TEXAS INSTRUMENTS

Improving Man's Effectiveness Through Electronics

Model 990 Computer

PROM Programming Module Installation and Operation

MANUAL NO. 945258-9701
ORIGINAL ISSUE 1 APRIL 1976
INCLUDES
CHANGE 1................1 OCTOBER 1976

Digital Systems Division

LIST OF EFFECTIVE PAGES

INSERT LATEST CHANGED PAGES DESTROY SUPERSEDED PAGES

Note: The portion of the text affected by the changes is indicated by a vertical bar in the outer margins of the page.

Model 990 Computer PROM Programming Module Installation and Operation (945258-9701)

Original Issue . . . . 1 April 1976
Change 1 . . . . 1 October 1976 (415057)

Total number of pages in this publication is 48 consisting of the following:

PAGE NO.	CHANGE NO.	PAGE NO.	CHANGE NO.	PAGE NO.	CHANGE NO.
Cover	1	2-9 - 2-10	1		
Effective Pages	1	3-1 - 3-2	0		
iii - v	1	3-3 - 3-6	1		
vi	0	3-7 - 3-12	0		
1-1	1	4-1 - 4-4	0		
1-2 - 1-6	0	Alphabetical Index Div.	0		
2-1 - 2-2	0	Index-1 - Index-4	0		
2-3	1	User's Response	1		
2-4	0	Business Reply	1		
2-5 - 2-6	1	Cover Blank	0		
2-7 - 2-8	0	Cover	1		

PREFACE

This manual provides detailed instructions for installing the Texas Instruments PROM Programming Module in conjunction with a Model 990 Computer System. In addition, it contains information required to program the computer to use the line printers, and a description of the units, with specific attention to the controls and indicators. The information is divided into the following four sections:

I General Description – Briefly describes the features and major components of the line printer subsystem to acquaint the reader with this subsystem.

II Installation – Provides step-by-step instructions for unpacking and installing the line printers in either a local or remote location.

III Programming – Presents interfacing information for use by a programmer in designing a service routine to control the line printer's activity.

IV Operation – Describes the controls and indicators of the line printers for operators. It includes line printer information as well as line printer status and maintenance indicators so that the operator can determine that the line printers are operating properly.

If you would like one of Texas Instruments' service personnel to install the programming module for you, please contact your Texas Instruments Sales or Service Offices. These offices can also obtain additional information concerning the module, if you should decide to perform the maintenance on the equipment.

The following related publications contain information about the programming module in conjunction with the 990 computer systems.

Title	Part Number
Model 990/4 Computer System Depot Maintenance Manual. Contains test bench setup and assembly repair instructions which apply to the PROM Programmer Module.	945403-9701
Model 990 Computer 733 ASR/KSR Terminal Installation and Operation. Provides information needed to operate the ASR/KSR Terminal when running the diagnostic program.	945259-9701
Model 990 Computer Diagnostic Handbook. This manual contains instructions for using the diagnostic software for the PROM Programmer Module.	945400-9701

Title	Part Number
Model 990 Computer Family Maintenance Drawings. Contains schematics and assembly drawings for the PROM Programmer Module.	945421-9701 and 945421-9702
Model 990 Computer PROM Programming Module Depot Maintenance. This manual provides information for component repair in the PROM programming module.	945405-9701

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

GENERAL DESCRIPTION

1.1 GENERAL

The PROM Programming Module (Part Number 944924) is a software controlled device that allows creation of custom ROMs from commercially available programmable read-only-memory circuits (PROMs). The module consists of a chassis, one or more personality cards, a CRU interface and control module, and a flat ribbon cable that connects the chassis to the interface module. Figure 1-1 illustrates the module kit. Designed to combine dependable performance with ease of use, the module incorporates the following features:

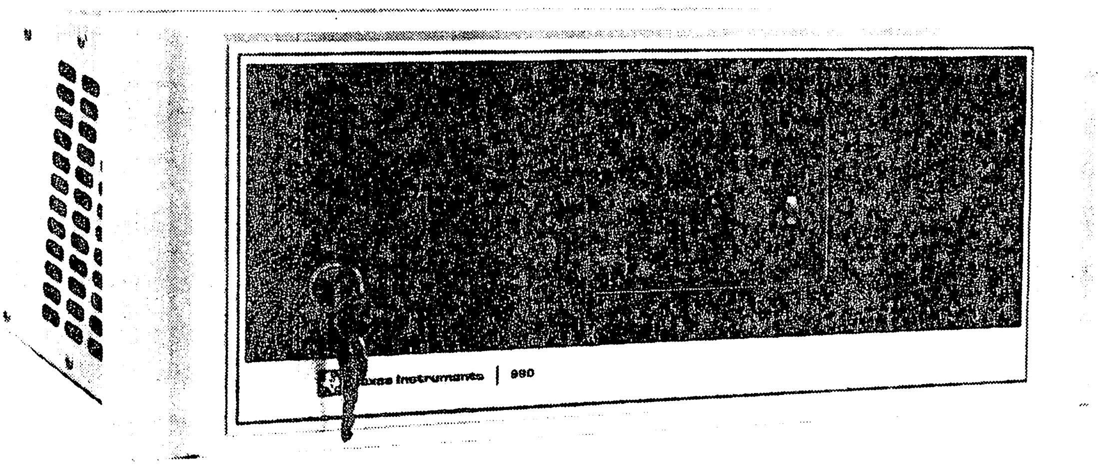
- Interchangeable personality cards for programming either fusible link bi-polar PROMs or Erasable PROMs
- Positive locking sockets hold the PROM firmly in place during programming cycle
- Six software selectable programming pulse widths to adapt to PROM requirements
- Addressing capability for 4K 8-bit PROM words
- Chassis can be mounted in a 19-inch rack or with optional housing can become an attractive, desk-top module.

