

OPERATIONS
MANUAL

PROM-100
PROM PROGRAMMER

COPYRIGHT © 1979
BY SD SYSTEMS
JULY 1979

REVISION A

TABLE OF CONTENTS

SECTION	DESCRIPTION	PAGE
I-1	INTRODUCTION	1
I-2	GENERAL DESCRIPTION	1
I-3	PHYSICAL DESCRIPTION	1
I-4	SPECIFICATION	1
II	FUNCTIONAL DESCRIPTION	
2-1	INTRODUCTION	3
2-2	DATA OUT BUS	3
2-3	DATA IN BUS	3
2-4	A ₀ -A ₇	3
2-5	I/O CONTROL LINES	4
2-6	ADDRESS DECODER	4
2-7	OUTPUT PORT 68H	4
2-8	OUTPUT PORT 69H	4
2-9	OUTPUT PORT 6AH	4
2-10	INPUT PORT 68H	5
III	CONTROL SOFTWARE	
3-1	INTRODUCTION	6
3-2	2758/2716/2732 PROGRAMMING	6
3-3	2708 PROGRAMMING	7
3-4	LOAD SEQUENCE	9
3-5	READ SEQUENCE	9
3-6	PROGRAMMING SEQUENCE	9
3-7	CAUTION:	10
3-8	USING THE PROM PROGRAMMER	10
3-9	LOADING AN OBJECT FILE	11
3-10	READING A PROM	12
3-11	PROGRAMMING A PROM	12
3-12	COPYING A PROM	14
IV	CONSTRUCTION	
4-1	INTRODUCTION	15
4-2	ASSEMBLY PROCEDURE	15
4-3	VOLTAGE CHECK	16
V	SWITCH SELECTION	
5-1	INTRODUCTION	18
VI	CHECK OUT	
6-1	INTRODUCTION	19
6-2	I/O PORT VERIFICATION	19
VII	SOFTWARE REQUIREMENTS	
7-1	INTRODUCTION	20
	APPENDICES	
APPENDIX A	SCHEMATIC	
APPENDIX B	PART LIST	
APPENDIX C	PARTS PLACEMENT	
APPENDIX D	SOFTWARE LISTING	

SECTION I

1-1 INTRODUCTION

The SD SYSTEMS Prom Programmer Board (PROM-100) provides a low cost means for programming read only memory capability for computers utilizing the S-100 bus structure.

1-2 GENERAL DESCRIPTION

The SD SYSTEMS PROM-100 board is a high performance circuit capable of programming the industry standard 2708 1K X 8 EPROM, Intel's 2758 1K X 8 EPROM, 2716 2K X 8 EPROM, 2732 4K X 8 EPROM and Texas Instruments 2516 2K X 8 EPROM. The PROM-100 is selected for the appropriate EPROM type through dip switches S1 and S2. Another notable feature of the PROM-100 is the 25V programming pulse generated on board, eliminating the need for an external 25V supply.

1-3 PHYSICAL

The SD SYSTEMS Prom Programmer is implemented on a single 7.00" X 10.0" X .065" printed circuit board and is interfaced to the System by connector J-1.

1-4 SPECIFICATION

Table 1-1 lists the overall specifications for the SD SYSTEMS Prom Programmer Board. Table 1-2 lists the pin usages of connector J-1 for the Prom Programmer Board.

TABLE 1-1

+8 to +10V	300 ma maximum
+16 to 18V	100 ma maximum
-16 to 18V	60 ma maximum
Operating temperature	0°C to 50°C
Interface levels	TTL Compatible
Programming time	100 seconds max for 16,389 bits

TABLE 1-2

PIN NO	SIGNAL NAME	DIRECTION	DESCRIPTION
1,51	+8V to +10V	INPUT	POWER
2	+16V to +20V	INPUT	POWER
100,50	GND		
79,80,81,31,30 29,82,83	A0-A7	INPUT	ADDRESS BUS
36,35,88,89,38 39,40,90	D0-0 to D0-7	INPUT	DATA BUS OUT
95,94,41,42,91, 92,93,43	DI-0 to DI-7	OUTPUT	DATA BUS IN
77	PWR		WRITE
78	PDBIN		DATA BUS IN
46	SINP		PORT INPUT CYCLE
45	SOUT	INPUT	PORT OUTPUT CYCLE
99	POC	INPUT	POWER ON CLEAR

SECTION II

FUNCTIONAL DESCRIPTION

2-1 INTRODUCTION

Functionally, the PROM Programmer consist of two main portions; hardware, and the software. The hardware allows the computer to select the port bits, generate the programming pulse, and control the decoder. The software must direct the hardware in each of the operations described above. The major functions contained in the PROM Programmer hardware are shown in the block diagram (Figure 2-1).

2-2 DATA OUT BUS

The 8-Bit Data Out Bus is the S-100 path for transferring data from the computer (CPU) to the output ports on the Prom Programmer board.

2-3 DATA IN BUS

The 8-Bit Data In Bus is the S-100 path for transferring data from the input port on the Prom Programmer Board to the computer (CPU).

2-4 A0-A7

The A0-A7 low order eight address lines are used by the computer (CPU) to select the various input/output ports through a decoder on the Prom Programmer Board.

2-5 I/O CONTROL LINES AND READ WRITE CONTROL

The I/O Control Lines consist of \overline{PWR} , PD BIN, SINP and SOUT.

These lines are used to control the input and output operations from/to the I/O ports on the board.

2-6 ADDRESS DECODER

The Address Decoder detects when a port address used by the Prom Programmer is present on (A0-A7). The output of the decoder is used to gate read and write pulses to the I/O ports.

2-7 OUTPUT PORT 68H

Output port 68H is an 8 bit data register that transfers the data from CPU to ROM.

2-8 OUTPUT PORT 69H

Output port 69H is an 8 bit address register that determines the location in the ROM for the data to be programmed or read.

2-9 OUTPUT PORT 6AH

Output port 6AH is an 8 bit control register with several control functions.

1. Bit 1 & 2 Two high order address bits A8 & A9 address inputs.
2. Bit 3 One high order bit A10 (address input for 2716).
3. Bit 4 One high order bit A11 (address input for 2732).
4. Bit 6 Program pulse for 2716/58.
5. Bit 7 Drive MOS clock drive.
6. Bit 8 Controls port 68H chip select for 2716/58/08.

2-10 INPUT PORT 68 H

Input port 68H is used to read the present data in the ROM.

PROM - 100

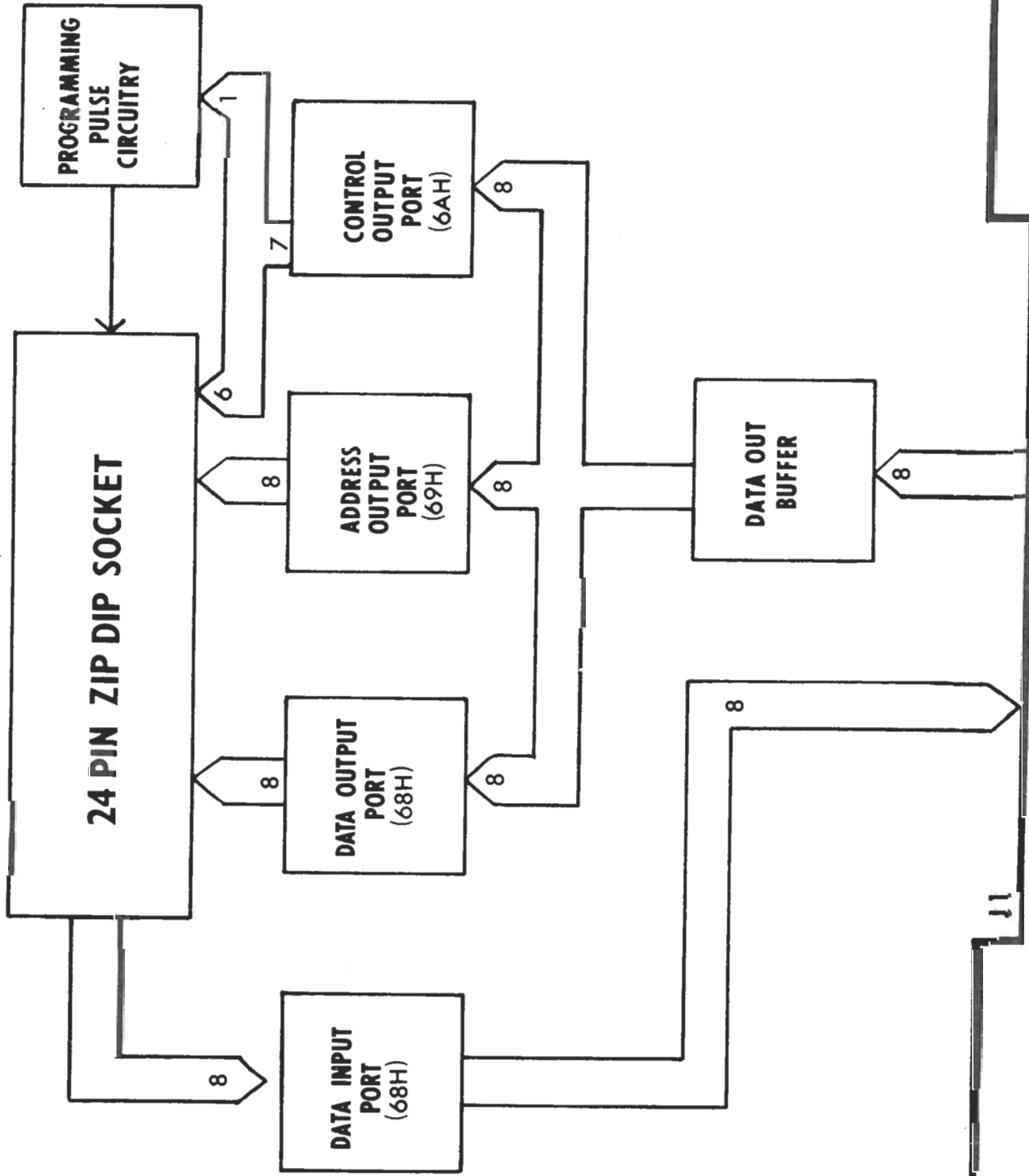


FIGURE 2-1

SECTION III
CONTROL SOFTWARE

3-1 INTRODUCTION

Certain sequences must be executed to ensure proper programming of the prom.

3-2 2758/16/32 PROGRAMMING

Initially, and after each erasure, all bits of the 2716 are in the "high" state. Data is introduced by selectively programming "0's" into the required locations. Although only "0's" will be programmed both "1's" and 0's" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2716 is in the programming mode when the VPP power supply is 25V and \overline{CS} is at a low TTL level. The data to be programmed is applied 8 bits in parallel to the data output port. The levels required for the address and data inputs are TTL.

When the addresses and data are stable a 50 MSEC, active high, TTL program pulse is applied to the appropriate input pin on the EPROM.

A program pulse must be applied to each address location to be programmed. The user can program any location at any time - either individually, sequentially, or at random. The program pulse has a maximum width of 55 MSEC.

A verify is performed on the program bits to determine that they are correctly programmed. The verify may be performed with VPP at 25V.

3-3 2708 PROGRAMMING

Initially and after each erasure all 8192 bits are in high state (output high). Information is introduced by selectively programming "0" into the desired bit location. A programmed "0" can be changed to a "1" by UV. erasure.

The circuit is set up for programming operation by raising the CS/WE input (pin 20) to +12V. The word address is selected in the same manner as in the read mode. Data to be programmed is presented 8 bits in parallel to the data output lines (D0₀-D0₇) Logic levels for address and data lines and the supply voltages are the same as for the read mode.

