

## Appendix H— Hardware Description for PROM Programmer Module

Fig. H1 shows a block diagram of the programmer module. It consists of two output ports that latch and hold addresses for the PROM's, an input and output port connected to the PROM data bus, and two more output ports that supply control signals such as chip select and programming levels to the PROM's. All internal signals operate at 5 volts as is common to the MCDS. Discrete transistor drivers operated by control signals are used to switch higher voltages to the PROM's.

### Group-Select Logic

Fig. H2 gives a block diagram of the group-selection logic on the PROM Programmer module. The execution of output instruction '61' generates a clock

input to the flip-flop. The D-input of that flip-flop is connected by a link to one of nineteen possible signals. The high-order four bits of the data bus are decoded into fifteen lines (the decode of all zero is not used); the low-order four bits are supplied directly. Thus, a total of nineteen unique addresses is available to be assigned to the PROM Programmer Module. The Module comes with the link installed in position C, connecting data bus bit DB2 as the enabling signal. The Module is thus enabled by output instruction '61' with data byte=00000100.

### Power Supply Control

The supply voltages to the PROM's are sent through the Power ON/OFF switch S2 and the software sets

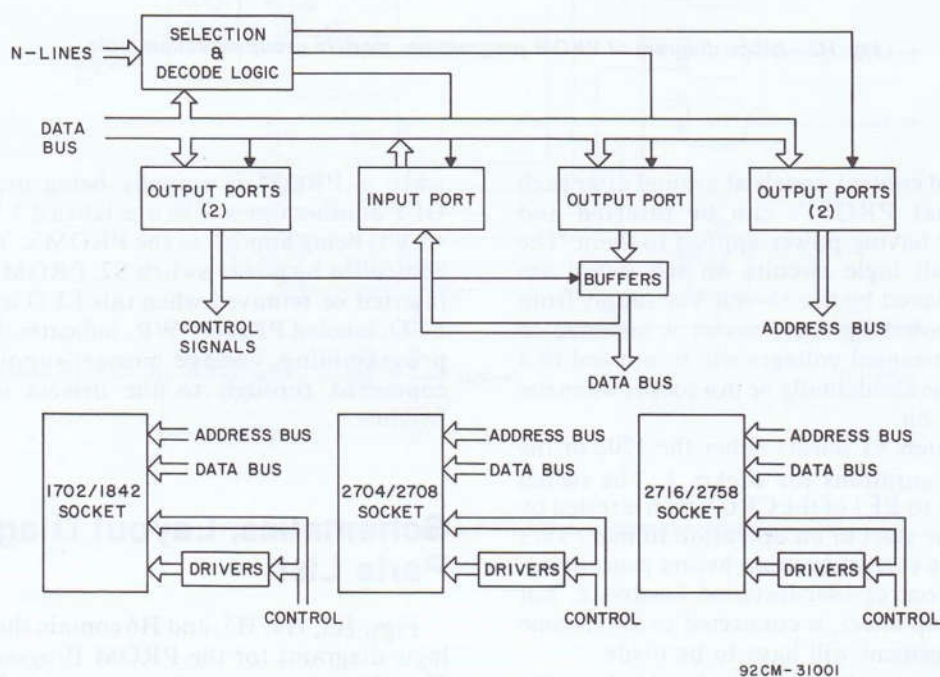


Fig. H1—Block diagram of PROM programmer module.

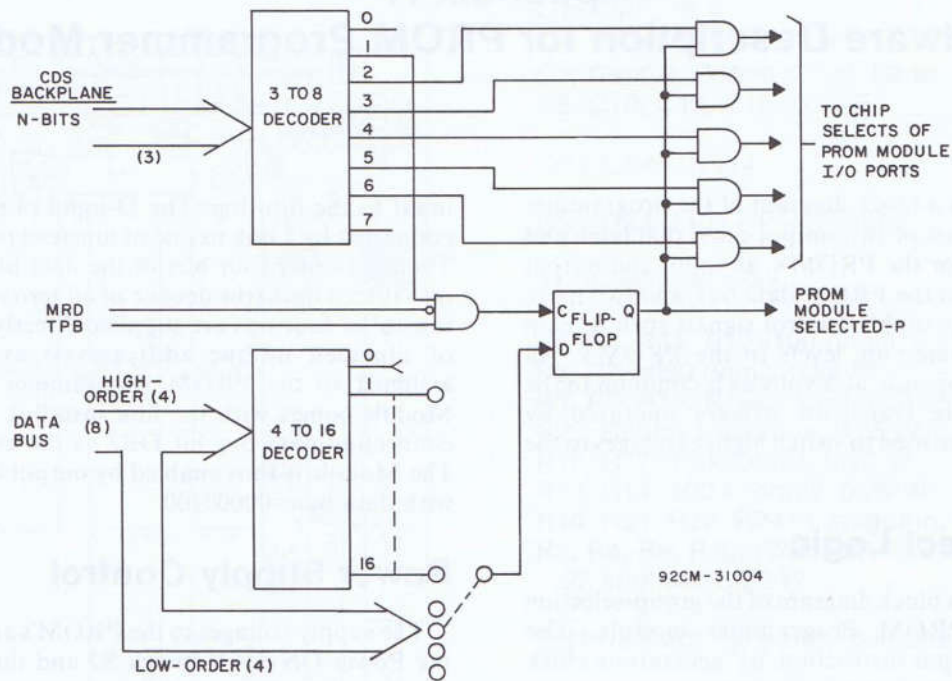


Fig. H2—Block diagram of PROM programmer module group-selection logic.

address, data, and control signals at ground after each operation so that PROM's can be inserted and removed without having power applied to them. The I/O ports and all logic circuits on the board are continuously powered by the +5-volt Vcc supply from the MCDS. A power-up reset circuit is included to assure that no unwanted voltages will be applied to a PROM should one accidentally be in a socket when the MCDS is turned on.

A selection switch S1 selects either the 1702 or the CDP18U42 configurations for socket 1. The switch generates a signal to EF1 of the CPU which is tested by the software at the start of an operation to make sure the switch is in the correct position before proceeding. If some other piece of user-installed hardware, not covered by a group select, is connected to EF1, some alternative arrangement will have to be made.

Three LED status indicators are provided on the edge of the Module. The one labeled PROG will be ON

when a PROM is actually being programmed and OFF at other times. The one labeled 5 V indicates that +5 V is being applied to the PROM's. This indicator is controlled by power switch S2. PROM's should not be inserted or removed when this LED is ON. The third LED, labeled PROG PWR, indicates that the external programming voltage power supply is ON and connected through to the drivers circuits on the Module.

## Schematics, Layout Diagram, and Parts List

Figs. H3, H4, H5, and H6 contain the schematic and logic diagrams for the PROM Programmer Module. Fig. H7 is the layout diagram and is followed by the parts list.