1.2 SYSTEM TERMINOLOGY

Figure 1-2 illustrates the relationship of the major components of the programming module. The following paragraphs provide an overview of the interaction of these components. More detailed information about the internal operation of the module components can be obtained in The *Model 990 Computer PROM Programming Module Depot Maintenance Manual.*

1.2.1 **COMMUNICATIONS REGISTER UNIT (CRU).** The Communications Register Unit (CRU) is the serial interface of the Texas Instruments Model 990 Computer. Data is transmitted between the controller and the computer through this interface one bit at a time, requiring eight data transfers for each 8-bit ROM word. Control information, such as address and pulse widths for the PROM, are also transmitted to the controller through this interface.

1.2.2 **PROM PROGRAMMER CONTROL CARD.** The control card contains all the control and interface electronics required to operate the programming module in conjunction with a Model 990 Computer. The module normally installs in the bottom front slot of the chassis, but it may occupy any available CRU slot of the computer. Included on the circuit board are an input register, an address register, a data register, a pulse width register, plus control circuitry to coordinate the programming cycle. The input register converts the serial input from the CRU interface into parallel data for loading into one of the other registers or for control information. See Section III of this manual for the disposition of the bits in this register. The address register is a 12-bit register that supplies address information to the PROM to be programmed. The value in this register must be changed each time a new word is entered into the PROM. The data register is an 8-bit register that holds the value to be entered into the ROM word addressed by the address register. The pulse width register is a 6-bit register that holds control bits to select one of the six available pulse widths to be used during the programming cycle. Only one bit of



133162 (990-476-11-7)

Figure 1-1. PROM Programming Module

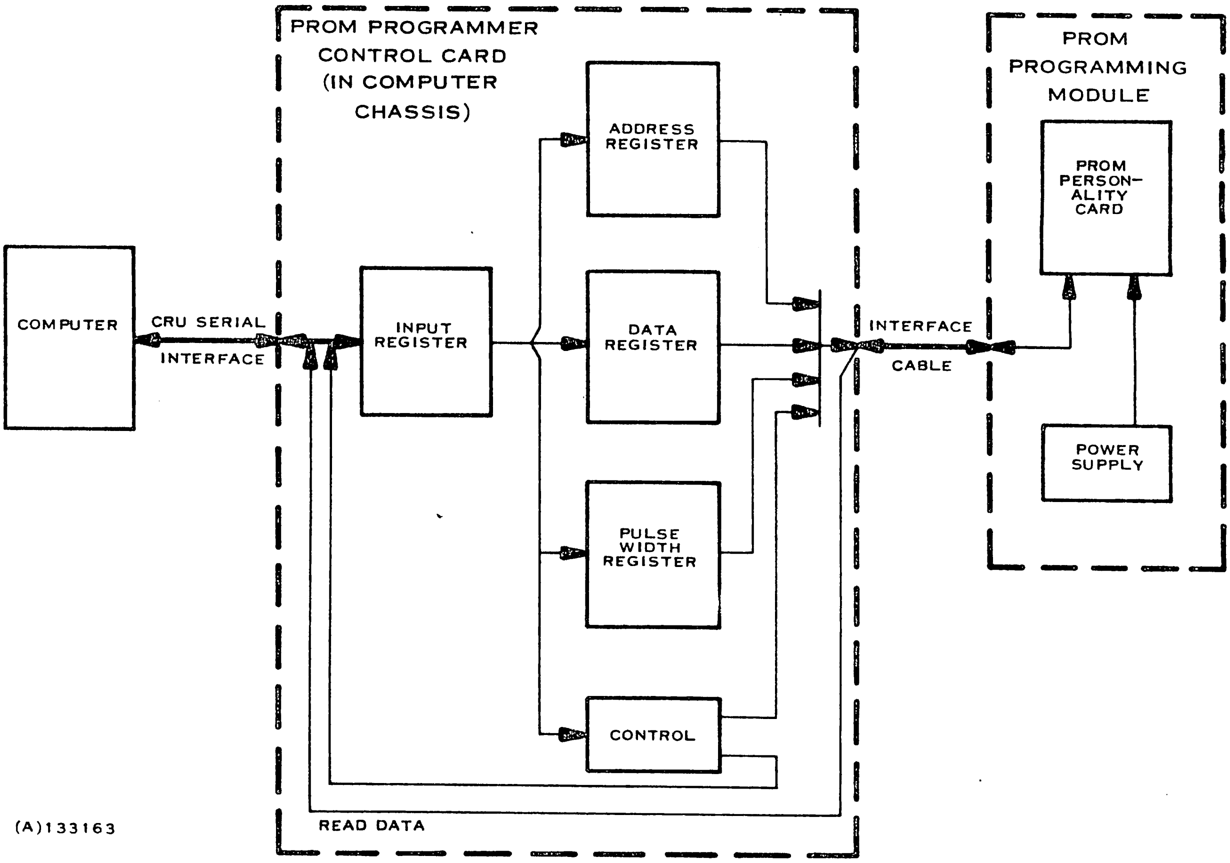


Figure 1-2. PROM Programming Module System Block Diagram

the pulse width register should be set at any one time. Section III of this manual provides the details required to select the desired pulse width.

Control logic on the control card monitors the input control bits in the input register and initiates a programming cycle at the direction of the computer. During the programming cycle, the control logic provides the required gates to apply a voltage to the PROM to program the specified data in the addressed area in the PROM. Feedback logic on the control card allows the computer to sample the contents of the three parameter registers (address, data and pulse width), to read the contents of the addressed ROM location, or to monitor the status of the programming module.