After address and data set up, one program pulse per address is applied to the program input pin 18. One pass through an address is defined as a program loop. The number of loops (N) required is a function of the program pulse width (tpw) allowing to $(N) \times (tpw) \geq 100$. tpw in MSEC.

The width of the program pulse is from 0.1 to 1ms. The number of loops (N) is from a minimum of 100 (tpw=1ms) to greater than 1000 (tpw=0.1). There must be N successive loops through all 1024 addresses. It is not permitted to apply N program pulses to an address and then change to the next address to be programmed.

*The PROM Programmer writes data received from the host computer into the PROM as guided by the address/control outputs supplied by the driver software. Basically, the PROM Programmer manipulates the data and the address/control inputs from the computer so that the subject PROM receives proper data, address and control inputs. The PROM-100 generates the programming pulse for the 2708 from the output of a step up switching voltage regulator provided on the board. The same pulse is used to provide VPP (25V) at pin 21 of 2716/58. This 25V is set up by the driver software of the PROM Programmer.

Driver software for PROM Programmer must meet three requirements. These are: loading the object modules (diskettes) into host computer memory, reading a PROM into memory and programming a PROM. Within the software, the executive system coordinates all three operations while each operation sequence is run by its respective routine. In regards to the PROM Programmer, only the read and program sequences are significant because they involve specific address/control inputs to the PROM Programmer. The requirements imply that, in addition to allocating a memory buffer for the driver software, the host computer must also supply a 1, 2 or 4K byte memory buffer for the data module to be loaded, read, and programmed.

3-4 LOAD SEQUENCE

The load sequence must be able to load data into memory at any location regardless of the routine load address contained in the file. Once the number of bytes to be loaded has been specified, the user may load and program a portion of the file and then return for the rest. For example, if a file to be programmed contains 4K bytes, the user could program 2K bytes into one 2716 PROM and then program the remaining 2K bytes of the module into a second PROM.

3-5 READ SEQUENCE

The read sequence is necessary in order to perform a PROM copy function. Since the PROM programmer contains only one 24 pin ZIP DIP socket, the master PROM must first be read into the 1K, 2K or 4K memory buffer (2708, 58 or 2716/32) and then programmed into an erased PROM. The host computer should offer memory editing to allow modification of the memory buffer before programming the new PROM.

3-6 PROGRAMMING SEQUENCE

The programming sequence actually involves a three step process.

1. Verify that PROM is erased.
2. Program the PROM.
3. Verify that the PROM contains correct data.

The entire PROM is read to verify erasure, that each location contains FF_H. If unerased locations are found, the details are output to the console.

The programming of the PROM involves outputting the data for each of the 1K, 2K or 4K locations sequentially along with the PROM address and the 50ms program pulse for 2758/16/32 or 1ms program pulse for 2708. After this programming attempt is complete, the entire PROM must be read and compared with the memory buffer. If any locations do not match, the address and data should be output to the console.

3-7 CAUTION:

In order to prevent possible destruction of PROMS, PROMS must never be inserted into the ZIP DIP socket on the PROM Programmer until the driver software is running. If the PROM is inserted prior to starting the driver software, the PROM may be subjected to a continuous program pulse which will destroy the PROM. For the same reason, the host computer must not be reset while the PROM is in the socket.

3-8 USING THE PROM PROGRAMMER

The driver software can be used to load object files from diskettes, read object code from PROMS and program object code into PROMS. As the driver software starts, the console prints the start/options message. The sign-on portion of the message indicates that the PROM Programmer software is operative. The options portion consist of the first of three questions about which operations are

to be performed (load a file, read a PROM, and program a PROM), in that order. The user answers the questions by entering a Y (for Yes) or N (for No) at the end of each question. If a Y is entered the respective operation is performed after the necessary parameters are entered.

3-9 LOADING AN OBJECT FILE DISK FILE ONLY

Prior to answering the first question: READY TO LOAD A FILE (Y/N)? the user sets up the object file by specifying the file name, when loading the PROM-100 Software: A> PROM08 filename.Ext. When the user answers the question with a Y, the PROM Programmer software asks the user to enter two parameters (load start address and load size) by printing HEX LOAD address, #Bytes: The parameters must be separated by a comma or space and terminated with a carriage return. If no parameters are entered in front of the carriage return, the object file will be loaded into the RAM locations specified on the module. However, since the PROM Programmer driver software occupies the memory location 0100_H through $5FF_H$ the object file must not overlay these locations. This condition can be prevented by using the load address parameter which allows relocating the object data to any area of memory. The number of bytes parameter allows the loading of a limited number of bytes of the file into RAM so that object file may be loaded and programmed in blocks. If the load size parameter is not entered, the entire module is loaded. If a period is entered before the end of the parameter, the software repeats

the request for parameters allowing the user to correct the parameter. When the load is complete the PROM Programmer software asks: READY TO PROGRAM A PROM (Y/N)?

3-10 READING A PROM

In the console printout - READY TO READ A PROM (Y/N)? If a Y is entered, the PROM Programmer software asks the user to enter three parameters (MEM start address, MEM end address, PROM start address) by printing MEMORY START, MEMORY END, PROM START:

The MEM start and END parameters specify the memory space into which the data is to be read. The PROM START defines the first PROM location to be read. The three parameters must be separated by a comma or space and terminated with a carriage return. If a period is entered before the end of the parameter the software repeats the request for parameters allowing the user to correct the parameters. When the read is complete or when the user answers the question: READY TO READ A PROM (Y/N)? With an N, the PROM Programmer software asks: READY TO PROGRAM A PROM (Y/N)?

3-11 PROGRAMMING A PROM

When the user answers the question: READY TO PROGRAM A PROM (Y/N)? with a Y, the programmer software asks the user to enter three parameters MEM start address, MEM end address and PROM start address by printing MEMORY START, MEMORY END, PROM START: The MEMORY START and END parameters specify the memory space from which the data is to be programmed. The three parameters must be separated by a comma or space and terminated with a carriage return.

If a period is entered before the end of the parameters, the software repeats its request for parameters allowing the user to correct the parameters.

The PROM Programmer software first verifies that the PROM is erased (contains FF_H in all locations). If the PROM Programmer software finds unerased locations, it prints the first unerased location. To check for other unerased locations the user enters a carriage return after each unerased location printed. If the PROM contains unerased location, the PROM must be erased again before starting programming. If there are unerased locations, the user may attempt programming by entering C. If there are no unerased locations, programming begins automatically. During the programming sequence, the PROGRAMMING LED COMES ON and stays on till the program ends. To stop programming during the sequence the user enters a period (.). The PROM Programmer software asks DO YOU WISH TO LOAD A MODULE (Y/N)? when the programming sequence is complete, the PROM Programmer reads the PROM to verify that the data has been programmed correctly. If the PROM Programmer software finds locations with BAD data, it prints the bad locations. To stop the error listing the user enters a period and the software returns to the start asking: READY TO LOAD A FILE (Y/N)? If the PROM contains bad locations it should be re-erased and re-programmed until it is correctly programmed. When programming is done the user can exit the PROM Programmer software by typing a period (.).

3-12 COPYING A PROM

Copying a master PROM into a subject PROM requires only that the master PROM be read as described in paragraph 3-10 and the subject PROM be programmed as described in paragraph 3-11.

SECTION IV
CONSTRUCTION

4-1 INTRODUCTION

The SD SYSTEMS PROM Programmer Board is intended for those people who have had some prior experience with kit building and digital electronics. If you do not fall into this category it is highly recommended that you either [1] find an experienced person to help you in assembly and check out the board. Appendix A shows the parts list for SD SYSTEMS PROM Programmer Board. Double check all parts against this parts list. If any differences are noted please contact SD SYSTEMS.

NOTE: General construction information assembly diagram and schematic diagram can be found in the appendices.

4-2 ASSEMBLY PROCEDURE (SEE APPENDIX C)

.1. Install the IC sockets in their proper locations.

NOTE: No socket or DIP switches: S1 & S2.

2. Install the resistors as follows:

A.	R8	220 Ohm 1/2W 5% (Red,Red,Brown)
B.	R4	22.1K Ohm 1/8W 1% (Precision Resistor)
C.	R5	1 Ohm 1/4W 5% (Black, Red,Black)
D.	R6	270 Ohm 1/4W 5% (Red,Violet,Brown)
E.	R7	100 Ohm 1/4W 5% (Brown,Black,Brown)
F.	R1,2,3	1K Ohm 1/4W 5% (Brown,Black, Red)

3. Install diodes CR1 with banded end as shown on the PC Board.
 - A. CR1 1N5803
 - B. CR2 Zener Diode 1N75] -5V
4. Install the capacitors as follows.
 - A. C1 - C3, C6 - C11, C14, C15 0.1 MF Ceramic
 - B. C12, C16 - C18 10 MF 20V Tantalum
 - C. C13 .002 MF Ceramic
 - D. C4 500 MF 35V Dielectric Axial Leads
 - E. C5 300pf
5. Install the voltage regulators with the heatsinks 6106-13 using #6-32 hardware supplies.
VR1 +5V 7805/LM340T-5
VR2 +12V 7812/LM340T-12
6. Install L1 coil.
L1 J12044 500 Micro Henry
7. Double check all solder connections for cold solder joints, unsoldered connections or shorted connections.

4-3 VOLTAGE CHECK

1. Install the board in the computer and measure the output of +5V regulator, +12V regulator and -5V of CR1 respectively.
 - A. VR1 = +5V (TOP Pin)
 - B. VR2 = +12V (Bottom Pin)
 - C. CR1 = -5V (Anode)

NOTE: Do not proceed with board check out until all power supply voltages are correct. The TTL and MOS logic can be permanently damaged if improper voltages are applied.

2. Install the IC's in their sockets observing the Pin 1 designation on each socket marked on the PC Board.

✓A.	U1	24 Pin Zip Dip socket (Pin 1 lever in lower left.)
✓B.	U2	TL497ACN
✓C.	U3	74LS30
✓D.	U4	74LS155
✓E.	U6, U8	74LS273
✓✓F.	U5, U10	74LS14
✓G.	U7	74LS373
✓H.	U9	7406
✓I.	U11	74LS08
✓J.	U12, 13	74LS244
✓K.	U14	DS0025CN

3. Double check all IC's for proper orientation and location.
4. Refer the switch selection option and position then as described in Section V.

SECTION V
SWITCH SELECTION

5-1 INTRODUCTION

PROM Programmer has been designed to accomodate a variety of PROMS. Care must be taken to select the right positions of the switches for the users specific PROM selection. Table 5-1 lists the proper switch positions for each PROM type.

PROM TYPE	S1	S2
2708 () 1K x 8	All other	2708
2758 (5V)	All other	All other
2716 (5V)	All other	All other
2732 (5V)	2732	All other

Table 5-1

SECTION VI

CHECK OUT

6-1 INTRODUCTION

This section will describe some basic checks that should be made on the PROM Programmer.

NOTE: It is assumed at this point that the voltage checks described in Section 4 have been previously made. The following checks require that the CPU board also be plugged into the bus.

6-2 I/O PORT WRITE/READ VERIFICATION

Using the monitor in the system or a short program. Write data to output port 68 and read it back. Verify that the data read back is the same as that was written. This can be done as follows if using the monitor.

```
.0 6A 80 (cr) - enable Port 68
.0 68 dd (cr) - output data (dd) to port 68
.I 68 (cr) - input from port 68
68 dd - Data Read should be identical to data output
```

SECTION VII
SOFTWARE REQUIREMENTS

6-1 INTRODUCTION

The SD Systems PROM-100 Programming Board software is available on disk on Intel 2716 EPROM.