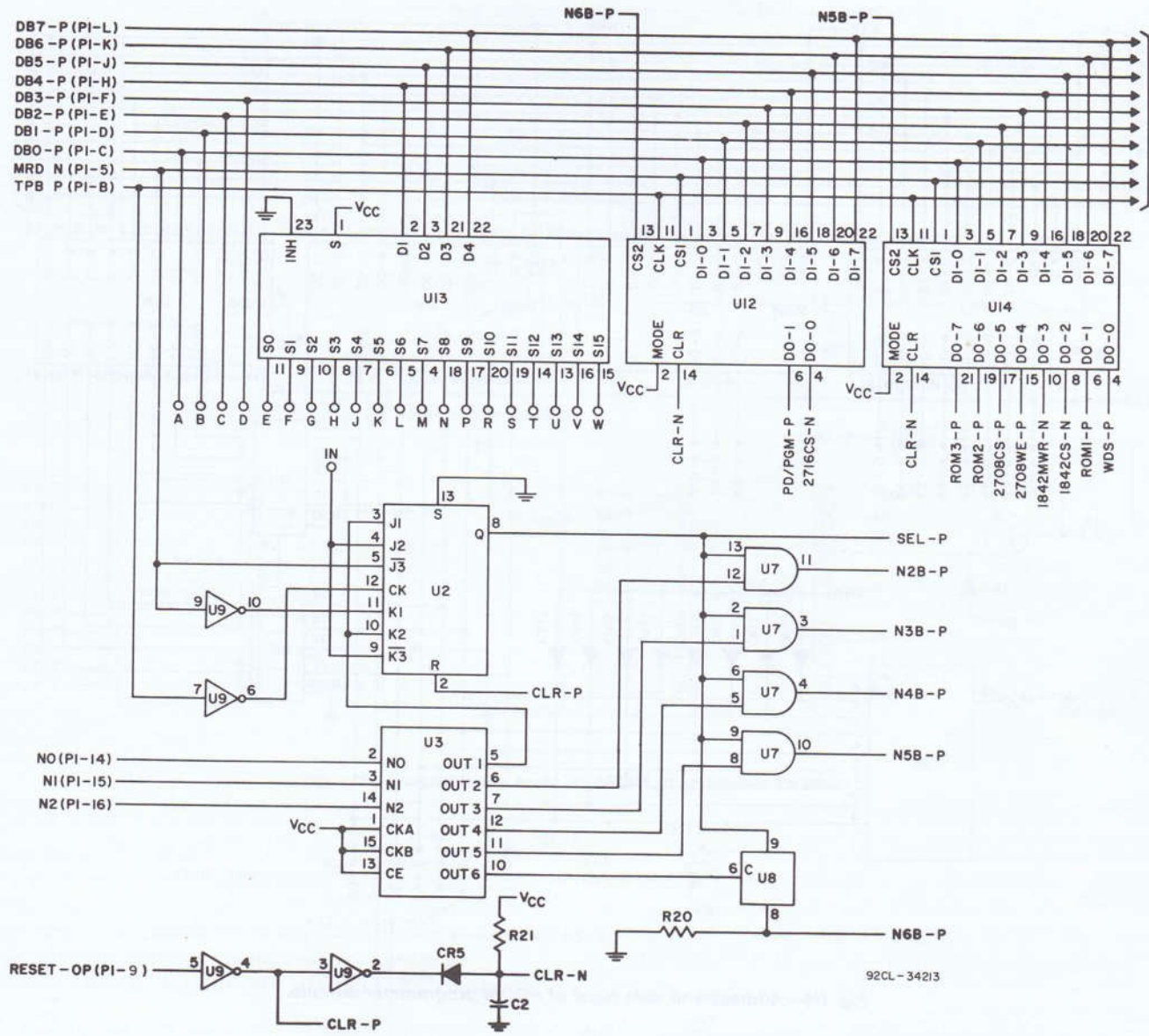


Fig. H3—Control logic of PROM programmer module.

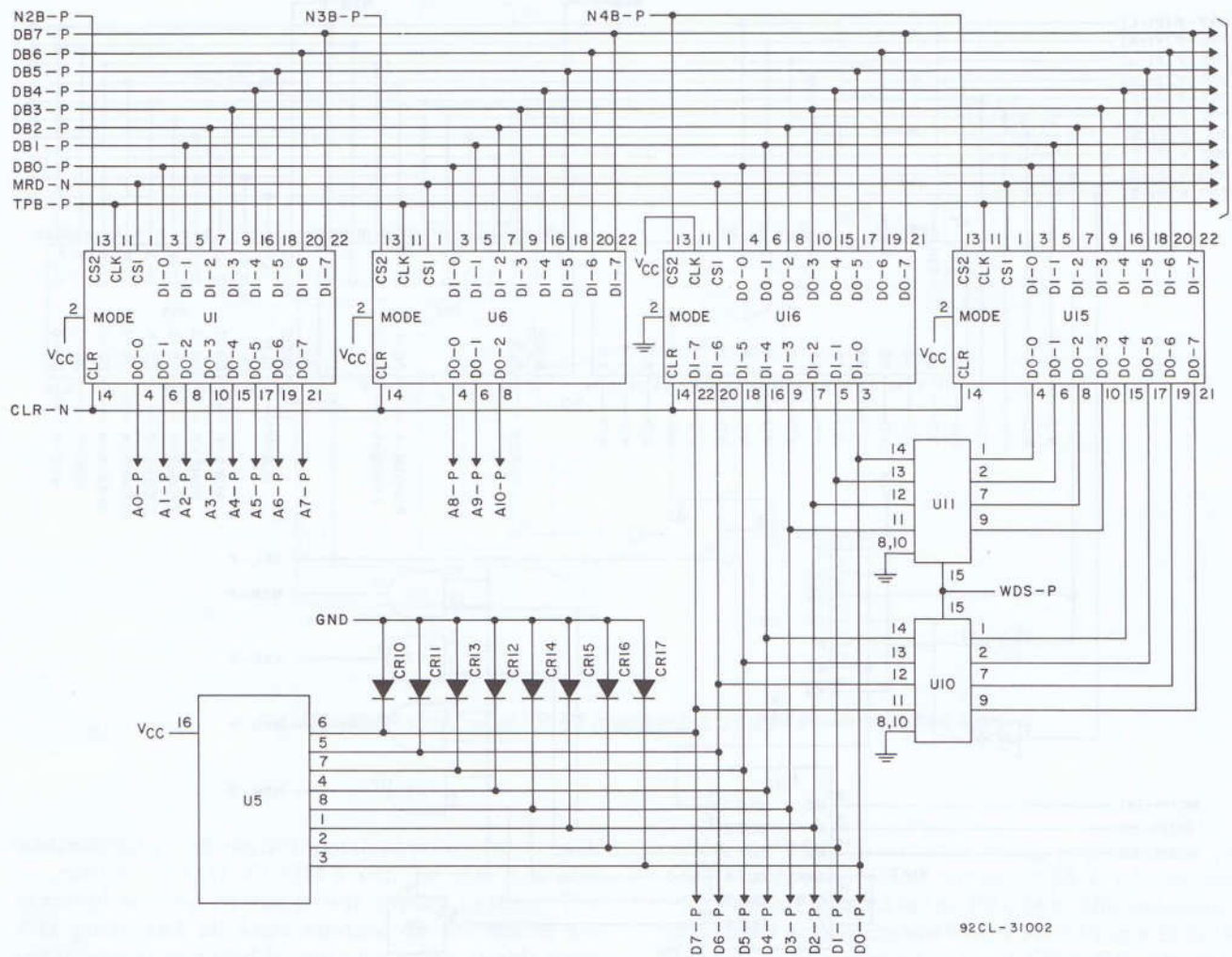


Fig. H4—Address and data logic of PROM programmer module.

