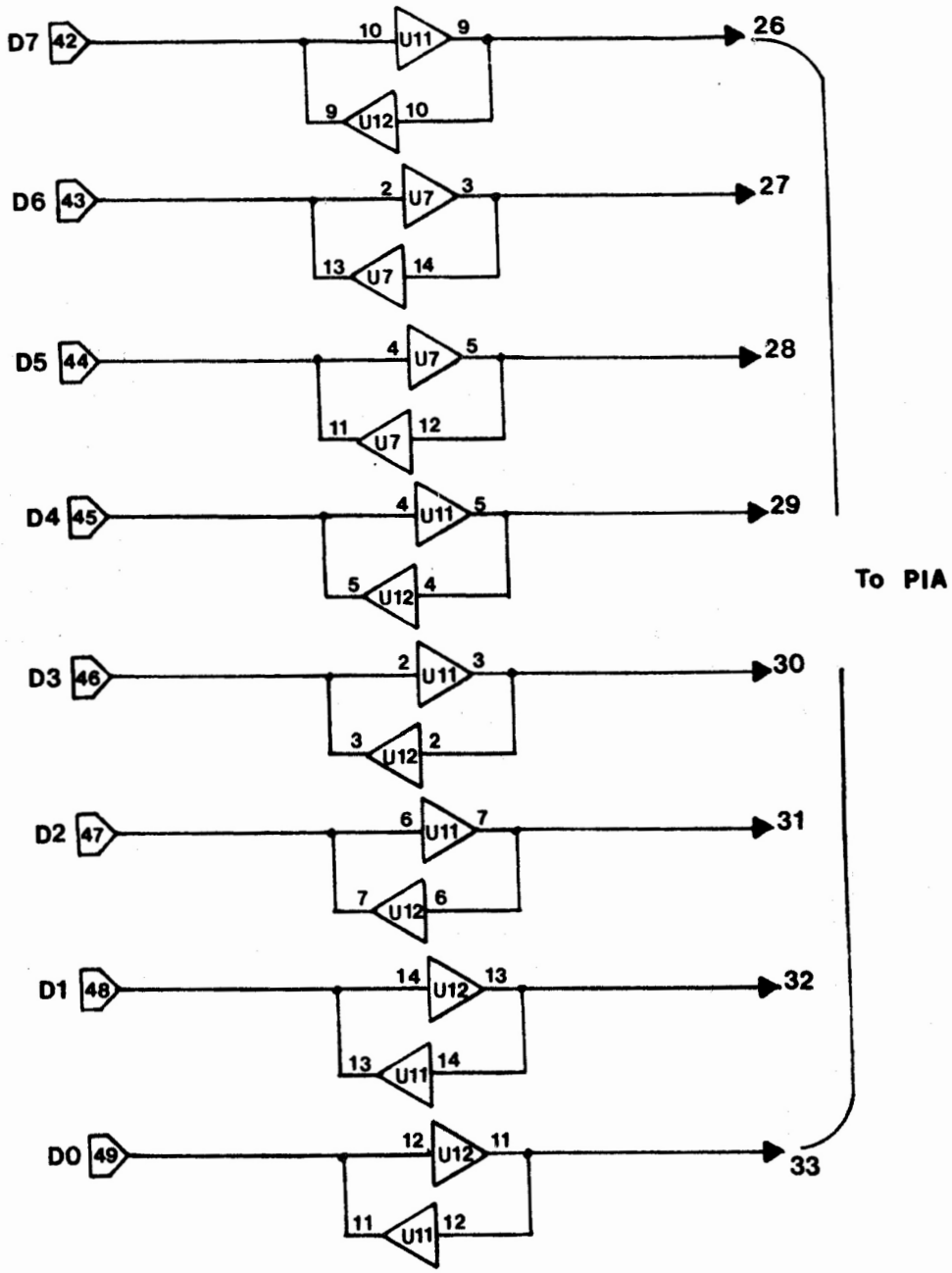


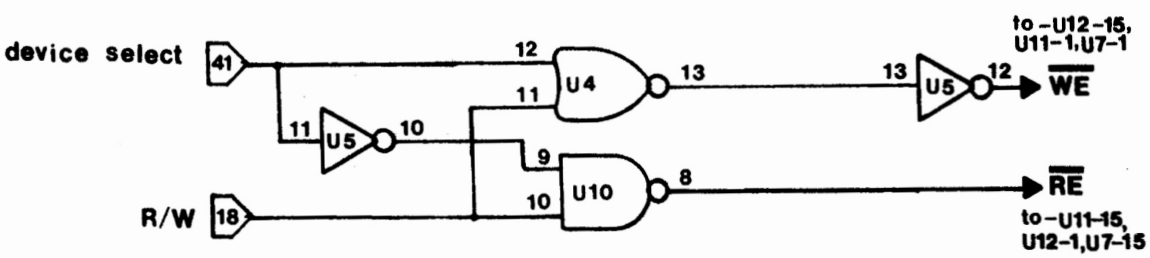
2732

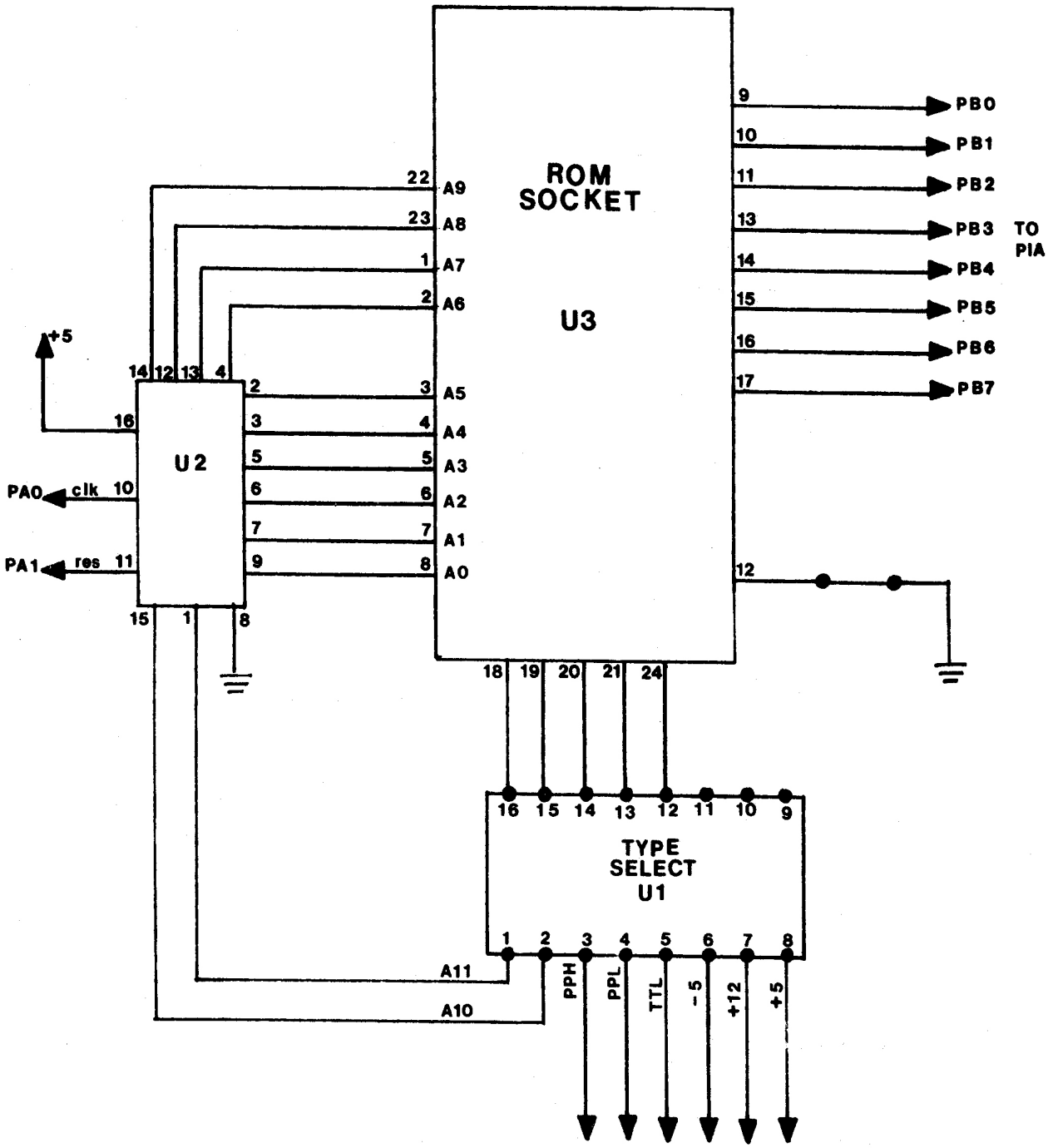