A listing of the software in EPROM is included in Appendix D, while the disk version is included on the disk. (.PRN extension)

The program in PROM must be moved to 100H prior to execution. PROM 08, the 2708 programming utility, is located in the first 1K section of the PROM (0-3FFH). PROM 16, the Intel 2758, 2716, 2732 programming utility is located in the latter 1K section of the PROM (400_H-7FFH).

APPENDIX A
SCHEMATIC

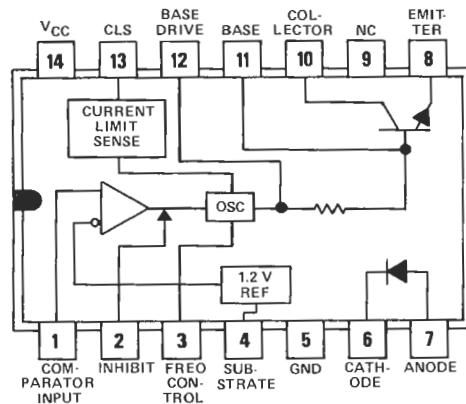
TL497A switching voltage regulator for DC-to-DC conversion

- Fixed on time simplifies design
- Single ended output (500 mA)
- Internal current protection of switching components
- Internal temperature stabilized 1.2 volt reference
- Capacitor programmed oscillator
- External inhibit control
- Adjustable output voltage
- Soft-start capability

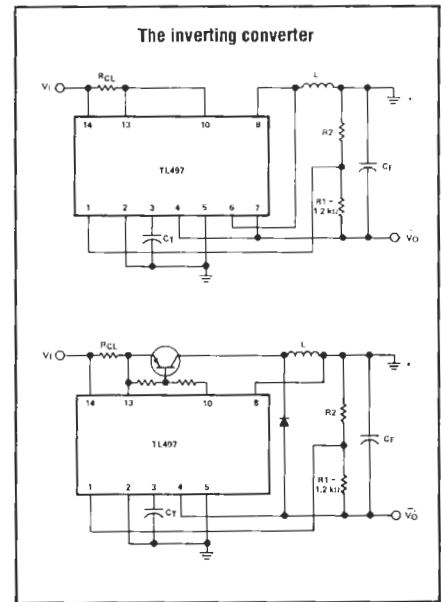
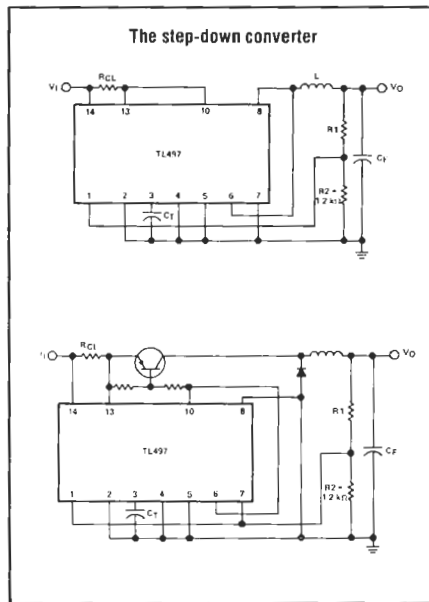
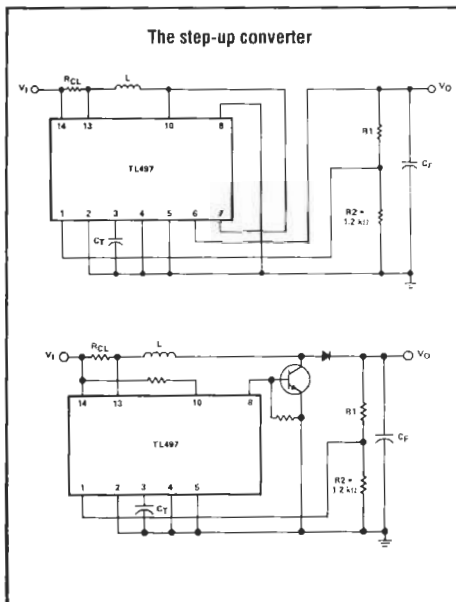
Description

Simplicity of design is a primary feature of the TL497A. With only six external components (three resistors, two capacitors, and one inductor), the TL497A will operate in numerous voltage conversion applications (step-up, step-down, invert) with as much as 85% of the source power delivered to the load. The TL497A replaces the TL497 in all applications.

DUAL-IN-LINE PACKAGE
(TOP VIEW)



Typical applications




APPENDIX B

PART LIST

SD Systems

P.O. Box 28810 • Dallas, Texas 75228 214-271-4667

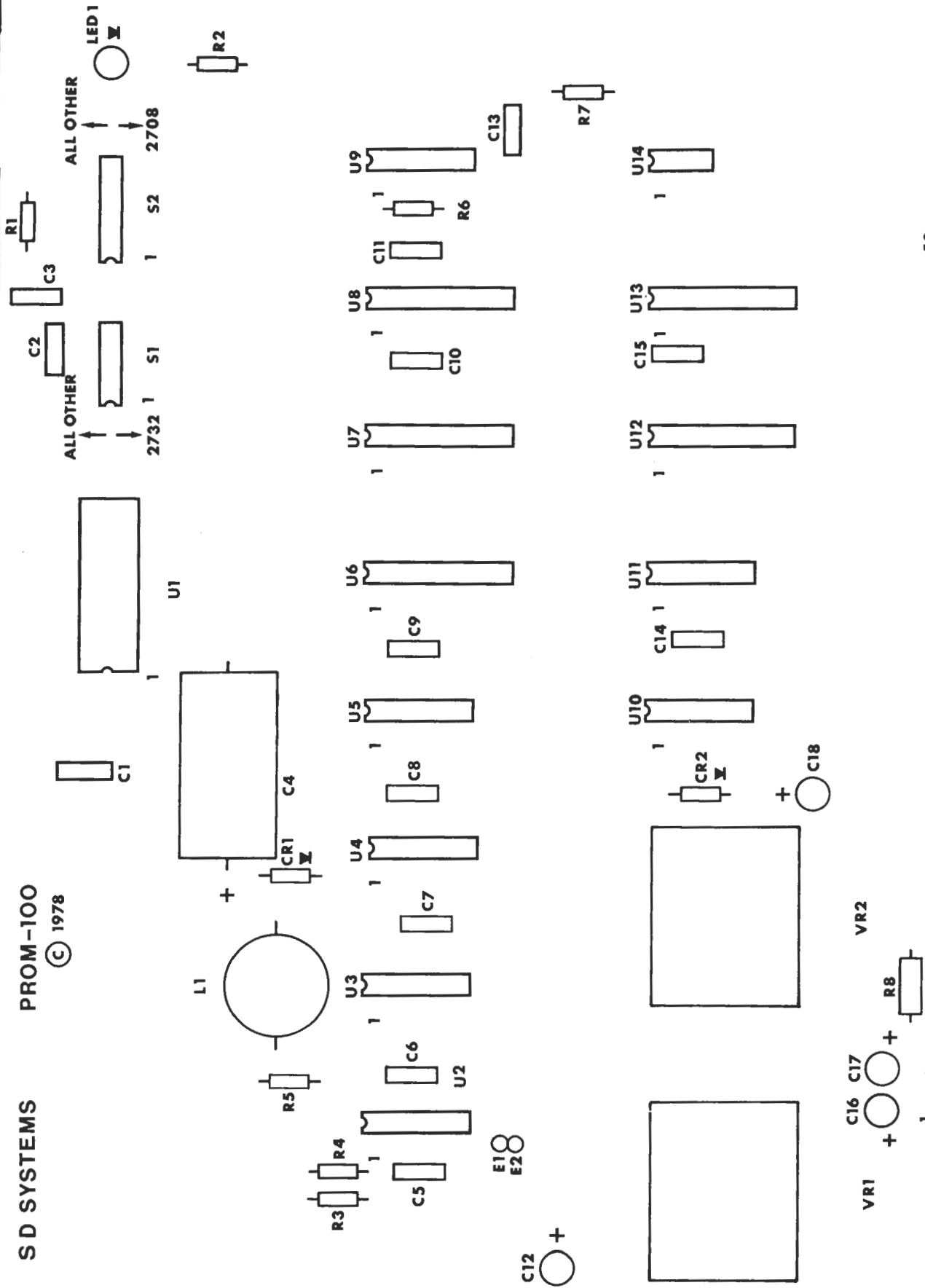
BILL OF MATERIALS

Title:			PL No.	Rev.	
PROM-100			0100140		
Date Released:		Approved:		Sheet	Of
5/14/79				1	2
Item no	Qty	SD-P/N	Description	Unit Cost	Extension
1	1	7010007	7406, U9		
2	1	7010166	74LS08, U11		
3	2	7010172	74LS14, U5, U10		
4	1	7010180	74LS30, U3		
5	1	7010226	74LS155, U4		
6	2	7010264	74LS244, U12, U13		
7	2	7010276	74LS273, U6, U8		
8	1	7010304	74LS373, U7		
9	1	7010353	DS0025CN, U14		
10	1	7010354	TL497ACN, U2		
11	1	7060010	24 PIN ZIP DIP, U1		
12	11	7030007	.1 MF Ceramic, C1-C3, C6-C11, C14-C15		
13	1	7030022	500MF 35V, C4		
14	1	7030006	300 pf, C5		
15	4	7030009	10 MF 20V, C12, C16-18		
16	1	7030034	.002 MF Ceramic, C13		
17	3	7020073	1K Ohm 1/4W 5%, R1-R3		
18	1	7020175	22.1K Ohm, 1%, R4		
19	1	7020001	1 Ohm 1/4W 5%, R5		
20	2	7020059	270 Ohm 1/4W 5%, R6, R7		
21	1	7020177	220 Ohm 1/2W, 5%, R8		
22	1	7060003	16 Pin Socket, 51		
23	1	7120001	L1 Inductor 500 uh J12044		
24	1	7040003	1N751, CR2		

APPENDIX C
PARTS PLACEMENT

SD SYSTEMS

PROM-100
© 1978



APPENDIX D
SOFTWARE LISTINGS


```

ADDR  CODE      STMT SOURCE STATEMENT

0001          NAME      PROM08
0002 ;
0003 ;
0004 ; 2708 PROM PROGRAMMER CONTROL PROGRAM
0005 ; 8/20/79
0006 ;
0007 ; VERSION 1.01  PROM VERSION (NON-DISK I/O)
0008 ;
0009 ;
0010 ; SYSTEM EQUATES
0011 ;
0012 ;
>0001 0013 SDATA  DEFL    01H      ; VIDEO DATA PORT
>0000 0014 SSTAT  DEFL    00H      ; VIDEO STATUS PORT
>0002 0015 RXRMSK DEFL    02H      ; RX RDY MASK
>0004 0016 TXRMSK DEFL    04H      ; TX RDY MASK
>E003 0017 EXIT   EQU     0E003H
>0100 0018 TBASE  EQU     100H      ; BASE OF TRANSIENT AREA
>0068 0019 DATA  EQU     68H      ; DATA PORT
>0069 0020 LADD   EQU     69H      ; LOW ADDRESS PORT
>006A 0021 HACTL  EQU     6AH      ; HIGH ADDRESS + CONTROL PORT
>0000 0022 RBOUT  EQU     0
>0064 0023 LOOPS  EQU     100
0024 ;
0025 ;
0026 ;
0027 ;
0028 ;
0029 ;
0030          PSECT   ABS
0031 ;
>0100 0032          ORG     TBASE    ; START AT 100H
0033 ;
0034 ;
0035 ;
0036 ; PROGRAM PROM ENTRY POINT
0037 ;
0038 ;
0039 ;
0100 312005 0040 PPG     LD      SP,SPVAL
0103 215B01 0041          LD      HL,MSG1
0106 CD4703 0042          CALL   PTXT    ; "PROM08 VERSION 1.0"
0109 312005 0043 PPG0    LD      SP,SPVAL
010C AF      0044          XOR     A        ; CLEAR ACC
010D D36A   0045          OUT    (HACTL),A
010F CD3803 0046          CALL   CRLF
0112 C3C501 0047          JP     READ2   ; JUMP READ
0115 CD3803 0048 PPG10A  CALL   CRLF
0118 216B01 0049          LD      HL,MSG4
011B CD4703 0050          CALL   PTXT    ; "READY TO PROG A PROM ?"
011E CD3901 0051          CALL   YORN
0121 3027   0052          JR     NC,RENTY-$
0123 CD3803 0053 PPG11  CALL   CRLF
0126 218A01 0054          LD      HL,MSG5
0129 CD4703 0055          CALL   PTXT    ; "MEM START, MEM END, PROM START"
012C CDBE03 0056          CALL   SCAN
012F 3A1F04 0057          LD      A,(NXCHR)
0132 FE2E   0058          CP     '.'      ; IF '.' EXIT