92CL-31002

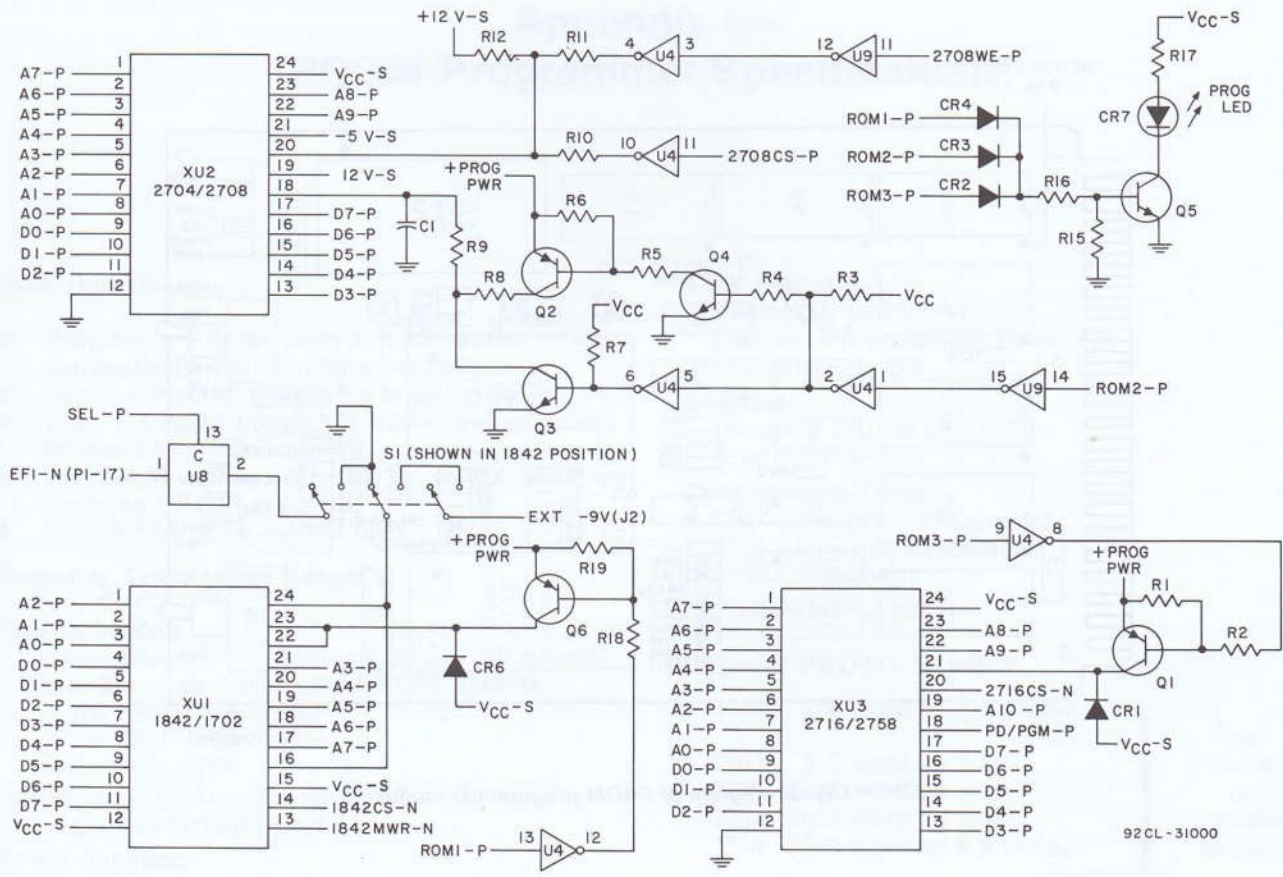


Fig. H5—Programming logic of PROM programmer module.

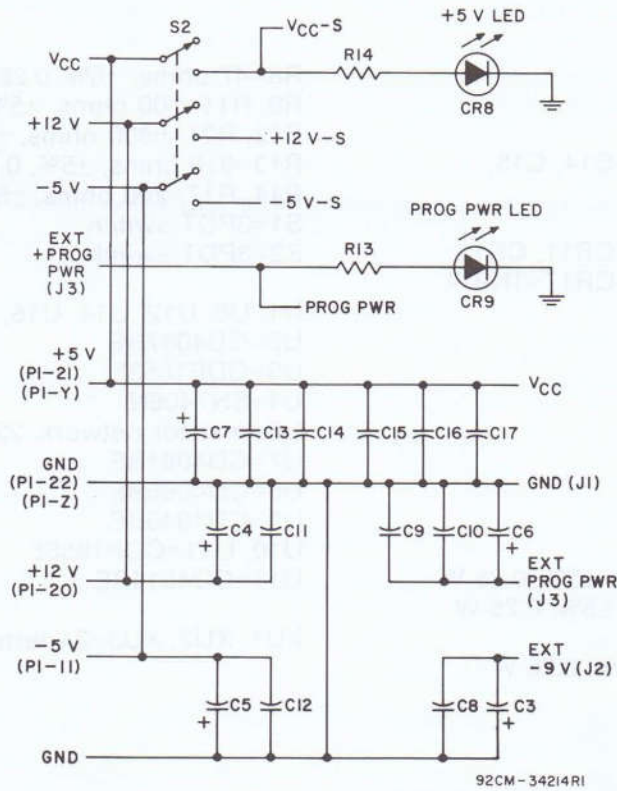


Fig. H6—Power supply of PROM programmer module.