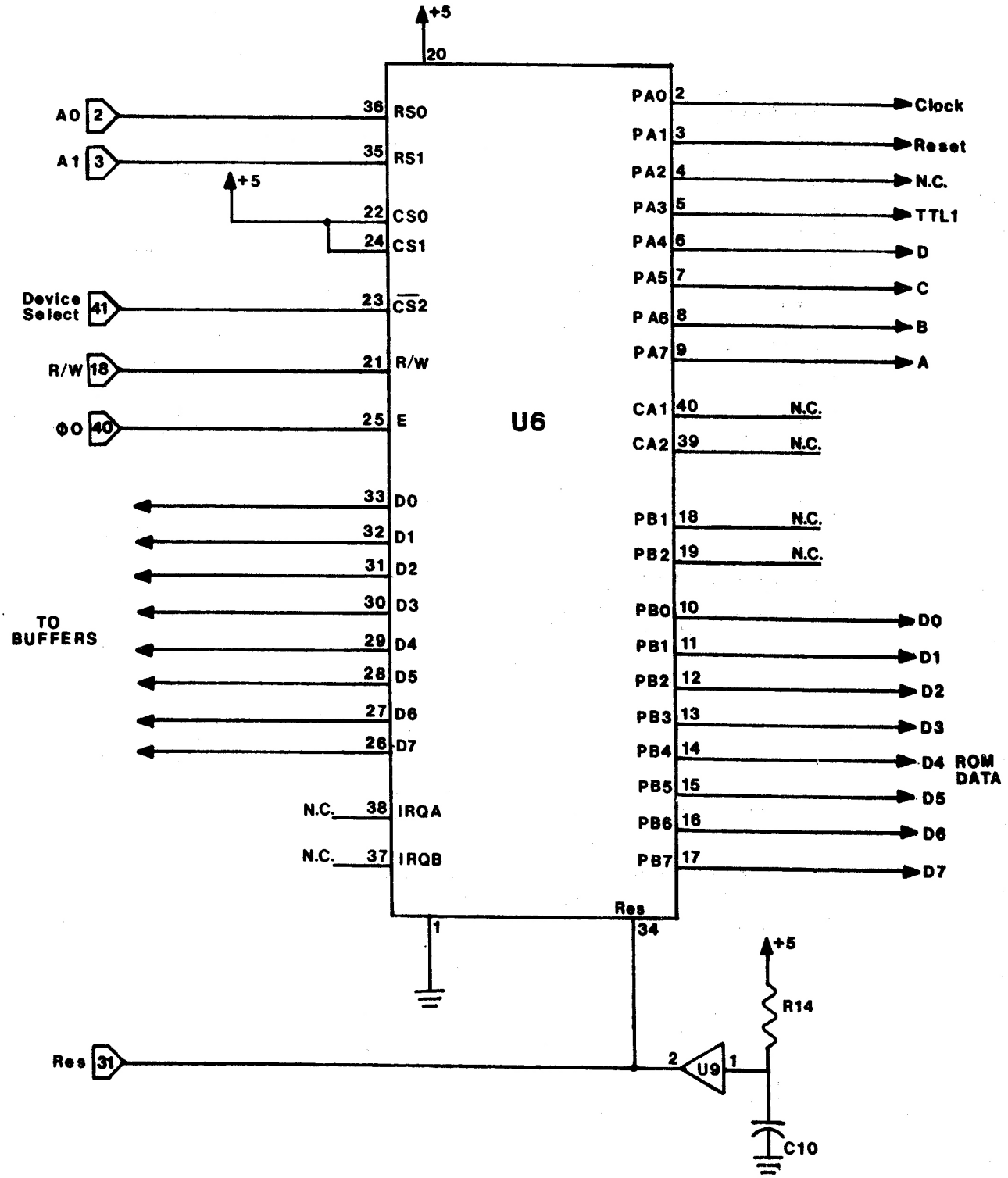


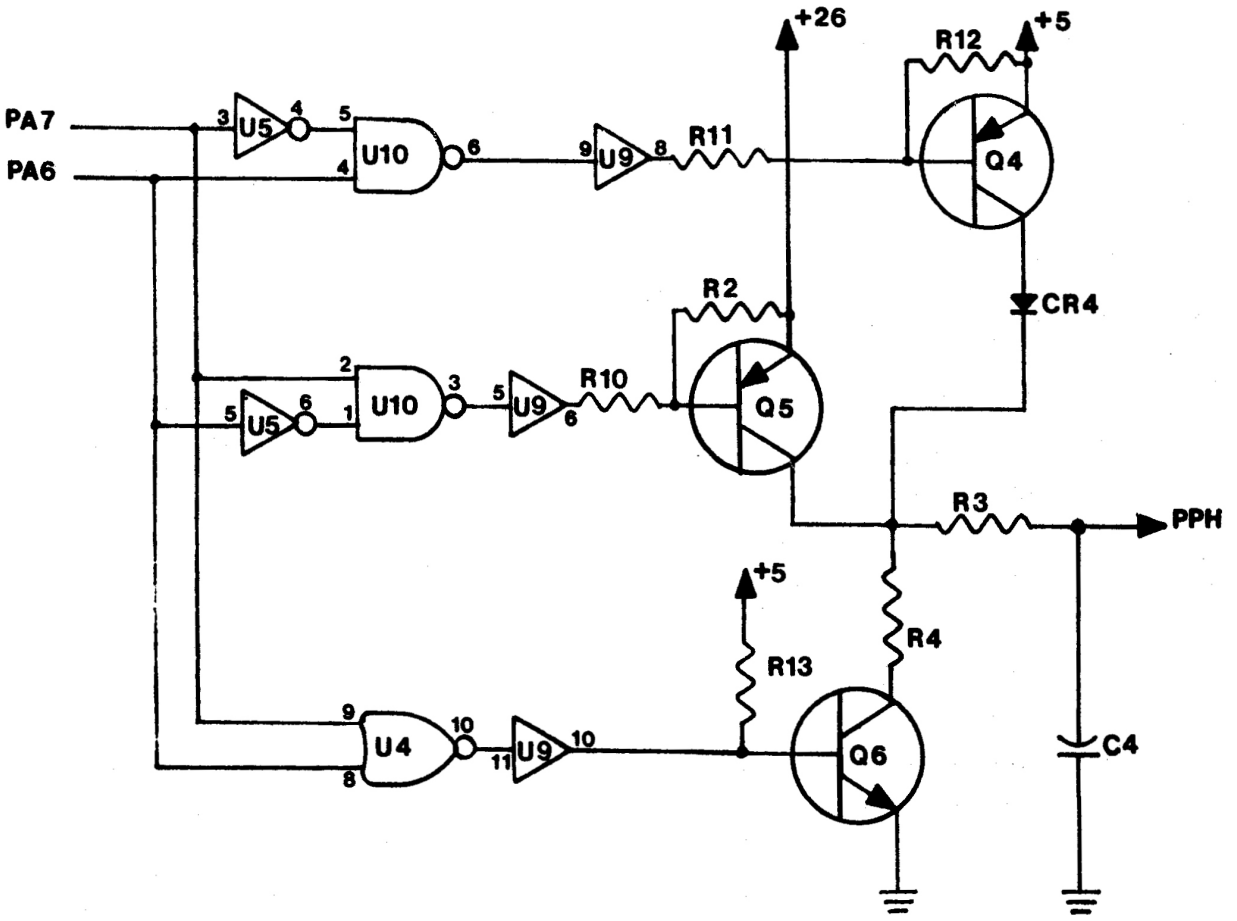



D7
42
10
U11
9
26
9
U12
10
D6
43
2
U7
3
27
13
U7
14
D5
44
4
U7
5
28
11
U7
12