```

ADDR	CODE	STMT	SOURCE	STATEMENT
0134	28ED	0059		JR Z,PPG11-\$
0136	C31402	0060		JP PROG
		0061	;	
		0062	;	
		0063	;	
		0064	;	
		0065	;	
0139	CDAC03	0066	YORN	CALL ECHO
013C	FE2E	0067		CP '.'
013E	280A	0068		JR Z,REENTRY-\$
0140	B7	0069		OR A
0141	FE4E	0070		CP 'N'
0143	C8	0071		RET Z
0144	FE59	0072		CP 'Y'
0146	20F1	0073		JR NZ,YORN-\$
0148	37	0074		SCF
0149	C9	0075		RET
		0076	;	
		0077	;	
014A	AF	0078	REENTRY	XOR A
014B	D36A	0079		OUT (HACTL),A
014D	CD3803	0080		CALL CRLF
0150	C303E0	0081		JP EXIT
		0082	;	
		0083	;	
0153	7C	0084	PADDR	LD A,H
0154	CD5203	0085		CALL PACC
0157	7D	0086		LD A,L
0158	C32F03	0087		JP PASP
		0088	;	
		0089	;	
		0090	;	
015B	50524F4D 30382020 5620312E 30	0091	MSG1	DEFM 'PROM08 V 1.0'
0168	0A0D	0092		DEFW ODOAH
016A	03	0093		DEFB 03H
016B	52454144 5920544F 2050524F 4752414D 20412050 524F4D20 28592F4E 293F	0094	MSG4	DEFM 'READY TO PROGRAM A PROM (Y/N)?'
0189	03	0095		DEFB 03H
018A	4D454D20 53544152 542C4D45 4D20454E 442C5052 4F4D2053 54415254 3A20	0096	MSG5	DEFM 'MEM START, MEM END, PROM START: '
01A8	03	0097		DEFB 03H
01A9	52454144 5920544F	0098	MSG6	DEFM 'READY TO READ A PROM (Y/N)?'

ADDR	CODE	STMT	SOURCE	STATEMENT
	20524541			
	44204120			
	50524F4D			
	2028592F			
	4E293F			
01C4	03	0099	DEFB	03H
		0100	;	
		0101	;	
		0102	;	
		0103	;	
		0104	;	PROM READ SEQUENCE
		0105	;	
		0106	;	
		0107	;	
01C5	CD3803	0108	READ2	CALL CRLF ; READ PROM INTO MEMORY
01C8	21A901	0109		LD HL,MSG6
01CB	CD4703	0110		CALL PTXT ; "READY TO READ (Y/N)?"
01CE	CD3901	0111		CALL YORN
01D1	D21501	0112		JP NC,PPG10A ; NO, THEN PROGRAM
01D4	CD3803	0113	READ0	CALL CRLF
01D7	218A01	0114		LD HL,MSG5
01DA	CD4703	0115		CALL PTXT ; "MEM START, MEM END, PROM START"
01DD	CDBE03	0116		CALL SCAN ; INPUT PARAMETERS
01E0	3A1F04	0117		LD A,(NXCHR)
01E3	FE2E	0118		CP '.'
01E5	28ED	0119		JR Z,READ0-\$
01E7	3E40	0120		LD A,40H
01E9	D36A	0121		OUT (HACTL),A ; TURN ON PROM VOLTAGE
01EB	2A2004	0122		LD HL,(OPR1) ; MEM START ADDR
01EE	ED5B2404	0123		LD DE,(OPR3) ; PROM START ADDR
01F2	CDDA02	0124	READ1	CALL PREAD ; READ A BYTE OF PROM
01F5	77	0125		LD (HL),A ; SAVE BYTE IN MEM
01F6	CD0102	0126		CALL ADCMP ; INC ADDR & COMPARE
01F9	20F7	0127		JR NZ,READ1-\$; IF NOT END, THEN JUMP
01FB	AF	0128		XOR A
01FC	D36A	0129		OUT (HACTL),A
01FE	C31501	0130		JP PPG10A ; WHEN COMPLETE, RETURN
		0131	;	
		0132	;	
		0133	;	
		0134	;	
		0135	;	INC HL & DE THEN COMPARE HL WITH (IX)
		0136	;	EXIT WITH ZERO FLAG SET IF EQUAL
		0137	;	
		0138	;	
0201	13	0139	ADCMP	INC DE ; INC PROM ADDR
0202	7A	0140		LD A,D
0203	E607	0141		AND 7H ; MAX PROM ADDR=7FF
0205	57	0142		LD D,A
0206	23	0143		INC HL
0207	D5	0144		PUSH DE
0208	E5	0145		PUSH HL ; SAVE HL & DE
0209	ED5B2204	0146		LD DE,(OPR2) ; GET RAM END ADDR
020D	13	0147		INC DE
020E	A7	0148		AND A
020F	ED52	0149		SBC HL,DE ; COMPARE
0211	E1	0150		POP HL
0212	D1	0151		POP DE

ADDR	CODE	STMT	SOURCE STATEMENT
0213	C9	0152	RET
		0153	;
		0154	;
		0155	;
		0156	;
		0157	; PROGRAMMING SEQUENCE
		0158	;
		0159	;
0214	2A2004	0160	PROG LD HL,(OPR1) ; MEM START ADDRESS
0217	ED5B2404	0161	LD DE,(OPR3) ; PROM START ADDRESS
021B	CDDA02	0162	PROG1 CALL PREAD ; READ A BYTE (PROM)
021E	FEFF	0163	CP OFFH ; CHECK FOR ERASED BYTES
0220	2007	0164	JR NZ,PROG2-\$; IF NOT ERASED JUMP
0222	CD0102	0165	PRG1A CALL ADCMP ; INC ADDR & CHECK FOR END
0225	20F4	0166	JR NZ,PROG1-\$; LOOP UNTIL END
0227	1821	0167	JR PROG3-\$; GO PROGRAM PROM
0229	E5	0168	PROG2 PUSH HL
022A	210803	0169	LD HL,MSG7
022D	CD4703	0170	CALL PTXT ; "NOT ERASED"
0230	E1	0171	POP HL
0231	DB68	0172	IN A,(DATA)
0233	CDE902	0173	CALL PRTER ; PRINT BAD LOCATION
0236	CDAC03	0174	CALL ECHO ; READ FROM CONSOLE
0239	FE2E	0175	CP '.' ; CHECK FOR PERIOD
023B	2806	0176	JR Z,PRG2A-\$; EXIT
023D	FE43	0177	CP 'C' ; CHECK FOR 'C'
023F	2809	0178	JR Z,PROG3-\$; CONTINUE IF 'C' ENTERED
0241	18DF	0179	JR PRG1A-\$; IF NOT ,PRINT NEXT ERROR
0243	3E00	0180	PRG2A LD A,0
0245	D36A	0181	OUT (HACTL),A
0247	C30901	0182	JP PPG0
024A	CD3803	0183	PROG3 CALL CRLF
024D	0665	0184	LD B,LOOPS+1
024F	2A2004	0185	PROG4 LD HL,(OPR1) ; SET MEM START ADDR
0252	ED5B2404	0186	LD DE,(OPR3) ; SET PROM START ADDR
0256	05	0187	DEC B
0257	2818	0188	JR Z,PROG6-\$
0259	C5	0189	PROG5 PUSH BC
025A	CDB502	0190	CALL PPROG ; PROGRAM ONE BYTE
025D	C1	0191	POP BC
025E	CD0004	0192	CALL CONST
0261	2807	0193	JR Z,PROG5A-\$
0263	CD0804	0194	CALL CONIN
0266	FE2E	0195	CP '.'
0268	2843	0196	JR Z,PROG9-\$
026A	CD0102	0197	PROG5A CALL ADCMP ; INC ADDR & CHECK END
026D	20EA	0198	JR NZ,PROG5-\$; FINISH THIS LOOP
026F	18DE	0199	JR PROG4-\$
0271	3E00	0200	PROG6 LD A,0
0273	D36A	0201	OUT (HACTL),A ; TURN OFF ALL CONTROL LINES
0275	1614	0202	LD D,20
0277	AF	0203	PROG6B XOR A
0278	CDD202	0204	CALL DELAY
027B	15	0205	DEC D
027C	20F9	0206	JR NZ,PROG6B-\$
027E	2A2004	0207	PROG6A LD HL,(OPR1) ; MEM START ADDR
0281	ED5B2404	0208	LD DE,(OPR3) ; PROM START ADDR
0285	CDDA02	0209	PROG7 CALL PREAD ; READ PROM DATA

ADDR	CODE	STMT	SOURCE	STATEMENT
0288	BE	0210		CP (HL)
0289	2007	0211		JR NZ,PROG8-\$; IF NOT MATCH,JUMP
028B	CD0102	0212	PRG7A	CALL ADCMP ; CHECK FOR END
028E	20F5	0213		JR NZ,PROG7-\$
0290	18B1	0214		JR PRG2A-\$; EXIT,COMPLETE
0292	F5	0215	PROG8	PUSH AF
0293	E5	0216		PUSH HL
0294	211503	0217		LD HL,MSG8
0297	CD4703	0218		CALL PTXT ; "BAD LOCATION"
029A	E1	0219		POP HL
029B	CD0004	0220		CALL CONST
029E	2807	0221		JR Z,PROG8A-\$
02A0	CD0804	0222		CALL CONIN
02A3	FE2E	0223		CP '.'
02A5	289C	0224		JR Z,PRG2A-\$
02A7	F1	0225	PROG8A	POP AF
02A8	CDE902	0226		CALL PRTER ; PRINT ERROR MESSAGE
02AB	18DE	0227		JR PRG7A-\$; PRINT ALL BAD LOCATIONS
02AD	211F03	0228	PROG9	LD HL,MSG9
02B0	CD4703	0229		CALL PTXT ; ABORTED
02B3	188E	0230		JR PRG2A-\$
		0231		;
		0232		;
		0233		;
		0234		; PULSE ONE PROM LOCATION
		0235		;
		0236		;
02B5	CDE502	0237	PPROG	CALL PADD ; LATCH LSB OF ADDRESS
02B8	7A	0238		LD A,D
02B9	F680	0239		OR 080H
02BB	D36A	0240		OUT (HACTL),A
02BD	7E	0241		LD A,(HL) ; FETCH DATA
02BE	D368	0242		OUT (DATA),A ; OUTPUT DATA
02C0	7A	0243		LD A,D
02C1	F6C0	0244		OR OCOH ; TURN ON PROG PULSE
02C3	D36A	0245		OUT (HACTL),A
02C5	3E01	0246		LD A,1
02C7	CDD202	0247		CALL DELAY ; 1 MILLISEC DELAY
02CA	7A	0248		LD A,D
02CB	F680	0249		OR 080H
02CD	E687	0250		AND 087H ; PROG PULSE OFF
02CF	D36A	0251		OUT (HACTL),A
02D1	C9	0252		RET
		0253		;
02D2	06BF	0254	DELAY	LD B,191
02D4	10FE	0255	DELAY 1	DJNZ DELAY1-\$
02D6	3D	0256		DEC A
02D7	20F9	0257		JR NZ,DELAY-\$
02D9	C9	0258		RET
		0259		;
		0260		;
		0261		;
		0262		;
		0263		; READ ONE LOCATION OF PROM
		0264		;
		0265		;
02DA	CDE502	0266	PREAD	CALL PADD ; LATCH LSB OF ADDRESS
02DD	7A	0267		LD A,D