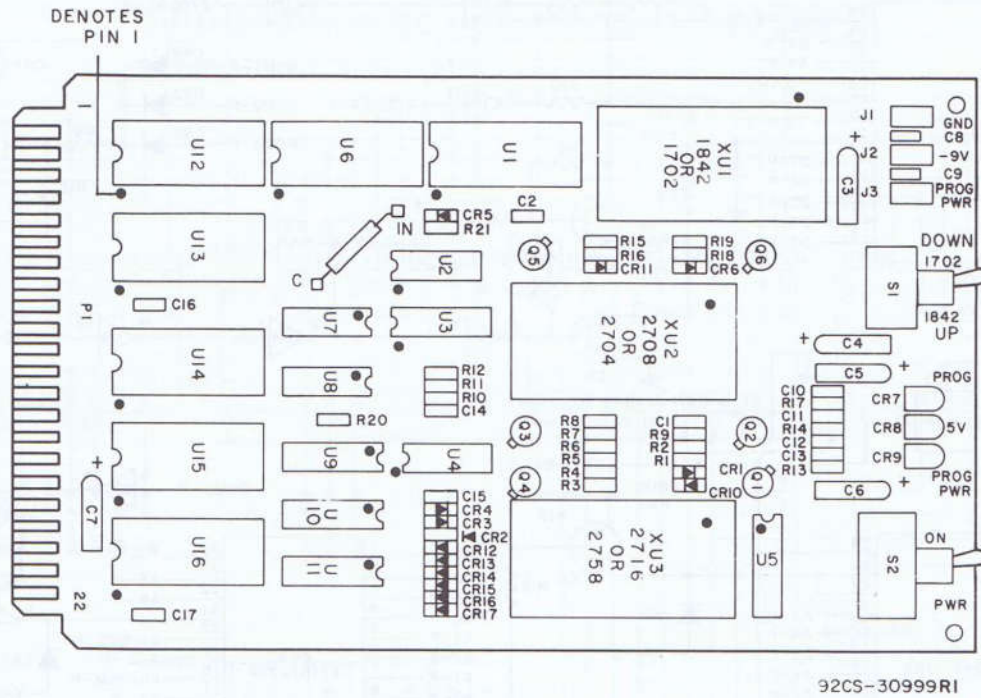


Fig. H7—Layout diagram of PROM programmer module.

Parts List

C1, C2=0.0022  $\mu$ F  
 C3, C4, C5, C7=15  $\mu$ F, 20 V  
 C6=15  $\mu$ F, 50 V  
 C8, C9, C10, C11, C12, C13, C14, C15,  
 C16, C17=0.1  $\mu$ F, 50 V  
 CR1, CR6=1N34A  
 CR2, CR3, CR4, CR5, CR10, CR11, CR12  
 CR13, CR14, CR15, CR16, CR17=1N914  
 CR7, CR8, CR9=LED red

J1=jack, black  
 J2=jack, green  
 J3=jack, yellow

Q1, Q2, Q6=2N2907  
 Q3, Q4, Q5=2N2222

R1, R6, R19, R20=10 kilohms,  $\pm$ 5%, 0.25 W  
 R2, R4, R5, R18=3000 ohms,  $\pm$ 5%, 0.25 W  
 R3=1000 ohms,  $\pm$ 5%, 0.25 W  
 R7, R10, R16=4700 ohms,  $\pm$ 5%, 0.25 W

R8=47 ohms,  $\pm$ 5%, 0.25 W  
 R9, R11=100 ohms,  $\pm$ 5%, 0.25 W  
 R12, R21=6800 ohms,  $\pm$ 5%, 0.25 W  
 R13=910 ohms,  $\pm$ 5%, 0.25 W  
 R14, R17=200 ohms,  $\pm$ 5%, 0.25 W  
 S1=3PDT switch  
 S2=3PDT switch

U1, U6, U12, U14, U15, U16=CDP1852E  
 U2=CD4097BE  
 U3=CDP1853E  
 U4=SN7406N  
 U5=resistor network, 22 kilohms  
 U7=CD4081BE  
 U8=CD4066BE  
 U9=CD4049BE  
 U10, U11=CDP1856E  
 U13=CD4514BE

XU1, XU2, XU3=24-terminal socket

## Appendix I— PROM Programmer Specifications

### Basic Operations:

- Program a PROM from a RAM buffer or file; automatically followed by a verification
- Verify a PROM against RAM buffer or file
- Copy a PROM into RAM buffer, automatically followed by a verification
- Fill RAM buffer with all 1's or 0's; used in verifying PROM erasure
- Save RAM buffer onto a file

### Operating Temperature Range:

0° to 43°C

### Plug-In Module

Dimensions: 4.5 × 7.5 inches (114.3 × 190.5 mm)

Three Zero-Insertion Force PROM Sockets

1 for 1702/CDP18U42

1 for 2704/2708

1 for 2716/2758

Plugs into Microboard chassis

Assigned to Group Select 4

### Power Supplies:

#### External-Programming Power:

See Manufacturer's data specifications sheet

### LED Indications:

Power ON to PROM

External Programming Power ON

Programming ON

### Switches:

Power to PROM ON/OFF

Selector Switch

### Programming Times:

2708 - 2 minutes 45 seconds

2716 - 1 minute 45 seconds

2758 - 50 seconds

CDP18U42 - 3 seconds

### Types of PROM's Handled:

1702 256 word by 8 bit-read  
only

2704 512 word by 8 bit

2708 1024 word by 8 bit

2758 1024 word by 8 bit

2716 2048 word by 8 bit-single  
voltage  
only

} Intel  
PROM's  
or  
equivalent

CDP18U42 256 word by 8 bit

## 10. Using the PROM Programmer

The PROM Programmer is a hardware and software package that facilitates user programming of RCA and other industry-standard PROM's. Specifically, the RCA CDP18U42, the Intel 2708, 2758, and 2716 UV-erasable PROM's, or equivalent PROM's can be programmed with this system. In addition, Intel 1702-type PROM's can be read (but not programmed) and copied into lower-power CDP18U42 CMOS PROM's or combined into larger-size PROM's. The PROM Programmer is included with the CDP-18S694V1 and V3 systems and may be added to the CDP18S693V1 and V3 systems by means of the PROM Programmer package CDP18S680.

The PROM Programmer package includes a plug-in Microboard module and a magnetic tape cassette containing the operating software. The software is designed for flexibility so that, in addition to the numerous basic operations provided, more sophisticated procedures can be derived from these building blocks. For example, a PROM can be copied from another PROM or programmed from a tape-resident file. The user can also combine two smaller PROM's into a larger one (two 2708's copied into one 2716). PROM's are automatically verified after programming. The user, however, can manually verify the erasure of a PROM, verify one PROM against another, or verify a PROM against a tape-resident file. These operations can be done with either positive (non-inverted) or negative (inverted) data.

An additional feature allows PROM's to be saved on a tape file (in a reloadable format) that can be used to generate masked ROM's. For PROM production purposes, a software feature facilitates rapid programming of many PROM's from the same source.

### Installation

The PROM Programmer Module, as provided with the CDP18S694V1 and V3, is installed in the bottom slot of the chassis. When the module is to be used for programming PROM's, the user should remove the top cover and exchange the PROM Programmer

module for the Microboard module in the top slot. By having the PROM programmer module in the top slot, access is provided to the three PROM programming sockets.