D4
45
4
U11
5
29
5
U12
4
To PIA
D3
46
2
U11
3
30
3
U12
2
D2
47
6
U11
7
31
7
U12
6
D1
48
14
U12
13
32
13
U11
14
D0
49
12
U12
11
33
11
U11
12

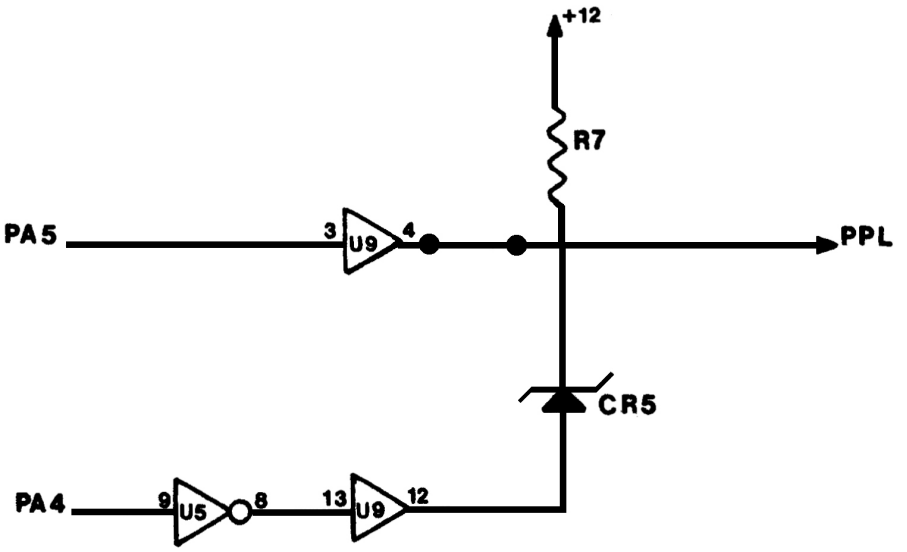

device select
41
12
U4
13
13
U5
12
to -U12-15,
U11-1,U7-1
WE
11
U5
10
11
9
U10
8
RE
R/W
18
10
to -U11-15,
U12-1,U7-15


ROM
SOCKET
U3
U2
+5
PA0
clk
PA1
res
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
TO
PIA
TYPE
SELECT
U1
A11
A10
PPH
PPL
TTL
-5
+12
+5