ADDR	CODE	STMT	SOURCE	STATEMENT
02DE	E607	0268		AND 7
02E0	D36A	0269		OUT (HACTL),A
02E2	DB68	0270		IN A,(DATA)
02E4	C9	0271		RET
		0272		;
		0273		;
		0274		; LATCH LOWER 8 BITS OF PROM ADDRESS
		0275		;
		0276		;
02E5	7B	0277	PADD	LD A,E
02E6	D369	0278		OUT (LADD),A
02E8	C9	0279		RET
		0280		;
		0281		;
		0282		;
		0283		;
		0284		;
		0285		; PRINTS ADDR/BAD DATA/GOOD DATA
		0286		; OF ERROR LOCATION
		0287		;
		0288		;
02E9	47	0289	PRTER	LD B,A
02EA	7A	0290		LD A,D
02EB	CD5203	0291		CALL PACC ; PRINT MSB OF ADDR
02EE	7B	0292		LD A,E
02EF	CD5203	0293		CALL PACC ; PRINT LSB OF ADDR
02F2	0E20	0294		LD C,20H
02F4	CD1304	0295		CALL CONOUT
02F7	78	0296		LD A,B
02F8	CD5203	0297		CALL PACC ; PRINT BAD DATA
02FB	0E20	0298		LD C,20H
02FD	CD1304	0299		CALL CONOUT
0300	7E	0300		LD A,(HL)
0301	CD5203	0301		CALL PACC ; PRINT GOOD DATA
0304	CD3803	0302		CALL CRLF
0307	C9	0303		RET
		0304		;
		0305		;
		0306		;
		0307		;
0308	4E4F5420 45524153 45442020	0308	MSG7	DEFM 'NOT ERASED '
0314	03	0309		DEFB 03H
0315	42414420 4C4F4320 20	0310	MSG8	DEFM 'BAD LOC '
031E	03	0311		DEFB 03H
031F	41424F52 544544	0312	MSG9	DEFM 'ABORTED'
0326	03	0313		DEFB 03H
		0314		;
		0315		;
		0316		;
		0317		; TWDG-READ AND CONVERT 2 DIGITS TO BINARY
		0318		;
		0319		;
		0320		;

ADDR CODE

STMT SOURCE STATEMENT

```
0321 ;
0322 ;
0323 ;
0324 ; ASCII TO BINARY CONVERSION
0325 ;
0326 ;
0327 D630 0327 ASBIN SUB 030H
0329 FE0A 0328 CP 10
032B F8 0329 RET M
032C D607 0330 SUB 7
032E C9 0331 RET
0332 ;
0333 ;
0334 ;
0335 ;
0336 ;
```

ADDR	CODE	STMT	SOURCE	STATEMENT
		0338	;	
		0339	;	
032F	C5	0340	PASP	PUSH BC
0330	CD5203	0341		CALL PACC
0333	CD4203	0342		CALL SPACE
0336	C1	0343		POP BC
0337	C9	0344		RET
		0345	;	
		0346	;	
		0347	;	
0338	0E0D	0348	CRLF	LD C,ODH
033A	CD1304	0349		CALL CONOUT
033D	0E0A	0350		LD C,0AH
033F	C31304	0351		JP CONOUT
		0352	;	
		0353	;	
0342	0E20	0354	SPACE	LD C,' '
0344	C31304	0355		JP CONOUT
		0356	;	
		0357	;	
		0358	;	
		0359	;	
		0360	;	
		0361	;	PRINT TEXT
		0362	;	
		0363	;	
0347	7E	0364	PTXT	LD A,(HL) ; FETCH A BYTE
0348	FE03	0365		CP 3
034A	C8	0366		RET Z
034B	4F	0367		LD C,A
034C	CD1304	0368		CALL CONOUT
034F	23	0369		INC HL
0350	18F5	0370		JR PTXT
		0371	;	
		0372	;	
		0373	;	PRINT ACCUMULATOR
		0374	;	
		0375	;	
0352	F5	0376	PACC	PUSH AF
0353	0F	0377		RRCA
0354	0F	0378		RRCA
0355	0F	0379		RRCA
0356	0F	0380		RRCA
0357	CD5B03	0381		CALL PRVAL
035A	F1	0382		POP AF
		0383	;	
		0384	;	
035B	E60F	0385	PRVAL	AND OFH
035D	C690	0386		ADD A,90H
035F	27	0387		DAA
0360	CE40	0388		ADC A,40H
0362	27	0389		DAA
0363	4F	0390		LD C,A
0364	C31304	0391		JP CONOUT ; PRINT IT
		0392	;	
		0393	;	
		0394	;	

ADDR	CODE	STMT	SOURCE	STATEMENT
		0395	;	
		0396	;	CHECK FOR VALID HEX CHARACTER
		0397	;	
		0398	;	
0367	FE30	0399	AORN	CP '0'
0369	380E	0400		JR C,AORN2 ; JUMP IF < 30H
036B	FE3A	0401		CP '9'+1
036D	3808	0402		JR C,AORN1 ; JUMP IF < 3AH
036F	FE40	0403		CP 'A'-1
0371	3806	0404		JR C,AORN2 ; JUMP IF < 'A'
0373	FE47	0405		CP 'F'+1
0375	3002	0406		JR NC,AORN2 ; JUMP IF < 'G'
0377	AF	0407	AORN1	XOR A
0378	C9	0408		RET ; VALID DATA RET
0379	AF	0409	AORN2	XOR A
037A	3C	0410		INC A
037B	C9	0411		RET ; NOT HEX CHAR
		0412	;	
		0413	;	
		0414	;	CHECK FOR TERMINATOR
		0415	;	
		0416	;	SPACE, COMMA, OR CARRIAGE RETURN
		0417	;	
		0418	;	
037C	FE20	0419	TERMCK	CP ' '
037E	C8	0420		RET Z
037F	FE2C	0421		CP ','
0381	C8	0422		RET Z
0382	FE2E	0423		CP '. '
0384	2803	0424		JR Z,TCHKO
0386	FE0D	0425		CP ODH
0388	C0	0426		RET NZ
0389	C5	0427	TCHKO	PUSH BC
038A	CD3803	0428		CALL CRLF
038D	C1	0429		POP BC
038E	AF	0430		XOR A
038F	C9	0431		RET
		0432	;	
		0433	;	
		0434	;	SCAN FOR OPERAND FROM KEYBOARD
		0435	;	
		0436	;	
		0437	;	EXIT WITH DATA IN HL, AND TERMINATOR
		0438	;	IN C. IF VALID DATA, RETURN WITH
		0439	;	ZERO FLAG SET. B CONTAINS # OF CHARACTERS ENTERED.
		0440	;	
		0441	;	
0390	210000	0442	KEYIN	LD HL,0
0393	45	0443		LD B,L
0394	CDAC03	0444	KEY1	CALL ECHO
0397	04	0445		INC B ; INC CHAR COUNT
0398	CD7C03	0446		CALL TERMCK
039B	C8	0447		RET Z ; IF TERMINATOR, RETURN
039C	CD6703	0448		CALL AORN ; VALID DATA CHECK
039F	C0	0449		RET NZ ; IF NOT RETURN
03A0	79	0450		LD A,C
03A1	CD2703	0451		CALL ASBIN ; CONVERT TO BINARY
03A4	29	0452		ADD HL,HL

ADDR	CODE	STMT	SOURCE	STATEMENT
03A5	29	0453		ADD HL,HL
03A6	29	0454		ADD HL,HL
03A7	29	0455		ADD HL,HL ;p SHIFT 4 BITS
03A8	85	0456		ADD A,L
03A9	6F	0457		LD L,A
03AA	18E8	0458		JR KEY1
		0459		;
		0460		;
		0461		;
03AC	CD0804	0462	ECHO	CALL CONIN
03AF	4F	0463		LD C,A
03B0	C31304	0464		JP CONOUT
		0465		;
		0466		;
		0467		;
03B3	0E3F	0468	INVCMD	LD C,'?'
03B5	CD1304	0469		CALL CONOUT
03B8	3E2E	0470		LD A,'.'
03BA	321F04	0471		LD (NXCHR),A
03BD	C9	0472		RET
		0473		;
		0474		;
		0475		;
		0476		;
		0477		;SCAN -- SCANS THE OPERANDS INPUT FROM THE
		0478		; CONSOLE. THE OPERANDS ARE SAVED IN OP1,
		0479		; AS THE ARGUMENTS TO EXECUTE THEIR COMMANDS.
		0480		;
03BE	AF	0481	SCAN	XOR A ;INITIALIZE
03BF	212004	0482		LD HL,OPRS ;CLEAR OPERANDS
03C2	E5	0483		PUSH HL
03C3	DDE1	0484		POP IX ;IX=POINTER TO OPERANDS
03C5	77	0485		LD (HL),A
03C6	010900	0486		LD BC,9
03C9	112104	0487		LD DE,OPRS+1
03CC	EDB0	0488		LDIR
03CE	321E04	0489		LD (OPCNT),A
03D1	CD9003	0490	SCAN1	CALL KEYIN ; GET ONE OPERAND
03D4	C2B303	0491		JP NZ,INVCMD
03D7	79	0492		LD A,C
03D8	321F04	0493		LD (NXCHR),A
03DB	FE20	0494		CP ','
03DD	2806	0495		JR Z,SCAN2-\$
03DF	FE2C	0496		CP ','
03E1	2802	0497		JR Z,SCAN2-\$
03E3	05	0498		DEC B
03E4	C8	0499		RET Z ; IF NO DATA, RET
03E5	DD7500	0500	SCAN2	LD (IX),L
03E8	DD7401	0501		LD (IX+1),H
03EB	3A1E04	0502		LD A,(OPCNT)
03EE	3C	0503		INC A
03EF	321E04	0504		LD (OPCNT),A
03F2	DD23	0505		INC IX
03F4	DD23	0506		INC IX ; POINT TO NEXT OPR
03F6	79	0507		LD A,C
03F7	FE20	0508		CP ','
03F9	28D6	0509		JR Z,SCAN1-\$
03FB	FE2C	0510		CP ','