An external power supply is required as a programming voltage to be connected to jack J3, labelled PROG PWR. The voltage required for each PROM is given in the PROM manufacturer's data specification sheet. The supply must be well regulated and capable of supplying peak currents of 100 milliamperes. A -9-volt supply connected to J2 is also necessary when the programmer is used with a 1702 PROM. J1 is a ground connection. In addition to the external power supplies (Programming Power and -9 volts), voltages of +12 volts and -5 volts (both  $\pm 5\%$ ) must be supplied to the Microboard Universal Backplane pins 20 and 11, respectively, when 1702 or 2708 devices are to be operated on. Detailed specifications for the power supply requirements are given in Appendix I.

The PROM programmer module is assigned to Group Select 4. The module generates a signal on flag line EF1 of the CPU indicating whether the 1702/1842 switch is in the correct position.

### Memory Organization and Use

The memory map for the PROM Programmer software is given in Fig. 17. The program loads starting at address 0000 and occupies the lowest two kilobytes of RAM. The hexadecimal addresses from 0800 to 0FFF comprise a data buffer area used to hold data moved to and from the PROM being operated on. The buffer memory is treated in pages, where a page is defined as 256 bytes. Thus, the first page of buffer memory occupies the hexadecimal addresses 0800 through 08FF and is called page 8; the second page, 0900 through 09FF, is called page 9; etc. The data of a 1702 or CDP18U42 occupy one page, a 2708 or 2758 four pages, and a 2716 eight pages.

The program asks for the starting page in the data buffer against which operations are to be performed.



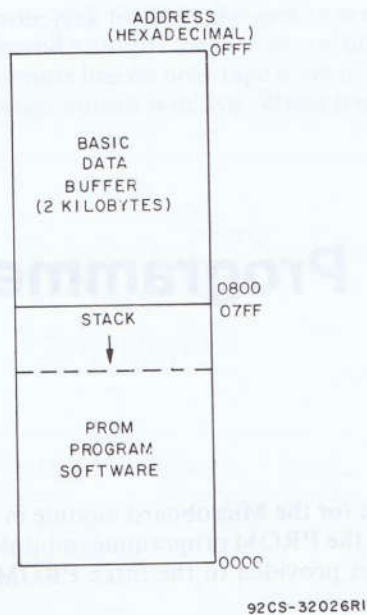


Fig. 17—Memory map for PROM Programmer software.

The response must be a one- or two-digit hexadecimal number equal to or greater than eight. Because a leading zero is assumed, only a single digit has to be entered for pages 8 through F. A number less than eight is not accepted by the program, thereby preventing accidental overwrite of some areas. Additional RAM can be used for additional page buffer areas.

The user can copy a PROM into any page ( $\geq 8$ ), program a PROM from any page, or verify a PROM against any page. He does not necessarily have to start at page 8 and can take advantage of additional RAM if the MCDS is equipped with more than 4 kilobytes. However, when an object file is loaded into the buffer area, loading starts at address 0800. Likewise, when the buffer area is saved on a disk, the SAVEing begins at address 0800. Refer to the Section Operations and Commands for details on the operation of the object file reader and SAVE programs.

It is relatively easy to copy two smaller PROM's into one larger one. For example, to copy two 2758's into one 2716, the following steps would be followed.

- (1) Copy the first 2758 starting at page 8. It will occupy two pages, addresses 0800 through 08FF.
- (2) Copy the second 2758 starting at page A. It will occupy addresses 0G00 through 0FFF.
- (3) Then, program a 2716 starting at page 8.

Reversing the procedure would permit programming several smaller PROM's from a larger one. Some care must be exercised in keeping track of where data is loaded in buffer memory for these combined operations. The software will not prevent overwriting of data in the buffer area because that may be the desired operation.

When programming takes place, data is taken from the starting page specified for the appropriate number

of bytes. Thus, the data to be programmed must be contiguous in the buffer area.

## Software Considerations

### General Procedure

The PROM Programmer software is designed to prompt each step of a procedure so that operation of the system is self-explanatory and, it is hoped, relatively error-proof. For each type of operation (except SAVE) the user is asked to identify the type of PROM being used and whether positive (noninverted) or negative (inverted) data is to be used. When the program is started, and after completion of each operation, the following message is typed, asking what operation is next.

F, V, P, C, S, J, U?

The single letters refer to the following operations. The user should enter the appropriate letter.

- F—Fill selected pages of the RAM buffer with 1's (positive logic) or 0's (negative logic)
- V—Verify a PROM against the RAM buffer or a file
- P—Program a PROM from the RAM buffer or a file
- C—Copy a PROM into the RAM buffer
- S—Save the RAM buffer on a file
- J—Rewind the tape
- U—Exit to the monitor

For all except the SAVE operation, the PROM Programmer asks for the PROM type by the message

TYPE#?

The user should enter one of the following:

1702, 1842, 2708, 2758, or 2716

followed by a carriage return (CR).

The next question asked is whether the operation is to be performed with positive logic (non-inverted data) or negative logic (inverted data).

LOGIC =P, N?

The user replies by typing a P for positive logic or an N for negative logic.

These three prompts are common for the V, C, P, and F commands. All operations also have in common the manipulation of data in the RAM buffer area previously described. Data to be programmed goes to a PROM from the buffer area. Likewise, data copied from a PROM goes into the buffer, etc. Succeeding questions ferret out the area of RAM to be used.

When an operation is completed, the program types

DONE

In the case of a verify (V) or programming (D) sequence, it asks if the user wants to repeat the operation with a question such as

REPROGRAM 1842?

The user responds with a "Y" for yes or an "N" for no. Because a "yes" response starts another programming

cycle immediately, the new PROM should be in place before an affirmative response is given. A "no" response restarts the program.

The Section **Operations and Commands** gives details and examples of each operation.

### Error Messages

Whenever the program receives an inappropriate response to a question, it types a "?" and repeats the question. It continues this reaction to each incorrect response until an acceptable one is received. Then, it will proceed.

The "?" is also generated if a number less than 8 is entered in response to the PAGE? question or if the user asks to program a 1702 (a non-existent function).

The following error messages may also be generated when a tape-read operation is being performed:

**NO OBJECT CODE**

or

**FORMAT ERROR**

or

**LOAD ABORTED**

The last message follows a non-recoverable error message.

### Aborting Operations

When a PROM is being verified, a list of errors is generated showing any mismatches between the PROM data and what is in the RAM buffer. A long timeout can be halted by pressing the BREAK key. Because the BREAK key is sampled at the end of each typed line, it must be held down long enough to be "seen" by the program. The message

**\*\*\*ABORTED\*\*\***

is typed and, when the BREAK key is released, the program restarts from the beginning.

A programming operation can be aborted by pressing and holding the BREAK key. Again, the message is typed and the program restarts. Response to this key may take a few seconds.