1.2.3 **PROM PROGRAMMING MODULE CHASSIS.** The programming module chassis houses a power supply, plus a personality card. The power supply receives ac input and generates the logic voltage required to power the logic on the personality card, and a voltage for use by the personality card for the programming pulse. The personality card contains sockets for mounting the PROM to be programmed and any required logic for gating data, address, and voltage levels to the proper pin configurations.

Two personality cards are available for use with the programming module. One personality card is used for bipolar PROMs and the other personality card is used for EPROMs. Table 1-1 lists the device types accepted by each personality card together with the characteristics of the devices. The personality cards are contained in a unitized assembly that includes the front panel of the programming module chassis. Replacement of the personality card can be performed quickly and easily.

Table 1-1. PROM Programming Module Programmable Devices

Personality Card	Device Type	Internal Configuration	Initial Logic State of cells	Maximum Duty Cycle	Bits Programmed Each Pulse	Recommended pulse Width Range [1]	Description
Bipolar (946761-0001)							
	SN74188A	32 x 8	0	35%	1	1 to 20 milliseconds	Bipolar, fusible link
	SN74S188	32 x 8	0	35%	1	1 to 20 milliseconds	Bipolar, fusible link
	SN74S288	32 x 8	0	35%	1	1 to 20 milliseconds	Bipolar, fusible link
	SN74S287	256 x 4	1	35%	1	1 to 20 milliseconds	Bipolar, fusible link
	SN74S387	256 x 4	1	35%	1	1 to 20 milliseconds	Bipolar, fusible link
	SN74S470	256 x 8	0	35%	1	1 to 20 milliseconds	Bipolar, fusible link
	SN74S471	256 x 8	0	35%	1	1 to 20 milliseconds	Bipolar, fusible link
	SN74S472	512 x 8	0	35%	1	1 to 20 milliseconds	Bipolar, fusible link
	SN74S473	512 x 8	0	35%	1	1 to 20 milliseconds	Bipolar, fusible link
EPROM (946761-0002)							
	MOS 2704	512 x 8	1	N/A	8 [2]	0.1 to 1.0 milli-seconds [2]	MOS, reprogrammable
	MOS 2708	1024 x 8	1	N/A	8 [2]	0.1 to 1.0 milli-seconds [2]	MOS, reprogrammable

[1] Pulse width ranges specified by device manufacturers

[2] Each PROM word must be pulsed several times. Pulse width determines the number of times the voltage must be applied to a particular PROM word for successful programming. The relationship is:

$$N(t_{pw}) \geqslant 100 \text{ milliseconds}$$

where N is the number of loops and t_{pw} is the pulse width. All words must be addressed in consecutive order.

SECTION II

INSTALLATION

2.1 GENERAL

This section provides information for planning the installation site, unpacking and packing the module, installing the module and its interface card at the site, and ensuring that the module is installed and operating properly. The instructions in this section require a moderate familiarity with cabling technique and use of common handtools, but do not assume any level of expertise in digital electronics.

2.2 SITE REQUIREMENTS

The PROM Programmer may be installed either as a desk-top unit or in a 19-inch rack-mounted configuration. Figures 2-1 and 2-2 illustrate the physical size of each of the configurations. An additional clearance of approximately 3 inches (7.6 cm) should be maintained on all sides of the desk-top unit to allow for cooling air flow. Table 2-1 lists the requirements of the PROM Programmer that affect planning for the installation site.