ADDR	CODE	STMT	SOURCE	STATEMENT	
03FD	28D2	0511		JR	Z,SCAN1-\$
03FF	C9	0512		RET	
		0513		;	
		0514		;	
		0515		;	
0400	DB00	0516	CONST	IN	A,(SSTAT) ; CONSOLE STATUS
0402	E602	0517		AND	RXRMSK
0404	C8	0518		RET	Z
0405	3EFF	0519		LD	A,OFFH
0407	C9	0520		RET	
		0521		;	
0408	CD0004	0522	CONIN	CALL	CONST ; CONSOLE INPUT
040B	CA0804	0523		JP	Z,CONIN
040E	DB01	0524		IN	A,(SDATA)
0410	E67F	0525		AND	7FH
0412	C9	0526		RET	
		0527		;	
0413	DB00	0528	CONOUT	IN	A,(SSTAT) ; CONSOLE OUTPUT
0415	E604	0529		AND	TXRMSK ; TX BFR EMPTY
0417	CA1304	0530		JP	Z,CONOUT
041A	79	0531		LD	A,C
041B	D301	0532		OUT	(SDATA),A
041D	C9	0533		RET	
		0534		;	
>041E		0535	OPCNT	DEFS	1
>041F		0536	NXCHR	DEFS	1
>0420		0537	OPRS	DEFS	6
>0420		0538	OPR1	EQU	OPRS
>0422		0539	OPR2	EQU	OPRS+2
>0424		0540	OPR3	EQU	OPRS+4
>0520		0541	SPVAL	EQU	OPRS+100H
		0542		;	
		0543		;	
		0544		;	

ADDR CODE STMT SOURCE STATEMENT

CROSS REFERENCE LISTING

SYMBOL VALUE TYPE STMT STATEMENT REFERENCES

SYMBOL	VALUE	TYPE	STMT	STATEMENT	REFERENCES
ADCMP	0201		0139	0212	0197 0165 0126
AORN	0367		0399	0448	
AORN1	0377		0407	0402	
AORN2	0379		0409	0406	0404 0400
ASBIN	0327		0327	0451	
CONIN	0408		0522	0523	0462 0222 0194
CONOUT	0413		0528	0530	0469 0464 0391 0368 0355 0351 0349 0299 0295
+					
CONST	0400		0516	0522	0220 0192
CRLF	0338		0348	0428	0302 0183 0113 0108 0080 0053 0048 0046
DATA	0068		0019	0270	0242 0172
DELAY	02D2		0254	0257	0247 0204
DELAY1	02D4		0255	0255	
ECHO	03AC		0462	0444	0174 0066
EXIT	E003		0017	0081	
HACTL	006A		0021	0269	0251 0245 0240 0201 0181 0129 0121 0079 0045
+					
INVCMD	03B3		0468	0491	
KEY1	0394		0444	0458	
KEYIN	0390		0442	0490	
LADD	0069		0020	0278	
LOOPS	0064		0023	0184	
MSG1	015B		0091	0041	
MSG4	016B		0094	0049	
MSG5	018A		0096	0114	0054
MSG6	01A9		0098	0109	
MSG7	0308		0308	0169	
MSG8	0315		0310	0217	
MSG9	031F		0312	0228	
NXCHR	041F		0536	0493	0471 0117 0057
OPCNT	041E		0535	0504	0502 0489
OPR1	0420		0538	0207	0185 0160 0122
OPR2	0422		0539	0146	
OPR3	0424		0540	0208	0186 0161 0123
OPRS	0420		0537	0541	0540 0539 0538 0487 0482
PACC	0352		0376	0341	0301 0297 0293 0291 0085
PADD	02E5		0277	0266	0237
PADDR	0153		0084		
PASP	032F		0340	0087	
PPG	0100		0040		
PPG0	0109		0043	0182	
PPG10A	0115		0048	0130	0112
PPG11	0123		0053	0059	
PPROG	02B5		0237	0190	
PREAD	02DA		0266	0209	0162 0124
PRG1A	0222		0165	0179	
PRG2A	0243		0180	0230	0224 0214 0176
PRG7A	028B		0212	0227	
PROG	0214		0160	0060	
PROG1	021B		0162	0166	
PROG2	0229		0168	0164	
PROG3	024A		0183	0178	0167
PROG4	024F		0185	0199	
PROG5	0259		0189	0198	

ADDR	CODE	STMT	SOURCE	STATEMENT
PROG5A	026A	0197		0193
PROG6	0271	0200		0188
PROG6A	027E	0207		
PROG6B	0277	0203		0206
PROG7	0285	0209		0213
PROG8	0292	0215		0211
PROG8A	02A7	0225		0221
PROG9	02AD	0228		0196
PRTER	02E9	0289		0226 0173
PRVAL	035B	0385		0381
PTXT	0347	0364		0370 0229 0218 0170 0115 0110 0055 0050 0042
RBOUT	0000	0022		
READ0	01D4	0113		0119
READ1	01F2	0124		0127
READ2	01C5	0108		0047
RENTY	014A	0078		0068 0052
RXRMSK	0002	0015		0517
SCAN	03BE	0481		0116 0056
SCAN1	03D1	0490		0511 0509
SCAN2	03E5	0500		0497 0495
SDATA	0001	0013		0532 0524
SPACE	0342	0354		0342
SPVAL	0520	0541		0043 0040
SSTAT	0000	0014		0528 0516
TBASE	0100	0018		0032
TCHKO	0389	0427		0424
TERMCK	037C	0419		0446
TXRMSK	0004	0016		0529
YORN	0139	0066		0111 0073 0051

ERRORS=0000

```

ADDR  CODE      STMT SOURCE STATEMENT

0001          NAME      PROM16
0002 ;
0003 ;
0004 ; 2716/2758 PROM PROGRAMMER CONTROL PROGRAM
0005 ; 8/20/79
0006 ;
0007 ; VERSION 1.01  PROM VERSION (NON-DISK I/O)
0008 ;
0009 ;
0010 ; SYSTEM EQUATES
0011 ;
0012 ;
>0001 0013 SDATA  DEFL    01H      ; VIDEO DATA PORT
>0000 0014 SSTAT  DEFL    00H      ; VIDEO STATUS PORT
>0002 0015 RXRMSK DEFL    02H      ; RX RDY MASK
>0004 0016 TXRMSK DEFL    04H      ; TX RDY MASK
>E003 0017 EXIT   EQU     0E003H
>0100 0018 TBASE  EQU     100H      ; BASE OF TRANSIENT AREA
>0068 0019 DATA  EQU     68H      ; DATA PORT
>0069 0020 LADD   EQU     69H      ; LOW ADDRESS PORT
>006A 0021 HACTL  EQU     6AH      ; HIGH ADDRESS + CONTROL PORT
>0000 0022 RBOUT  EQU     0
0023 ;
0024 ;
0025 ;
0026 ;
0027 ;
0028 ;
0029          PSECT    ABS
0030 ;
>0100 0031          ORG     TBASE    ; START AT 100H
0032 ;
0033 ;
0034 ;
0035 ; PROGRAM PROM ENTRY POINT
0036 ;
0037 ;
0038 ;
0100 312405 0039 PPG     LD      SP,SPVAL
0103 215D01 0040          LD      HL,MSG1
0106 CD4703 0041          CALL   PTXT      ; "PROM16 VERSION 1.0"
0109 312405 0042 PPG0    LD      SP,SPVAL
010C AF      0043          XOR     A          ; CLEAR ACC
010D D36A   0044          OUT    (HACTL),A
010F CD3803 0045          CALL   CRLF
0112 C3D301 0046          JP     READ2      ; JUMP READ
0115 CD3803 0047 PPG10A  CALL   CRLF
0118 217301 0048          LD      HL,MSG4
011B CD4703 0049          CALL   PTXT      ; "READY TO PROGRAM A PROM ?"
011E CD3A01 0050          CALL   YORN
0121 D24C01 0051          JP     NC,REENTRY
0124 CD3803 0052 PPG11  CALL   CRLF
0127 219201 0053          LD      HL,MSG5
012A CD4703 0054          CALL   PTXT      ; "MEM START, MEM END, PROM START"
012D CDC203 0055          CALL   SCAN
0130 3A2304 0056          LD      A,(NXCHR)
0133 FE2E   0057          CP     '.'        ; IF '.' EXIT
0135 28ED   0058          JR     Z,PPG11-$

```

ADDR	CODE	STMT	SOURCE	STATEMENT
0137	C32202	0059		JP PROG
		0060		;
		0061		;
		0062		;
		0063		;
		0064		;
013A	CDB003	0065	YORN	CALL ECHO
013D	FE2E	0066		CP '.'
013F	CA4C01	0067		JP Z, RENTRY
0142	B7	0068		OR A
0143	FE4E	0069		CP 'N'
0145	C8	0070		RET Z
0146	FE59	0071		CP 'Y'
0148	20F0	0072		JR NZ, YORN-\$
014A	37	0073		SCF
014B	C9	0074		RET
		0075		;
		0076		;
014C	AF	0077	RENTY	XOR A
014D	D36A	0078		OUT (HACTL), A
014F	CD3803	0079		CALL CRLF
0152	C303E0	0080		JP EXIT
		0081		;
		0082		;
0155	7C	0083	PADDR	LD A, H
0156	CD5303	0084		CALL PACC
0159	7D	0085		LD A, L
015A	C32F03	0086		JP PASP
		0087		;
		0088		;
		0089		;
015D	50524F4D 31362020 56455253 494F4E20 312E30	0090	MSG1	DEFM 'PROM16 VERSION 1.0'
0170	0A0D	0091		DEFW ODOAH
0172	03	0092		DEFB 03H
0173	52454144 5920544F 2050524F 4752414D 20412050 524F4D20 28592F4E 293F	0093	MSG4	DEFM 'READY TO PROGRAM A PROM (Y/N)?'
0191	03	0094		DEFB 03H
0192	4D454D4F 52592053 54415254 2C4D454D 4F525920 454E442C 50524F4D 20535441 52543A20	0095	MSG5	DEFM 'MEMORY START, MEMORY END, PROM START: '
01B6	03	0096		DEFB 03H
01B7	52454144	0097	MSG6	DEFM 'READY TO READ A PROM (Y/N)?'