### Correcting Typing Errors

All numbers entered in response to a question must be followed by a carriage return (CR). The program looks only at the last two or four digits, as appropriate, that have been entered before the CR. Any preceding digits are ignored. This feature provides a convenient means of correcting typing errors. The user needs only to continue typing until the correct digits have been entered before pushing CR. For example, typing

27142716(CR) is equivalent to

2716(CR)

Or, in response to the PAGE# = question, typing

41008(CR) is equivalent to

08(CR)

Inappropriate responses result in a "?" appearing and the question reasked as described in the subsection

"Error Messages". These kinds of errors are corrected by simply entering a valid response on the next try.

Errors of the third kind, in which an incorrect though valid response is made, must be corrected by pressing the RESET button on the MCDS and restarting the program with the monitor program's P command. For example, if a "V" command is inadvertently entered instead of the "P" command, the program should be restarted to correct the situation.

### Precautions

A PROM should never be inserted or removed with power on. The user should always make sure that the power ON/OFF switch S2 on the PROM programmer module is in the OFF position before inserting or removing a PROM.

To prevent any potential problems, the external power supplies (programming power and -9 volts), when present, should be sequenced on simultaneously or after the MCDS is turned on and shut off before or at the same time the MCDS is shut off.

There is another switch (S1) on the PROM Programmer board that selects between a 1702 and CDP18U42 configuration for socket 1.

**NOTE: This switch should never be in the 1702 position if a CDP18U42 is in the socket and the -9-volt supply active.** If such were the case, -9 volts would be connected to an input pin of the CDP18U42, and a destructive latch-up condition could be triggered in the device. If the program generates the warning message

**SET SWITCH!**

it detects the wrong position after the program has started and waits before proceeding. This warning, however, should not be relied on to prevent damage to the device. **Before inserting a CDP18U42, be sure switch S1 is in the correct position.**

The user should be sure to insert each PROM into its correct socket and be sure that it is properly oriented. Pin 1 is the pin adjacent to the handle of the zero-insertion force sockets. **Never have more than one PROM in the module at any time.** Because the PROM sockets share common address and data buses, contention or loading problems could be caused if more than one device is inserted. Additionally, extraneous signals to a socket which should be empty might cause harm to a device left in it.

The PROM programmer software derives its timing from the 2.0-MHz clock frequency of the crystal on the CDP18S601 Microboard Computer Module. If this frequency is changed, the results are unpredictable.

Also, note the power requirements for the PROM Programmer module as given in Appendix I. If additional modules are installed in the MCDS that could tax the internal power supplies to their limits, these modules should be removed before attempting to run the programmer.

Before starting a programming operation, be sure that the external programming power supply (connected to J3) is at the proper voltage. Excessive voltage can cause permanent damage to a PROM.

It is recommended that the programmer module not be run on an extender card in the MCDS system. An extender card will exaggerate bus-loading effects and may lead to intermittent operation of the programmer, particularly in systems already loaded with extra I/O or memory modules. If anomalous operation is observed, a first step should be to remove any extender cards and extraneous modules from the system.

## Operations and Commands

Following is a discussion of each command and operating sequence.

### Programming Operations— The “P” Command

The system is capable of programming the RCA CDP18U42 CMOS PROM and Intel 2708, 2758, and 2716 type PROM's. Data to be programmed comes from the RAM Buffer area, pages 8 and up in the MCDS RAM. The data must first be loaded into the appropriate page(s) by COPYing another PROM into them or by loading the appropriate section of a tape file into them. The file loading, when requested, leads automatically into a programming cycle. Refer to the section, **File Reader Subroutines** for a detailed discussion of the file loaders. Programming begins immediately from the specified page.

#### Procedure

The program always begins by printing  
**MCDS PROM PROGRAMMER VER X**  
**F, V, P, C, S, J, U?**

After the user types the letter ‘P’, the program asks  
**TYPE #?**

The user should enter one of the following:

**1842, 2708, 2758, or 2716**

followed by a (CR).

The program next asks if operation is to be with true or inverted data by typing

**LOGIC=P, N?**

A “P” response causes the operation to occur with true (non-inverted) data; an “N” response causes the operation to proceed using an inverse of the data. For example, to copy one PROM into another, inverting data, one could copy the master PROM using positive logic and program the copy PROM using negative logic or vice-versa. Negative logic would not be used for both the copy and program step because that would give a double inversion.

The program next asks

**(T)APE OR (R)AM?**

The user should enter the letter “R” if the data to be programmed is already in the buffer RAM.

The next prompt is

**PAGE# =**

asking for the first page on which data is located. Data is taken from that page and subsequent pages as required for the PROM being programmed.

If the user enters the letter “T”, indicating data to be programmed is on a tape file, the program will ask for the unit by typing

**TAPE#**

The user should respond with either a “0” or a “1”.

Next, the program asks for the address range of the PROM by typing

**LOWEST PROM ADDRESS (XX00)?**

The user should respond with a four-digit number followed by a (CR). This number should be the lowest address of data to be programmed into the PROM. For instance, if a 2704 PROM is to be programmed for data addresses 0200 through 03FF, then 0200 would be typed as a response. A default value of 0000 is entered if the user hits the carriage return without typing in a number. If a number is entered, **all four digits must be entered**. The file reader searches the object file and loads only the data falling between the lowest address specified and up to 2 kilobytes above that address. The appropriate data is copied into RAM starting at page 8 and then the actual programming begins automatically.

Note that programming will be initiated when the 2-kilobyte RAM area is filled or an end-of-file marker is read from tape. Thus, if the user enters an improper start address, the PROM will be programmed with invalid data.

Any programming operation is automatically followed by a VERIFY sequence. If the verify is successful, the program types

**DONE**

If programming is not successful, a list of errors is typed. This list can be stopped by depressing the BREAK key until the

**\*\*\*ABORTED\*\*\***

message is typed.

If programming has been done successfully, a message such as

**REPROGRAM 1842?**

appears, identifying the PROM type just programmed. If the user does not want to program another device of the same type with the same information, he should type “N” for NO. If the programming of another PROM of the same type is required, the user should take the following steps:

- Shut off power switch S2
- Remove the PROM just programmed
- Insert the new PROM to be programmed
- Turn on switch S2
- Type the letter “Y” for yes on the terminal.

Another programming sequence will begin immediately.

During a programming operation, the LED indicator labeled PROG on the programmer board is ON, providing a visual indication of operation. The programming operation can be halted at any time by

pressing and holding the BREAK key until the message

**\*\*\*ABORTED\*\*\***  
is typed.

The programming time required for each PROM type is given in Table XVI.

**Table XVI Programming Times—Typical Values.**

Type No.	Time
CDP18U42	3 seconds
2708	2 minutes 45 seconds
2716	1 minute 45 seconds
2758	50 seconds

### Examples

In the following examples, system-generated output is underlined.