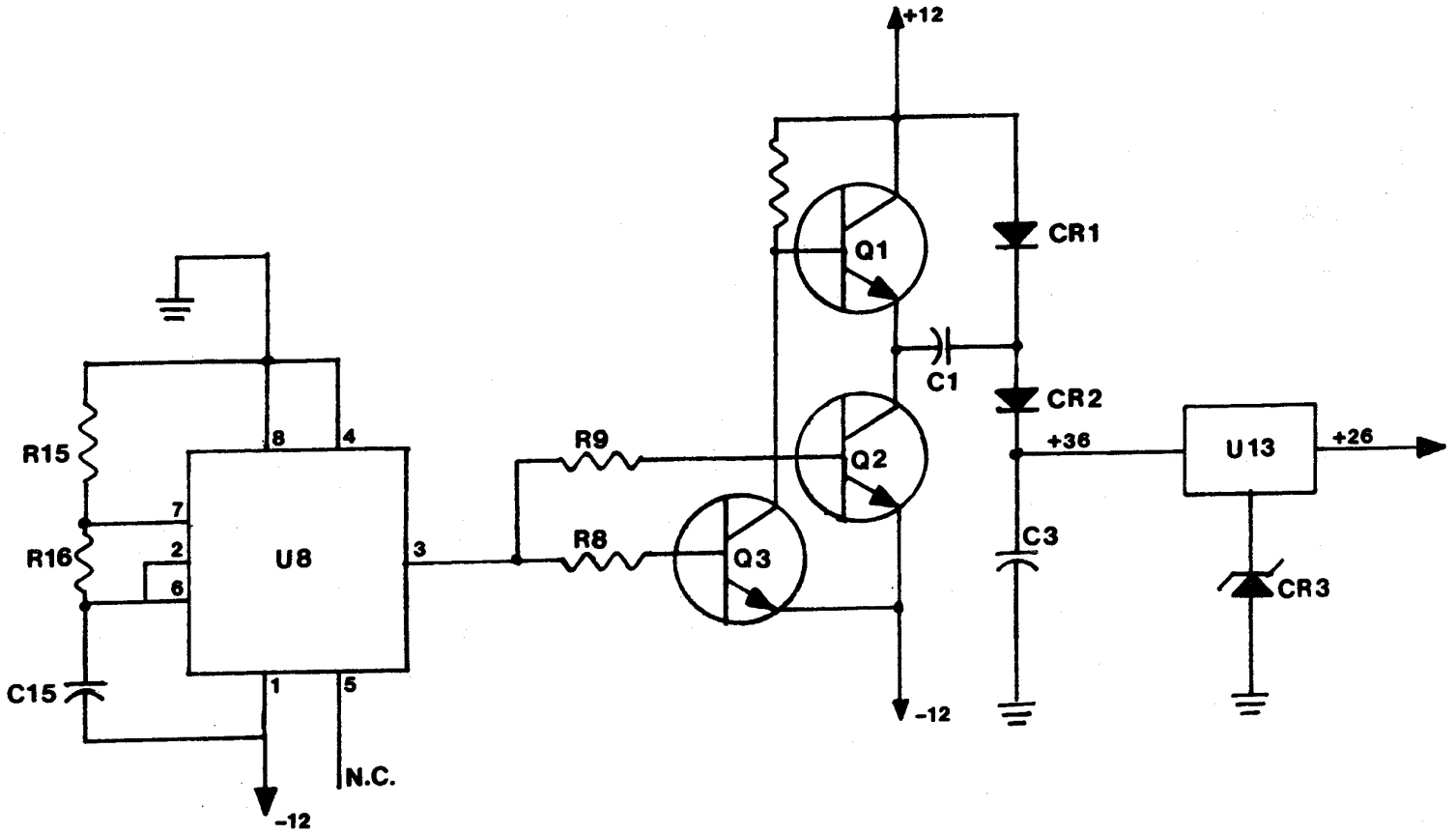

+5
20
A0
2
36
RS0
A1
3
35
RS1
+5
22
CS0
24
CS1
Device Select
41
23
CS2
R/W
18
21
R/W
ΦO
40
25
E
U6
33
D0
32
D1
31
D2
30
D3
TO BUFFERS
29
D4
28
D5
27
D6
26
D7
N.C.
38
IRQA
N.C.
37
IRQB
1
Res
34
PA0
2
Clock
PA1
3
Reset
PA2
4
N.C.
PA3
5
TTL1
PA4
6
D
PA5
7
C
PA6
8
B
PA7
9
A
CA1
40
N.C.
CA2
39
N.C.
PB1
18
N.C.
PB2
19
N.C.
PB0
10
D0
PB1
11
D1
PB2
12
D2
PB3
13
D3
PB4
14
D4
ROM DATA
PB5
15
D5
PB6
16
D6
PB7
17
D7
+5
R14
Res
31
2
U9
1
C10



PA-7	PA-6	PPH
O	O	GND
O	I	+5
I	O	+26
I	I	TRI



PA5	PA4	PPL
0	0	0
0	1	0
1	0	12v
1	1	5v


+12
CR1
Q1
C1
CR2
+36
U13
+26
R9
Q2
R8
Q3
C3
CR3
-12
R15
R16
C15
U8
8
4
7
2
6
3
1
5
N.C.
-12