ADDR	CODE	STMT	SOURCE	STATEMENT
	5920544F			
	20524541			
	44204120			
	50524F4D			
	2028592F			
	4E293F			
01D2	03	0098	DEFB	03H
		0099	;	
		0100	;	
		0101	;	
		0102	;	
		0103	;	PROM READ SEQUENCE
		0104	;	
		0105	;	
		0106	;	
01D3	CD3803	0107	READ2	CALL CRLF ; READ PROM INTO MEMORY
01D6	21B701	0108		LD HL,MSG6
01D9	CD4703	0109		CALL PTXT ; "READY TO READ (Y/N)?"
01DC	CD3A01	0110		CALL YORN
01DF	D21501	0111		JP NC,PPG10A ; NO, THEN PROGRAM
01E2	CD3803	0112	READ0	CALL CRLF
01E5	219201	0113		LD HL,MSG5
01E8	CD4703	0114		CALL PTXT ; "MEM START, MEM END, PROM START"
01EB	CDC203	0115		CALL SCAN ; INPUT PARAMETERS
01EE	3A2304	0116		LD A,(NXCHR)
01F1	FE2E	0117		CP '.'
01F3	28ED	0118		JR Z,READ0-\$
01F5	3E40	0119		LD A,40H
01F7	D36A	0120		OUT (HACTL),A ; TURN ON PROM VOLTAGE
01F9	2A2404	0121		LD HL,(OPR1) ; MEM START ADDR
01FC	ED5B2804	0122		LD DE,(OPR3) ; PROM START ADDR
0200	CDDA02	0123	READ1	CALL PREAD ; READ A BYTE OF PROM
0203	77	0124		LD (HL),A ; SAVE BYTE IN MEM
0204	CD0F02	0125		CALL ADCMP ; INC ADDR & COMPARE
0207	20F7	0126		JR NZ,READ1-\$; IF NOT END, THEN JUMP
0209	AF	0127		XOR A
020A	D36A	0128		OUT (HACTL),A
020C	C31501	0129		JP PPG10A ; WHEN COMPLETE, RETURN
		0130	;	
		0131	;	
		0132	;	
		0133	;	
		0134	;	INC HL & DE THEN COMPARE HL WITH (IX)
		0135	;	EXIT WITH ZERO FLAG SET IF EQUAL
		0136	;	
		0137	;	
020F	13	0138	ADCMP	INC DE ; INC PROM ADDR
0210	7A	0139		LD A,D
0211	E607	0140		AND 7H ; MAX PROM ADDR=7FF
0213	57	0141		LD D,A
0214	23	0142		INC HL
0215	D5	0143		PUSH DE
0216	E5	0144		PUSH HL ; SAVE HL & DE
0217	ED5B2604	0145		LD DE,(OPR2) ; GET RAM END ADDR
021B	13	0146		INC DE
021C	A7	0147		AND A
021D	ED52	0148		SBC HL,DE ; COMPARE
021F	E1	0149		POP HL

ADDR	CODE	STMT	SOURCE	STATEMENT
0220	D1	0150		POP DE
0221	C9	0151		RET
		0152		;
		0153		;
		0154		;
		0155		;
		0156		PROGRAMMING SEQUENCE
		0157		;
		0158		;
0222	3E40	0159	PROG	LD A,40H
0224	D36A	0160		OUT (HACTL),A ; TURN ON PROM VOLTAGE
0226	2A2404	0161		LD HL,(OPR1) ; MEM START ADDRESS
0229	ED5B2804	0162		LD DE,(OPR3) ; PROM START ADDRESS
022D	CDDA02	0163	PROG1	CALL PREAD ; READ A BYTE (PROM)
0230	FEFF	0164		CP OFFH ; CHECK FOR ERASED BYTES
0232	2007	0165		JR NZ,PROG2-\$; IF NOT ERASED JUMP
0234	CD0F02	0166	PRG1A	CALL ADCMP ; INC ADDR & CHECK FOR END
0237	20F4	0167		JR NZ,PROG1-\$; LOOP UNTIL END
0239	1821	0168		JR PROG3-\$; GO PROGRAM PROM
023B	E5	0169	PROG2	PUSH HL
023C	210803	0170		LD HL,MSG7
023F	CD4703	0171		CALL PTXT ; "NOT ERASED"
0242	E1	0172		POP HL
0243	DB68	0173		IN A,(DATA)
0245	CDE902	0174		CALL PRTER ; PRINT BAD LOCATION
0248	CDB003	0175		CALL ECHO ; READ FROM CONSOLE
024B	FE2E	0176		CP '.' ; CHECK FOR PERIOD
024D	2806	0177		JR Z,PRG2A-\$; EXIT
024F	FE43	0178		CP 'C' ; CHECK FOR 'C'
0251	2809	0179		JR Z,PROG3-\$; CONTINUE IF 'C' ENTERED
0253	18DF	0180		JR PRG1A-\$; IF NOT ,PRINT NEXT ERROR
0255	3E00	0181	PRG2A	LD A,0
0257	D36A	0182		OUT (HACTL),A
0259	C30901	0183		JP PPG0
025C	CD3803	0184	PROG3	CALL CRLF
025F	2A2404	0185	PROG4	LD HL,(OPR1) ; SET MEM START ADDR
0262	ED5B2804	0186		LD DE,(OPR3) ; SET PROM START ADDR
0266	CDB502	0187	PROG5	CALL PPROG ; PROGRAM ONE BYTE
0269	CD0404	0188		CALL CONST
026C	2807	0189		JR Z,PROG5A-\$
026E	CD0C04	0190		CALL CONIN
0271	FE2E	0191		CP '.'
0273	2838	0192		JR Z,PROG9-\$
0275	CD0F02	0193	PROG5A	CALL ADCMP ; INC ADDR & CHECK END
0278	20EC	0194		JR NZ,PROG5-\$; FINISH THIS LOOP
027A	3E00	0195	PROG6	LD A,0
027C	D36A	0196		OUT (HACTL),A ; TURN OFF ALL CONTROL LINES
027E	2A2404	0197	PROG6A	LD HL,(OPR1) ; MEM START ADDR
0281	ED5B2804	0198		LD DE,(OPR3) ; PROM START ADDR
0285	CDDA02	0199	PROG7	CALL PREAD ; READ PROM DATA
0288	BE	0200		CP (HL)
0289	2007	0201		JR NZ,PROG8-\$; IF NOT MATCH, JUMP
028B	CD0F02	0202	PRG7A	CALL ADCMP ; CHECK FOR END
028E	20F5	0203		JR NZ,PROG7-\$
0290	18C3	0204		JR PRG2A-\$; EXIT, COMPLETE
0292	F5	0205	PROG8	PUSH AF
0293	E5	0206		PUSH HL
0294	211503	0207		LD HL,MSG8

ADDR	CODE	STMT	SOURCE	STATEMENT
0297	CD4703	0208		CALL PTXT ; "BAD LOCATION"
029A	E1	0209		POP HL
029B	CD0404	0210		CALL CONST
029E	2807	0211		JR Z,PROG8A-\$
02A0	CD0C04	0212		CALL CONIN
02A3	FE2E	0213		CP '.'
02A5	28AE	0214		JR Z,PRG2A-\$
02A7	F1	0215	PROG8A	POP AF
02A8	CDE902	0216		CALL PRTER ; PRINT ERROR MESSAGE
02AB	18DE	0217		JR PRG7A-\$; PRINT ALL BAD LOCATIONS
02AD	211F03	0218	PROG9	LD HL,MSG9
02B0	CD4703	0219		CALL PTXT ; ABORTED
02B3	18A0	0220		JR PRG2A-\$
		0221		;
		0222		;
		0223		;
		0224		; PULSE ONE PROM LOCATION
		0225		;
		0226		;
02B5	CDE502	0227	PPROG	CALL PADD ; LATCH LSB OF ADDRESS
02B8	7A	0228		LD A,D
02B9	F6C0	0229		OR OCOH
02BB	D36A	0230		OUT (HACTL),A
02BD	7E	0231		LD A,(HL) ; FETCH DATA
02BE	D368	0232		OUT (DATA),A ; OUTPUT DATA
02C0	7A	0233		LD A,D
02C1	F6E0	0234		OR OEOH ; TURN ON PROG PULSE
02C3	D36A	0235		OUT (HACTL),A
02C5	3E32	0236		LD A,50
02C7	CDD202	0237		CALL DELAY ; 50 MILLISEC DELAY
02CA	7A	0238		LD A,D
02CB	F6C0	0239		OR OCOH
02CD	E6C7	0240		AND OC7H ; PROG PULSE OFF
02CF	D36A	0241		OUT (HACTL),A
02D1	C9	0242		RET
		0243		;
02D2	06BF	0244	DELAY	LD B,191
02D4	10FE	0245	DELAY1	DJNZ DELAY1-\$
02D6	3D	0246		DEC A
02D7	20F9	0247		JR NZ,DELAY-\$
02D9	C9	0248		RET
		0249		;
		0250		;
		0251		;
		0252		;
		0253		; READ ONE LOCATION OF PROM
		0254		;
		0255		;
02DA	CDE502	0256	PREAD	CALL PADD ; LATCH LSB OF ADDRESS
02DD	7A	0257		LD A,D
02DE	F640	0258		OR 40H
02E0	D36A	0259		OUT (HACTL),A
02E2	DB68	0260		IN A,(DATA)
02E4	C9	0261		RET
		0262		;
		0263		;
		0264		; LATCH LOWER 8 BITS OF PROM ADDRESS
		0265		;

ADDR	CODE	STMT	SOURCE	STATEMENT
		0266	;	
02E5	7B	0267	PADD	LD A,E
02E6	D369	0268		OUT (LADD),A
02E8	C9	0269		RET
		0270	;	
		0271	;	
		0272	;	
		0273	;	
		0274	;	
		0275	;	PRINTS ADDR/BAD DATA/GOOD DATA
		0276	;	OF ERROR LOCATION
		0277	;	
		0278	;	
02E9	47	0279	PRTER	LD B,A
02EA	7A	0280		LD A,D
02EB	CD5303	0281		CALL PACC ; PRINT MSB OF ADDR
02EE	7B	0282		LD A,E
02EF	CD5303	0283		CALL PACC ; PRINT LSB OF ADDR
02F2	0E20	0284		LD C,20H
02F4	CD1704	0285		CALL CONOUT
02F7	78	0286		LD A,B
02F8	CD5303	0287		CALL PACC ; PRINT BAD DATA
02FB	0E20	0288		LD C,20H
02FD	CD1704	0289		CALL CONOUT
0300	7E	0290		LD A,(HL)
0301	CD5303	0291		CALL PACC ; PRINT GOOD DATA
0304	CD3803	0292		CALL CRLF
0307	C9	0293		RET
		0294	;	
		0295	;	
		0296	;	
		0297	;	
0308	4E4F5420 45524153 45442020	0298	MSG7	DEFM 'NOT ERASED '
0314	03	0299		DEFB 03H
0315	42414420 4C4F4320 20	0300	MSG8	DEFM 'BAD LOC '
031E	03	0301		DEFB 03H
031F	41424F52 544544	0302	MSG9	DEFM 'ABORTED'
0326	03	0303		DEFB 03H
		0304	;	
		0305	;	
		0306	;	
		0307	;	TWDG-READ AND CONVERT 2 DIGITS TO BINARY
		0308	;	
		0309	;	
		0310	;	
		0311	;	
		0312	;	
		0313	;	
		0314	;	ASCII TO BINARY CONVERSION
		0315	;	
		0316	;	
0327	D630	0317	ASBIN	SUB 030H
0329	FE0A	0318		CP 10

ADDR CODE STMT SOURCE STATEMENT

032B	F8	0319	RET	M
032C	D607	0320	SUB	7
032E	C9	0321	RET	
		0322	;	
		0323	;	
		0324	;	
		0325	;	
		0326	;	

ADDR	CODE	STMT	SOURCE	STATEMENT
		0328	;	
		0329	;	
032F	C5	0330	PASP	PUSH BC
0330	CD5303	0331		CALL PACC
0333	CD4203	0332		CALL SPACE
0336	C1	0333		POP BC
0337	C9	0334		RET
		0335	;	
		0336	;	
		0337	;	
0338	OE0D	0338	CRLF	LD C,ODH
033A	CD1704	0339		CALL CONOUT
033D	OE0A	0340		LD C,0AH
033F	C31704	0341		JP CONOUT
		0342	;	
		0343	;	
0342	OE20	0344	SPACE	LD C,' '
0344	C31704	0345		JP CONOUT
		0346	;	
		0347	;	
		0348	;	
		0349	;	
		0350	;	
		0351	;	PRINT TEXT
		0352	;	
		0353	;	
0347	7E	0354	PTXT	LD A,(HL) ; FETCH A BYTE
0348	FE03	0355		CP 3
034A	C8	0356		RET Z
034B	4F	0357		LD C,A
034C	CD1704	0358		CALL CONOUT
034F	23	0359		INC HL
0350	C34703	0360		JP PTXT
		0361	;	
		0362	;	
		0363	;	PRINT ACCUMULATOR
		0364	;	
		0365	;	
0353	F5	0366	PACC	PUSH AF
0354	0F	0367		RRCA
0355	0F	0368		RRCA
0356	0F	0369		RRCA
0357	0F	0370		RRCA
0358	CD5C03	0371		CALL PRVAL
035B	F1	0372		POP AF
		0373	;	
		0374	;	
035C	E60F	0375	PRVAL	AND OFH
035E	C690	0376		ADD A,90H
0360	27	0377		DAA
0361	CE40	0378		ADC A,40H
0363	27	0379		DAA
0364	4F	0380		LD C,A
0365	C31704	0381		JP CONOUT ; PRINT IT
		0382	;	
		0383	;	
		0384	;	