1. Program a 2758 from RAM pages 8, 9

```
MCDS PROM PROGRAMMER VER X
F, V, P, C, S, J, U? P
TYPE #? 2758
LOGIC=P, N? P
(T)APE OR (R)AM? R    PAGE# = 8
DONE
REPROGRAM 2758? N
MCDS PROM PROGRAMMER VER X
F, V, P, C, S, J, U?
```

2. Program two CDP1842's with the same data from a tape on drive 1. Program for address 0100 to 01FF.

```
MCDS PROM PROGRAMMER VER X
F, V, P, C, S, J, U? P
TYPE #? 1842
LOGIC=P, N? P
(T)APE OR (R)AM? T
TAPE#? 1
LOWEST PROM ADDRESS (XX00)? 0100
LOADING COMPLETED    ..FILE LOADED
                        ..PROGRAMMING BEGINS
                        ..PROGRAMMING AND
                        ..VERIFY DONE
REPROGRAM 1842? Y    ..INSERT NEXT 1842,
                        ..THEN TYPE Y.
DONE                ..PROGRAMMING AND
                        ..VERIFY DONE
REPROGRAM 1842? N    ..RESTART PROGRAM
MCDS PROM PROGRAMMER VER X
F, V, P, C, S, J, U? U    ..RETURN TO MONITOR
*
```

### Verify Operations— The "V" Command

Verify operations compare a selected area of the RAM buffer against the contents of a PROM. As with the programming operation, the buffer is loaded by either COPYING a PROM into it or loading data from a file. The user can select the page(s) against which a

PROM is to be verified, or, if he chooses to use a file, the appropriate data is loaded starting at page 8 and automatically verified from there. See subsection "File Reader Subroutines" later in this section for details on the file reader operation.

### Procedure

A verify operation is automatically done after a PROGRAM or COPY operation. If there are no errors, the DONE message appears. Any mismatch between a PROM address and the corresponding buffer address results in an error listing such as

**ERR;M(0800)=FF, PROM=00**

This error listing says that location 0800 in the RAM buffer contains an FF, but the corresponding location of the PROM (which would be its first address in this case) contains a 00. A long error listing can be stopped by pressing and holding the BREAK key until the message

**\*\*\*ABORTED\*\*\***

is typed. The program then restarts from the beginning when the BREAK key is released.

The program begins by asking for the type of operation to be performed, for the PROM type number, and whether the operation is with positive or negative logic. The verify command has exactly the same command format and requirements as the "P" command.

At the end of a verify operation, the program will ask if another PROM of the same type is to be verified against the same RAM data. If the answer is no, the user should type an "N". If the answer is yes, the following steps should be taken.

- Shut off power switch S2
- Remove the PROM just verified
- Insert the new PROM to be verified
- Turn on switch S2
- Type the letter Y for yes on the terminal

### Examples

1. Verify a 2708 against RAM pages 9 through C.

```
MCDS PROM PROGRAMMER VER X
F, V, P, C, S, J, U? V
TYPE #? 2716
LOGIC=P, N? P
(T)APE OR (R)AM? R    PAGE# =9
DONE
REVERIFY 2708? N
MCDS PROM PROGRAMMER VER X
F, V, P, C, S, J, U?
```

2. Verify a 2716 against a tape file. Verify against addresses 0500 through 0CFF of the file.

```
MCDS PROM PROGRAMMER VER X
F, V, P, C, S, J, U? V
TYPE #? 2716
LOGIC=P, N? P
(T)APE OR (R)AM? T
TAPE#? 1
```

LOWEST PROM ADDRESS (XX00)? 0500  
LOADING COMPLETED

ERR:M(0800)=18, PROM=FF .ERROR LSTG STARTS  
ERR:M(0801)=C0, PROM=FF  
ERR:M(0802)=04, PROM=FF .BREAK KEY PRESSED  
 \*\*\*ABORTED\*\*\*

MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U?

## Copy Operations— The “C” Command

### Procedure

The copy operation reads the contents of a PROM into the selected page(s) of the buffer RAM. Make sure there is sufficient free RAM to hold all the data to be copied. For example, copying a 2716 requires two kilobytes of RAM buffer and the copy must begin on page 8 if there are only four kilobytes of RAM in a MCDS system. The program does not test for the presence of RAM at the assigned page(s). A copy operation is automatically followed by a verify. Upon completion of a copying operation, the program responds with a “DONE” message or a list of errors if the copy did not verify. Errors could be due to a hardware problem or attempting to copy into non-existent RAM.

A direct copy of the PROM is made if positive logic is chosen; the data is inverted if negative logic is chosen. A COPY followed by a PROGRAM operation is the sequence used to reproduce PROM's. Refer to the next Section for examples of compound operations.

### Example

Copy a 2758 into RAM pages 8, 9, A and B.

MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? C  
TYPE #? 2758  
LOGIC=P, N? P PAGE# =8  
 DONE  
MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U?

## The Fill Command

### Procedure

This command is used to fill selected page(s) of the buffer RAM with all 1's for positive logic, or all 0's for negative logic. The user should respond to the PAGE#= prompt with the starting page, as usual. The number of pages that will be filled is determined by the PROM type number entered—1 page for 1702 or CDP18U42, 4 pages for a 2758, etc.

This command is used in verifying that a PROM has been completely erased. The erased PROM is VERIFY'ed against a buffer filled with 1's or 0's, as appropriate.

### Example

Fill RAM pages B through E with 0's (4 pages).

MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? F  
TYPE #? 2708 .2708 IS 4-PG DEVICE  
LOGIC=P, N? N PAGE# =B  
 DONE  
MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U?

## The “U” Command

### Procedure

The “U” command returns control to the monitor program which responds by printing an asterisk (\*) prompt.

### Example

MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? U  
 \*

## The “J” Command

### Procedure

The “J” command allows the user to rewind tapes.

### Example

MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? J  
REWIND TAPES, THEN TYPE ANY KEY X  
MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U?

## The SAVE Operation— The “S” Command

The SAVE routine allows the user to copy any number of pages of the buffer RAM onto a file in reloadable format. The file thus generated will be in ASCII-Hex and can be used to program other PROM's. Also, it can be loaded into RAM for a ROM emulator function (a Micromonitor feature) or it can serve as the data for generation of masked ROM's such as the CDP1831/33.

### Procedure

The prompts for this command are different from the previous ones. The user is not asked for the type of PROM or the starting page. Saving always starts on page 8 and is done in positive logic. The number of pages to be saved is designated in hexadecimal—from 1 to F7. The S command, in conjunction with the tape file reader and other commands, can be used to merge or truncate files and combine several smaller PROM's for the generation of one larger ROM. See the next section for examples of compound command usage.

When the SAVE operation is started, it asks for the number of pages of data to be saved on tape by typing

# PAGES?

The SAVE operation then asks for the tape number the data is to be written to by typing

TAPE#?

The user should respond with either a "0" or a "1".  
The response to the prompt

**LOWEST PROM ADDRESS (XX00)?**

is used to generate the starting address on the file. For example, the response "0100" causes data to be written on the file as follows:

```
I
0100 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX;
0110 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX;
      etc.
```

Be sure the correct starting address is assigned.

SAVE'd files on tape cassette will have a leader and trailer of nulls appended automatically.