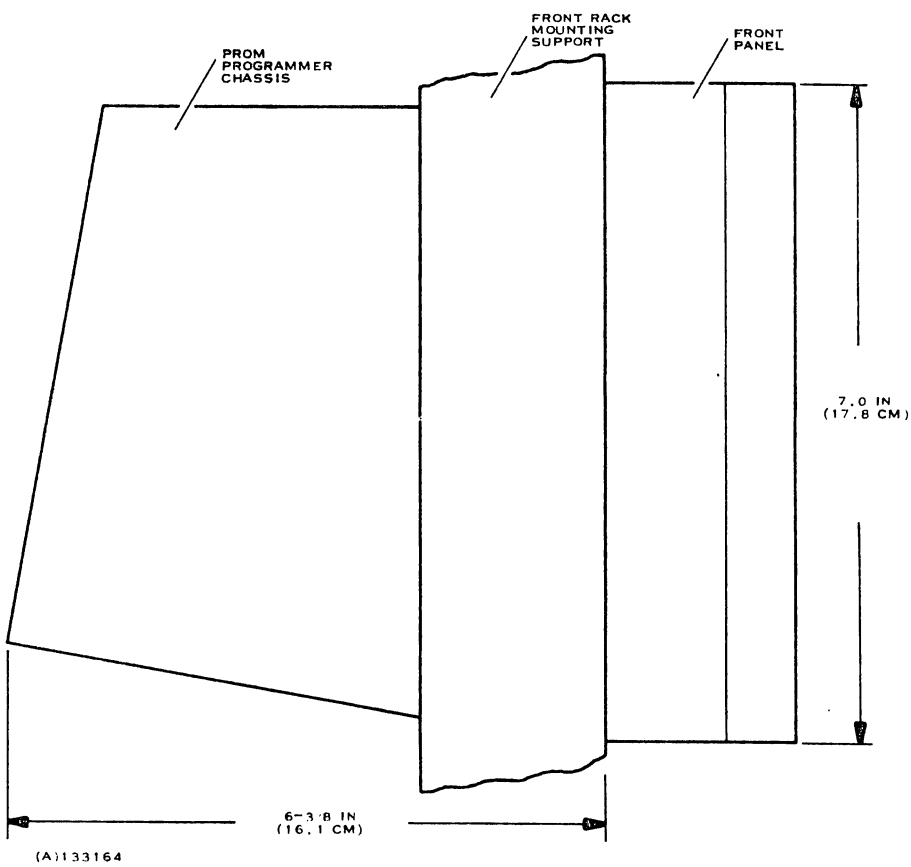


Figure 2-1. PROM Programmer Rack Mounted Dimensions

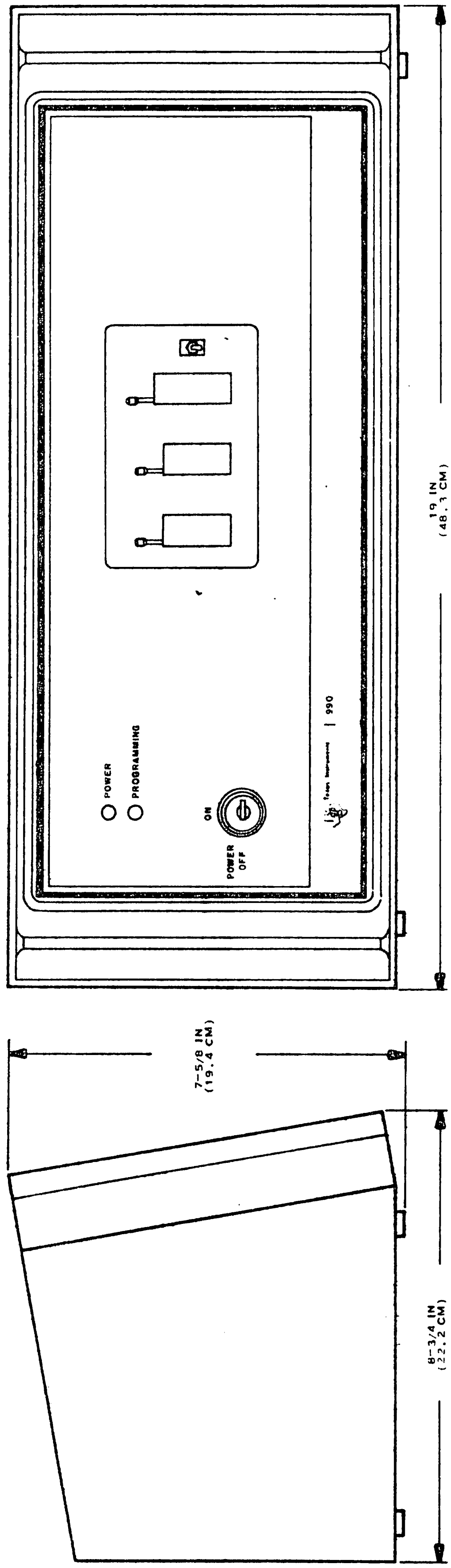


Figure 2-2. PROM Programmer Table Top Cabinet Dimensions

Table 2-1. PROM Programmer Physical Characteristics

Characteristic	Requirement
Power	
Interface Card	0.8 amps (maximum) of +5 vdc logic voltage (supplied by computer chassis)
Programmer	0.6 amps (maximum) of 115 vac, +15%, -10%, or 0.7 amps (maximum) of 100 vac, +15%, -10%, or 0.3 amps (maximum) of 230 vac, +15%, -10%
Temperature	
Operating	32 to 152 degrees F (0 to 65 degrees C)
Storage	-40 to 158 degrees F (-40 to 70 degrees C)
Humidity	0 to 95% without condensation
Cable length	
Power cord	6 feet from Programmer to ac power source
Interface cable	12 feet (3.6 m) from Programmer to interface card in computer chassis

2.3 UNPACKING/PACKING

The PROM Programmer is shipped in a corrugated cardboard container as illustrated in figure 2-3. If included as part of a computer system shipment, the interface board is shipped in the computer chassis. If the PROM Programmer is shipped alone, the interface board is wrapped in bubble-pack and included in the same carton as the PROM Programmer. The programming module is shipped fully assembled with the personality card (if ordered) installed on the front of the PROM Programmer. If a second personality card is included in the shipment, it is wrapped in bubble-pack and included in the same carton as the PROM Programmer. Interface cables are shipped with the personality card.

Upon receipt of the shipping container, inspect it to ensure that there is no physical damage. After this preliminary inspection, perform the following steps to remove the PROM Programmer components from the shipping container and ready them for installation in the computer system. Figure 2-4 illustrates the system components that are packed in the shipping container.

NOTE

Save shipping container and all packing material for use in reshipment.

1. Position the shipping container so that the address label is right-side up.

2. Open the top of the container and remove the packet containing the documentation for the PROM Programming system, the packet containing the interface circuit board (if system is shipped separately), and the packet containing the extra personality card (if ordered).

3. Carefully remove packing material from extra personality card (if extra card is included in shipment), and set personality card in a safe place.