ADDR	CODE	STMT	SOURCE	STATEMENT
		0385	;	
		0386	;	CHECK FOR VALID HEX CHARACTER
		0387	;	
		0388	;	
0368	FE30	0389	AORN	CP '0'
036A	380F	0390		JR C,AORN2 ; JUMR IF < 30H
036C	FE3A	0391		CP '9'+1
036E	3809	0392		JR C,AORN1 ; JUMP IF < 3AH
0370	FE40	0393		CP 'A'-1
0372	3807	0394		JR C,AORN2 ; JUMP IF < 'A'
0374	FE47	0395		CP 'F'+1
0376	D27B03	0396		JP NC,AORN2 ; JUMP IF < 'G'
0379	AF	0397	AORN1	XOR A
037A	C9	0398		RET ; VALID DATA RET
037B	AF	0399	AORN2	XOR A
037C	3C	0400		INC A
037D	C9	0401		RET ; NOT HEX CHAR
		0402	;	
		0403	;	
		0404	;	CHECK FOR TERMINATOR
		0405	;	
		0406	;	SPACE, COMMA, OR CARRIAGE RETURN
		0407	;	
		0408	;	
037E	FE20	0409	TERMCK	CP ' '
0380	C8	0410		RET Z
0381	FE2C	0411		CP ', '
0383	C8	0412		RET Z
0384	FE2E	0413		CP '. '
0386	CA8C03	0414		JP Z,TCHKO
0389	FE0D	0415		CP ODH
038B	C0	0416		RET NZ
038C	C5	0417	TCHKO	PUSH BC
038D	CD3803	0418		CALL CRLF
0390	C1	0419		POP BC
0391	AF	0420		XOR A
0392	C9	0421		RET
		0422	;	
		0423	;	
		0424	;	SCAN FOR OPERAND FROM KEYBOARD
		0425	;	
		0426	;	
		0427	;	EXIT WITH DATA IN HL, AND TERMINATOR
		0428	;	IN C. IF VALID DATA, RETURN WITH
		0429	;	ZERO FLAG SET. B CONTAINS # OF CHARACTERS ENTERED.
		0430	;	
		0431	;	
0393	210000	0432	KEYIN	LD HL,0
0396	45	0433		LD B,L
0397	CDB003	0434	KEY1	CALL ECHO
039A	04	0435		INC B ; INC CHAR COUNT
039B	CD7E03	0436		CALL TERMCK
039E	C8	0437		RET Z ; IF TERMINATOR, RETURN
039F	CD6803	0438		CALL AORN ; VALID DATA CHECK
03A2	C0	0439		RET NZ ; IF NOT RETURN
03A3	79	0440		LD A,C
03A4	CD2703	0441		CALL ASBIN ; CONVERT TO BINARY
03A7	29	0442		ADD HL,HL

ADDR	CODE	STMT	SOURCE	STATEMENT
03A8	29	0443		ADD HL,HL
03A9	29	0444		ADD HL,HL
03AA	29	0445		ADD HL,HL ;p SHIFT 4 BITS
03AB	85	0446		ADD A,L
03AC	6F	0447		LD L,A
03AD	C39703	0448		JP KEY1
		0449		;
		0450		;
		0451		;
03B0	CD0C04	0452	ECHO	CALL CONIN
03B3	4F	0453		LD C,A
03B4	C31704	0454		JP CONOUT
		0455		;
		0456		;
		0457		;
03B7	0E3F	0458	INVCMD	LD C,'?'
03B9	CD1704	0459		CALL CONOUT
03BC	3E2E	0460		LD A,'.'
03BE	322304	0461		LD (NXCHR),A
03C1	C9	0462		RET
		0463		;
		0464		;
		0465		;
		0466		;
		0467		;SCAN -- SCANS THE OPERANDS INPUT FROM THE
		0468		; CONSOLE. THE OPERANDS ARE SAVED IN OP1,
		0469		; AS THE ARGUMENTS TO EXECUTE THEIR COMMANDS.
		0470		;
03C2	AF	0471	SCAN	XOR A ;INITIALIZE
03C3	212404	0472		LD HL,OPRS ;CLEAR OPERANDS
03C6	E5	0473		PUSH HL
03C7	DDE1	0474		POP IX ;IX=POINTER TO OPERANDS
03C9	77	0475		LD (HL),A
03CA	010900	0476		LD BC,9
03CD	112504	0477		LD DE,OPRS+1
03D0	EDB0	0478		LDIR
03D2	322204	0479		LD (OPCNT),A
03D5	CD9303	0480	SCAN1	CALL KEYIN ; GET ONE OPERAND
03D8	C2B703	0481		JP NZ,INVCMD
03DB	79	0482		LD A,C
03DC	322304	0483		LD (NXCHR),A
03DF	FE20	0484		CP ','
03E1	2806	0485		JR Z,SCAN2-\$
03E3	FE2C	0486		CP ','
03E5	2802	0487		JR Z,SCAN2-\$
03E7	05	0488		DEC B
03E8	C8	0489		RET Z ; IF NO DATA, RET
03E9	DD7500	0490	SCAN2	LD (IX),L
03EC	DD7401	0491		LD (IX+1),H
03EF	3A2204	0492		LD A,(OPCNT)
03F2	3C	0493		INC A
03F3	322204	0494		LD (OPCNT),A
03F6	DD23	0495		INC IX
03F8	DD23	0496		INC IX ; POINT TO NEXT OPR
03FA	79	0497		LD A,C
03FB	FE20	0498		CP ','
03FD	28D6	0499		JR Z,SCAN1-\$
03FF	FE2C	0500		CP ','

ADDR	CODE	STMT	SOURCE	STATEMENT	
0401	28D2	0501		JR	Z,SCAN1-\$
0403	C9	0502		RET	
		0503		;	
		0504		;	
0404	DB00	0505	CONST	IN	A,(SSTAT) ; CONSOLE STATUS
0406	E602	0506		AND	RXRMSK
0408	C8	0507		RET	Z
0409	3EFF	0508		LD	A,OFFH
040B	C9	0509		RET	
		0510		;	
040C	CD0404	0511	CONIN	CALL	CONST ; CONSOLE INPUT
040F	CA0C04	0512		JP	Z,CONIN
0412	DB01	0513		IN	A,(SDATA)
0414	E67F	0514		AND	7FH
0416	C9	0515		RET	
		0516		;	
0417	DB00	0517	CONOUT	IN	A,(SSTAT) ; CONSOLE OUTPUT
0419	E604	0518		AND	TXRMSK ; TX BFR EMPTY
041B	CA1704	0519		JP	Z,CONOUT
041E	79	0520		LD	A,C
041F	D301	0521		OUT	(SDATA),A
0421	C9	0522		RET	
		0523		;	
		0524		;	
		0525		;	
>0422		0526	OPCNT	DEFS	1
>0423		0527	NXCHR	DEFS	1
>0424		0528	OPRS	DEFS	6
>0424		0529	OPR1	EQU	OPRS
>0426		0530	OPR2	EQU	OPRS+2
>0428		0531	OPR3	EQU	OPRS+4
>0524		0532	SPVAL	EQU	OPRS+100H
		0533		;	
		0534		;	
		0535		;	

ADDR	CODE	STMT	SOURCE	STATEMENT
------	------	------	--------	-----------

CROSS REFERENCE LISTING

SYMBOL	VALUE	TYPE	STMT	STATEMENT REFERENCES
ADCMP	020F		0138	0202 0193 0166 0125
AORN	0368		0389	0438
AORN1	0379		0397	0392
AORN2	037B		0399	0396 0394 0390
ASBIN	0327		0317	0441
CONIN	040C		0511	0512 0452 0212 0190
CONOUT	0417		0517	0519 0459 0454 0381 0358 0345 0341 0339 0289 0285
+				
CONST	0404		0505	0511 0210 0188
CRLF	0338		0338	0418 0292 0184 0112 0107 0079 0052 0047 0045
DATA	0068		0019	0260 0232 0173
DELAY	02D2		0244	0247 0237
DELAY1	02D4		0245	0245
ECHO	03B0		0452	0434 0175 0065
EXIT	E003		0017	0080
HACTL	006A		0021	0259 0241 0235 0230 0196 0182 0160 0128 0120 0078
+				0044
INVCMD	03B7		0458	0481
KEY1	0397		0434	0448
KEYIN	0393		0432	0480
LADD	0069		0020	0268
MSG1	015D		0090	0040
MSG4	0173		0093	0048
MSG5	0192		0095	0113 0053
MSG6	01B7		0097	0108
MSG7	0308		0298	0170
MSG8	0315		0300	0207
MSG9	031F		0302	0218
NXCHR	0423		0527	0483 0461 0116 0056
OPCNT	0422		0526	0494 0492 0479
OPR1	0424		0529	0197 0185 0161 0121
OPR2	0426		0530	0145
OPR3	0428		0531	0198 0186 0162 0122
OPRS	0424		0528	0532 0531 0530 0529 0477 0472
PACC	0353		0366	0331 0291 0287 0283 0281 0084
PADD	02E5		0267	0256 0227
PADDR	0155		0083	
PASP	032F		0330	0086
PPG	0100		0039	
PPG0	0109		0042	0183
PPG10A	0115		0047	0129 0111
PPG11	0124		0052	0058
PPROG	02B5		0227	0187
PREAD	02DA		0256	0199 0163 0123
PRG1A	0234		0166	0180
PRG2A	0255		0181	0220 0214 0204 0177
PRG7A	028B		0202	0217
PROG	0222		0159	0059
PROG1	022D		0163	0167
PROG2	023B		0169	0165
PROG3	025C		0184	0179 0168
PROG4	025F		0185	
PROG5	0266		0187	0194
PROG5A	0275		0193	0189

ADDR	CODE	STMT	SOURCE	STATEMENT
PROG6	027A	0195		
PROG6A	027E	0197		
PROG7	0285	0199	0203	
PROG8	0292	0205	0201	
PROG8A	02A7	0215	0211	
PROG9	02AD	0218	0192	
PRTER	02E9	0279	0216	0174
PRVAL	035C	0375	0371	
PTXT	0347	0354	0360	0219 0208 0171 0114 0109 0054 0049 0041
RBOUT	0000	0022		
READ0	01E2	0112	0118	
READ1	0200	0123	0126	
READ2	01D3	0107	0046	
RENTY	014C	0077	0067	0051
RXRMSK	0002	0015	0506	
SCAN	03C2	0471	0115	0055
SCAN1	03D5	0480	0501	0499
SCAN2	03E9	0490	0487	0485
SDATA	0001	0013	0521	0513
SPACE	0342	0344	0332	
SPVAL	0524	0532	0042	0039
SSTAT	0000	0014	0517	0505
TBASE	0100	0018	0031	
TCHKO	038C	0417	0414	
TERMCK	037E	0409	0436	
TXRMSK	0004	0016	0518	
YORN	013A	0065	0110	0072 0050

ERRORS=0000