#### Example

Save RAM pages 08 through 12 (hex) under the file on a tape in drive #1. The data is from a PROM with starting address 0200.

```
MCDS PROM PROGRAMMER VER X
F, V, P, C, S, J, U? S
# PAGES? A ..TEN PAGES WILL BE SAVED
TAPE#? 1
LOWEST PROM ADDRESS (XX00)? 0200
DONE
MCDS PROM PROGRAMMER VER X
F, V, P, C, S, J, U?
```

### File Reader Subroutines

The File Reader Subroutine is a program called by either the PROGRAM or VERIFY routines. It loads data from a file into the buffer RAM starting at page 8. It is an intelligent loader that will search through a file and load data in the address range of interest (+2 kilobytes from the initial address) ignoring extraneous data. It will handle either hex or listing files created by the COSMAC assemblers and files created by the monitor program (W command). When the user replies to the prompt

**LOWEST PROM ADDRESS (XX00)?**

the program takes that address as the first address in the range and adds 2K to get the last address. It will thus load up to 2 kilobytes of data, filling the basic buffer (but not writing into any additional CDS RAM) if there is sufficient data. Of course, if the file contains less than 2 kilobytes in the address range, some of the buffer will not be used. Once a 2-kilobyte file is loaded, small PROM's can be programmed from successive pages of the buffer. Refer to the next section for an example.

If no data is found, the program types

**NO OBJECT CODE**

In the tape cassette system, the program will restart from the beginning.

Format errors detected on the cassette will cause the program to type the following messages and restart from the beginning.

**FORMAT ERROR**

**LOAD ABORTED**

When loading starts, the message

**LOADING**

is typed. At the completion of a successful load, the word COMPLETED is added to the sentence. The program then goes automatically into the program or verify operation.

**NOTE:** In response to the prompt

**LOWEST PROM ADDRESS (XX00)?**

a full four-digit number followed by a carriage return (CR) should be entered. If the user hits CR without entering any numbers, the default address of 0000 is used. Typing mistakes can be corrected by continuing to type until the correct four digits have been entered before hitting CR. Only the last four digits before the CR are recognized and only the first two of the group are meaningful—the last two are always taken as 00. Below are some examples.

Response	Yields
0200(CR)	0200
0251(CR)	0200
2(CR)	0000
030244(CR)	0200
03029(CR)	3000
(CR)	0000

### Application Hints

It is good practice to verify that a PROM is erased before attempting to program it. A suitable technique is to use the "F" command to fill the buffer with 1's or 0's and then the "V" command to verify the PROM against it. When a PROM is being programmed to work in an RCA 1800-series microprocessor-based system, it is advisable to first fill the buffer with all 0's before loading in the data to be programmed. This step will set data in the unused addresses of the PROM to 00, which is an IDLE instruction in the RCA 1800-series instruction set. Thus, an erratic program fetching from an erroneous address will pick up the idle instruction and stop. A PROM thus programmed must use the same procedure when it is verified. That is, the buffer is first filled with 0's, the data loaded, and the verify operation is made. This technique assures that no "errors" will appear when the programmed "00" locations are verified against random data left in the buffer.

An application note (ICAN-6847) describing a hardware adjunct that can be used to program Intel 2732 PROM's with this system is available on request.

## Examples of Compound Operations

This section gives five examples of compound operations.

### Example 1

Program a CDP18U42 from a 1702.  
MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? C ..COPY THE 1702 INTO RAM  
 TYPE #? 1702  
 LOGIC=P, N? P PAGE#=8  
 DONE ..REMOVE 1702, SET SWITCH,  
 ..INSERT 1842

MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? P  
 TYPE #? 1842  
 LOGIC=P, N? P  
 (T)APE OR (R)AM? R PAGE#=8  
 DONE  
 REPROGRAM 1842? N ..RESTART PROGRAM

### Example 2

Verify that a 2758 has been erased, then program it from a file. The PROM will occupy addresses from 400-5FF.

MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? F  
 TYPE#? 2758  
 LOGIC=P, N? P PAGE#=8  
 ..FILL PGS. 8, 9, A, AND B WITH FF'S  
 DONE  
MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? V  
 TYPE #? 2758  
 LOGIC=P, N? P  
 (T)APE OR (R)AM? R PAGE#=8  
 DONE ..ERASURE VERIFIED—NOW PROGRAM  
 REVERIFY 2758? N  
MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? P  
 TYPE#? 2758  
 LOGIC=P, N? P  
 (T)APE OR (R)AM? T  
 TAPE#? 1  
 LOWEST PROM ADDRESS (XX00)? 0400  
 LOADING COMPLETED  
 DONE ..PROGRAMMED AND VERIFIED  
 REPROGRAM 2758? N ..RESTART PROGRAM

### Example 3

Combine and save two CDP18U42's on tape to make a CDP1831 ROM with address range 0100-02FF.

MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? C ..COPY 1ST PROM INTO PG 8  
 TYPE #? 1842  
 LOGIC=P, N? P PAGE# =8  
 DONE

MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? C ..COPY 2ND PROM INTO PG 9  
 TYPE #? 1842  
 LOGIC=P, N? P PAGE# =9  
 DONE  
MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? S  
 # PAGES? 2  
 TAPE#? 1  
 LOWEST PROM ADDRESS (XX00)? 0100  
 DONE

### Example 4

Program two 2708's from a file. One occupies addresses 0000-03FF and the other 0400-07FF.

MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? P  
 TYPE #? 2708  
 LOGIC=P, N? P  
 (T)APE OR (R)AM? T  
 TAPE#? 0  
 LOWEST PROM ADDRESS (XX00)? (CR)  
 LOADING COMPLETED  
 DONE ..1ST PROM PROGRAMMED  
 ..FOR 0000-03FF  
 REPROGRAM 2708? N  
MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? P  
 TYPE#? 2708  
 LOGIC=P, N? P  
 (T)APE OR (R)AM? R PAGE#= C  
 DONE ..FOR 0400-07FF  
 REPROGRAM 2708? N

### Example 5

Program two 2716's from a file. One occupies addresses 0000-07FF and the other 0800-0FFF.

MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? P  
 TYPE #? 2716  
 LOGIC=P, N? P  
 (T)APE OR (R)AM? T  
 TAPE#? 0 ..LOAD 1ST ADDRESS RANGE  
 LOWEST PROM ADDRESS (XX00)? (CR)  
 LOADING COMPLETED  
 DONE ..1ST PROM PROGRAMMED  
 REPROGRAM 2716? N  
MCDS PROM PROGRAMMER VER X  
 F, V, P, C, S, J, U? P  
 TYPE #? 2716  
 LOGIC=P, N? P  
 (T)APE OR (R)AM? T  
 TAPE#? 0 ..LOAD 2ND ADDRESS RANGE  
 LOWEST PROM ADDRESS (XX00)? 0800  
 LOADING COMPLETED  
 DONE ..2ND PROM PROGRAMMED  
 REPROGRAM 2716? N