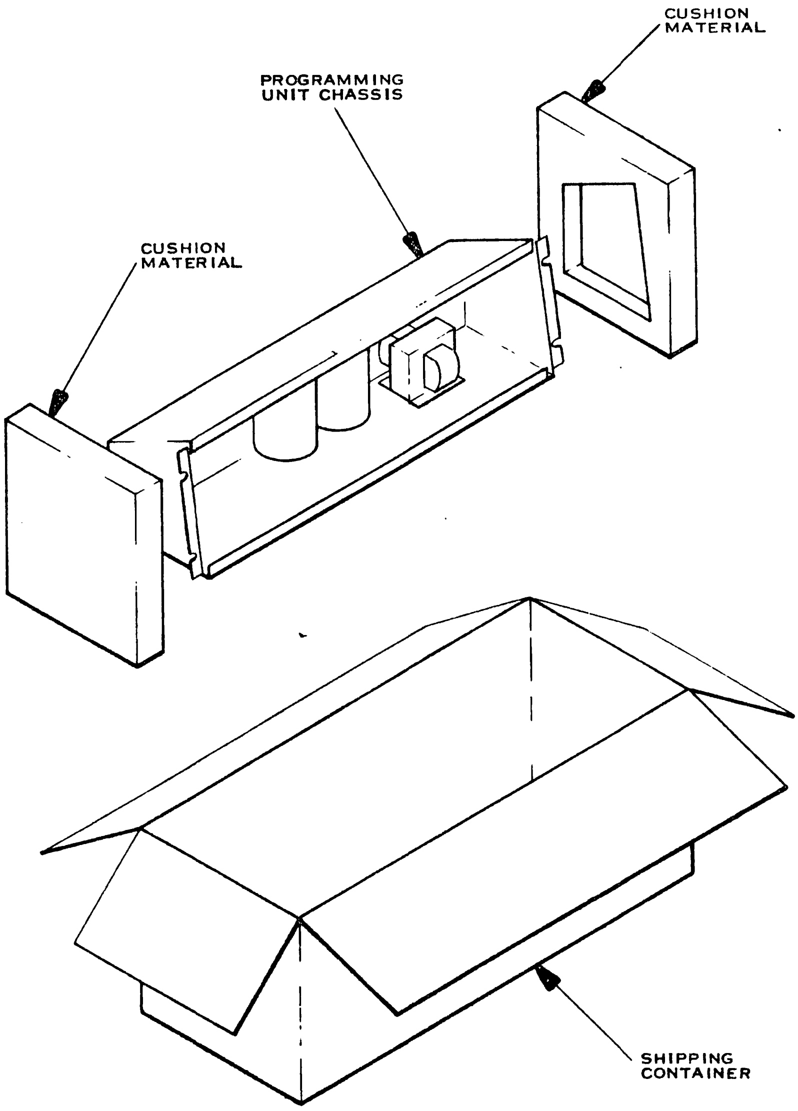


Figure 2-3. PROM Programmer Shipping Container

4. Remove the packing assembly from the shipping carton (packing assembly includes cushioning material and PROM Programmer) and place assembly on a table or desk.

5. Remove cushioning material from PROM Programmer and set programmer on the table.

6. Ensure that no physical damage is apparent to the PROM Programmer, the interface circuit board or the extra personality card.

7. Replace all packing material into the shipping container and store container for future reuse.

To repack the unit for shipment, perform the unpacking procedure in reverse order. Reseal the shipping container with filament reinforced tape.

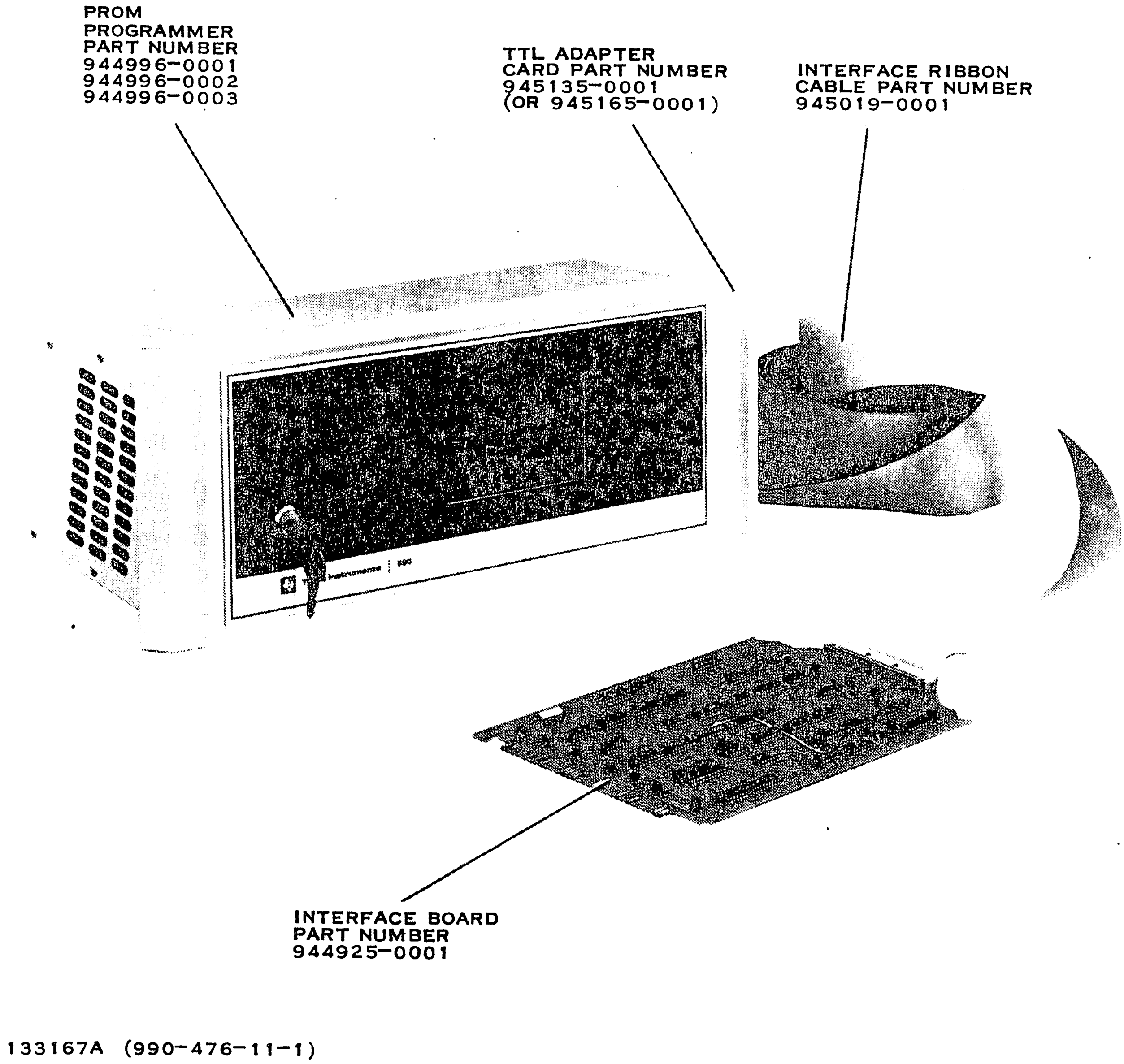


Figure 2-4. PROM Programmer Kit Component

2.4 INSTALLATION OF PROM PROGRAMMER (TABLETOP)

Perform the following steps to install the tabletop model of the PROM Programmer at the site selected for it:

1. Remove front panel/personality card assembly from the programmer by pulling the front panel assembly out from the programmer.

2. Ensure that the personality card is correct for the PROM devices to be used (TTL PROM card has three sockets; EPROM card has one socket).

CAUTION

> Improper alignment of interface cable connector could result in damage to interface module circuits and/or PROM Programmer circuits.

3. Connect the flat ribbon interface cable to the personality module by first aligning the triangle indicator on the cable connector with the matching indicator on the personality module connector, and firmly seating the two connectors together. (Refer to figure 2-5.)

4. Ensure that the logic power and the ac power connectors from the interior of the PROM Programmer are properly connected to the personality card.

5. Replace front panel/personality card assembly on the front of the programmer by first aligning the panel with the sides of the programmer chassis and then pressing on the front panel to mate it to the chassis.

6. Place the programmer in the desired location.

7. Remove the cable tie from the ac power cord at the rear of the programmer. Connect the power cord to the grounded receptacle for applicable electrical power (either 100 vac, 115 vac or 230 vac; 50 or 60 Hz). Do not turn the programming module on at this time.

2.5 INSTALLATION OF PROM PROGRAMMER (RACKMOUNT)

The rackmounted version of the PROM Programmer occupies 7 inches of vertical space in the rack. It is mounted to the rack by two bolts on either side that fit into predrilled holes in the rack. The mounting holes for the programmer are 4 inches apart. The following steps describe the procedure for mounting the programmer in a 19-inch rack:

1. Remove the front panel/personality card assembly from the programmer by pulling the front panel assembly out from the programmer.

2. Disconnect ribbon cable, logic power and ac power connectors from front panel assembly and set front panel assembly aside.

3. Select two mounting holes in the vertical rack on one side that are four inches apart and equally spaced within the 7 inch vertical space reserved for the PROM Programmer. Insert one mounting bolt in each of the two mounting holes. Turn bolts to start threads into holes.

4. Engage slots in mounting flange of programmer on the bolts inserted in *step 3* so that the bolts support the weight of one end of the programmer.

5. Align slots in mounting flange on opposite side of programmer with holes in rack. Insert and tighten two mounting bolts to secure the second side.

6. Tighten the first two bolts to completely secure the programmer to the rack.

7. Ensure that the personality module is correct for the PROM devices to be used (TTL PROM personality module has 3 sockets; EPROM personality module has 1 socket).

CAUTION

Improper alignment of interface cable connector could result in damage to interface module circuits and/or PROM Programmer circuits.

8. Connect the flat ribbon interface cable to the personality module by first aligning the triangle indicator on the cable connector with the matching indicator on the personality module connector, and firmly seating the two connectors together. (Refer to figure 2-5.)

9. Connect the logic power and ac power connectors from the interior of the PROM Programmer to the connectors on the personality card (these connectors were removed in step 2).

10. Replace the front panel/personality card assembly on the front of the programmer by aligning the panel with the sides of the programmer chassis and then pressing on the front panel to mate it to the chassis.

11. Remove the cable tie from the ac power cord at the rear of the programmer. Connect the power cord to a grounded receptacle for applicable electrical power (either 100 Vac, 115 Vac or 230 Vac; 50 or 60 Hz). Do not turn the programming module on at this time.

2.6 CONNECTION TO COMPUTER

The programmer interface module is a plug-in circuit board that serves as an interface between the computer and the PROM Programmer. This module (part number 944925-0001) can be installed in a CRU connector in the computer chassis, or in an expansion chassis connected to the computer. In all cases, the computer and/or the expansion chassis equipment must be installed and operating properly. Refer to the computer hardware reference manual listed in the Preface for installation procedures for this equipment.

2.6.1 LOCATION OF INTERFACE MODULE. The assigned location of the interface module, whether installed in the computer chassis or the expansion chassis, determines the CRU address to which it responds. Conversely, the CRU address which the system software expects determines the chassis location of the interface module. The standard location for the interface module in the computer is slot 13 in the 13-slot chassis and slot 6 in the 6-slot chassis. In both cases the CRU base address is 0020_{16} (front half of the slot) and the interrupt level is 15. The standard configuration for the computer chassis connects this CRU slot to interrupt level 15. If the interface module is installed in any other location than slot 13 or slot 6, then the interrupt jumper plug in the computer chassis must be removed, modified to make the interrupt level compatible with the new address, and then reinstalled. However, TI supplied software for the

PROM Programmer (PROMPG) disables the interrupt on the module (interrupt is generated at completion of one programming cycle), so that allowance for an interrupt is only required if user-supplied software depends on the interrupt. For instructions to modify an interrupt level, refer to either the *Model 990/4 Computer System Hardware Reference Manual,* part number 945251-9701, or the *Model 990/10 Computer System Hardware Reference Manual,* part number 945417-9701.

2.6.2 **INSTALLING THE INTERFACE MODULE.** Once the proper location for the interface module has been determined, the module can be installed in the computer or in the expansion chassis and connected to the flat ribbon cable. Refer to figure 2-5 for an interconnection cabling illustration. To install the interface module and connect the PROM Programmer, perform the following steps.

1. Set the POWER switch on the back of the computer or expansion chassis to the OFF position.

NOTE

> The interface module circuit board has two plastic, pivoted tabs (board ejectors) on one end of the board. This is the outside edge of the board; the opposite edge inserts into the connector in the computer chassis.

2. Insert the interface module circuit board (part number 944925-0001) into the chassis location corresponding to the desired address. Ensure that the component side of the circuit board is facing upward and the slots in the circuit board mate properly with the alignment comb on the backplane connector.

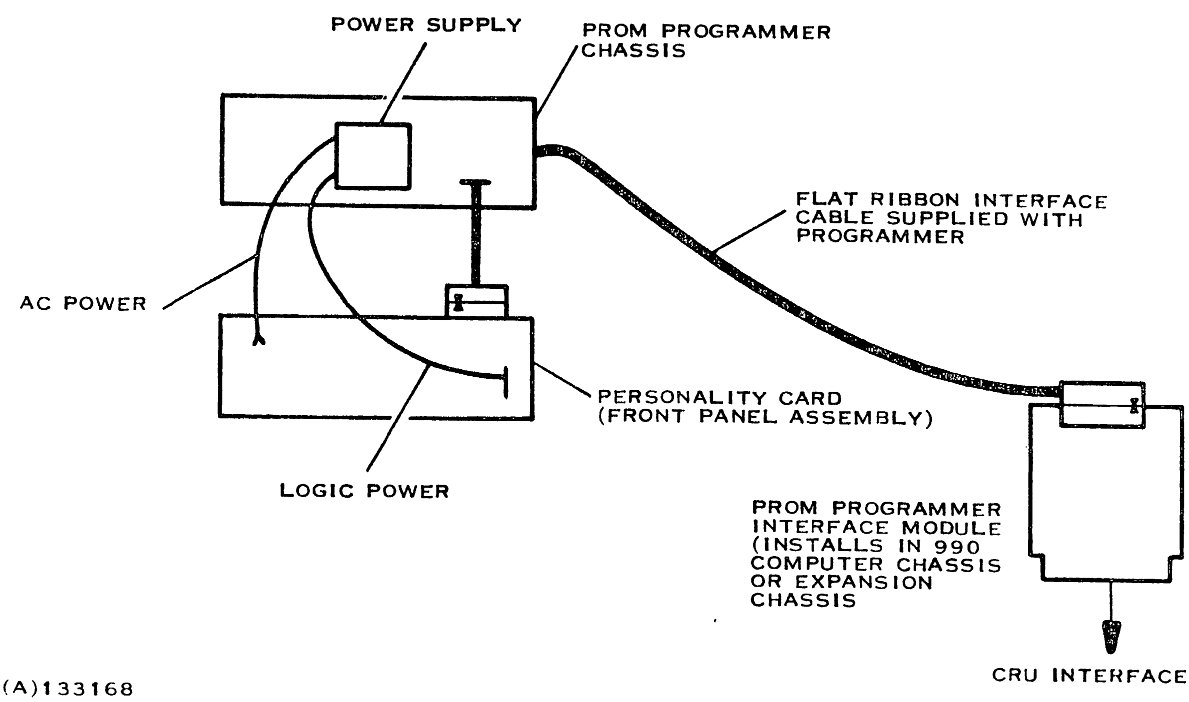


Figure 2-5. PROM Programming System Cabling

CAUTION

Improper alignment of interface cable connector could result in damage to interface module circuits and/or PROM Programmer circuits.

3. Connect the flat ribbon interface cable (part number 945019-0001) to the connector of the interface module. Ensure that the embossed triangle on the cable connector is aligned with the matching triangle on the interface board connector. The cable dresses down from the interface board.

4. Route the cable out through the rear of the computer chassis.

5. Apply power to the computer, expansion chassis and PROM Programmer.

SECTION III

PROGRAMMING

3.1 GENERAL

This section provides information required to generate a program that interfaces a member of the 990 Computer Family to the PROM Programming Module. This section assumes that the user is experienced in using 990 Computer Assembly Language as described in the *Model 990 Computer TMS9900 Microprocessor Assembly Language Programmer's Guide,* part number 943441-9701. In addition, the examples in this section assume that the user has established the format of the data to be stored in a PROM circuit, and that the data is stored in a known area of memory. The information in this section applies only if the user has not elected to use the software interface to the module (PROMPG) included in the Prototyping System Software. For information regarding use of the PROM Programming Module with the Prototyping System, refer to the *Model 990 Computer Prototyping System Operation Guide.*

3.2 CRU INTERFACE

The PROM Programming Module is controlled through manipulation of specified bits of the Communications Register Unit (CRU) interface of the 990 Computer. The module is assigned 16 bit addresses within the CRU. Each bit address can be used to transfer information either to or from the module using the appropriate bit transfer (SBZ, SBO, TB) or register transfer (STCR, LDCR) instructions of the computer. Although the base address of the module varies with the location of the module within the computer chassis, the displacement address assignments remain constant. Figure 3-1 illustrates the bit address assignments for the module during both an input and an output operation. Table 3-1 defines the output bit assignments; table 3-2 defines the input assignments. The programming examples in this section assume that the module interface card is installed in the bottom front chassis location and responds to CRU base address 0020_{16}.

3.3 PROM PROGRAMMING

The software interface between the computer and the module must perform five basic operations to transfer data from a location in computer memory to the desired bit pattern in the PROM circuit. These five operations are:

1. Establish the ROM word address of the area in the PROM device.
2. Transfer the data to be stored in the PROM to the module.
3. Specify the duration of the programming pulse used to implant the data in the PROM.
4. Initiate the programming cycle.
5. Time the programming cycle and delay the next cyle to allow for duty cycle restrictions on a TTL PROM device.

The following paragraphs describe a method for performing these operations. This method is intended for illustrative purposes only, and may not be the best or most efficient means of performing the step for a particular application.

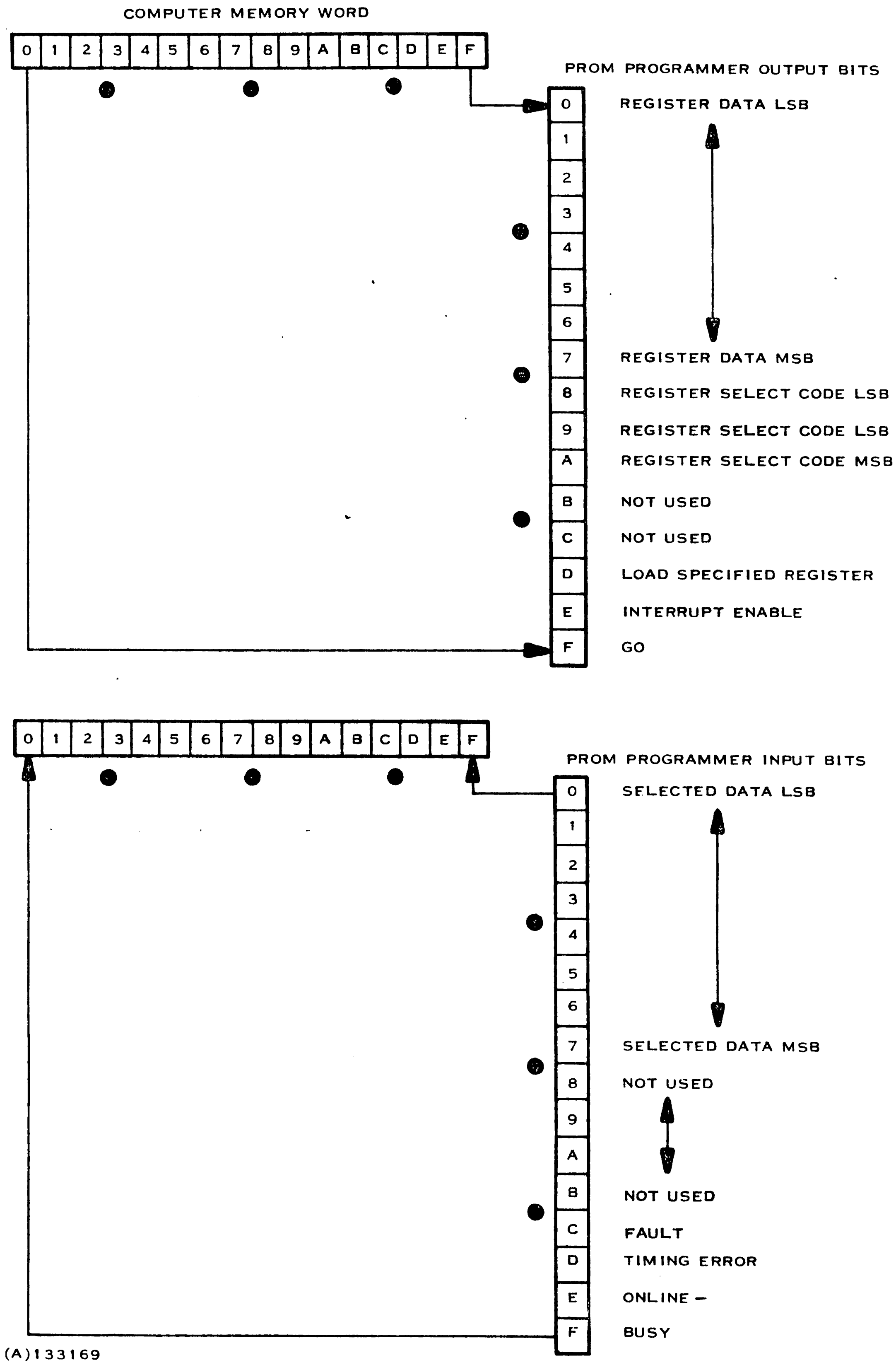


Figure 3-1. CRU Output and Input Bit Assignments

Table 3-1. PROM Programmer CRU Interface Output Bit Assignments

Base Relative Bit Number (Hexadecimal)	Description
00 Through 07	*Register Data:* Eight bits of data to be loaded into either the PROM Address Register, the PROM Data Register, or the Pulse Width Register as specified by the Register Select Code (Output Bits 8 - A).
08 Through 0A	*Register Select Code:* A 3-bit code that specifies the destination register of the data in Output Bits 0 - 7, or the source of data to be supplied to Input Bits 0 - 7. The codes select sources and destinations as follows:

8	9	A	Output Destination	Input Source
0	0	0	PROM Addr 0-3 (MSB)	PROM Addr 0-3 (MSB)
1	0	0	PROM Addr 4-11	PROM Addr 4-11
0	1	0	No Assignment	No Assignment
1	1	0	PROM Data 0-7	PROM Data 0-7
0	0	1	No Assignment	No Assignment
1	0	1	Pulse Width 1-6	Pulse Width 0-5
0	1	1	No Assignment	No Assignment
1	1	1	No Assignment	PROM Read 0-7

Base Relative Bit Number (Hexadecimal)	Description
0B,0C	Not Used
0D	*Load Specified Register:* When this bit is set to a "1", the Register Data from bits 0-7 is transferred to the register specified by the Register Select Code. The selected register load does not always involve the entire Register Data byte.
0E	*Interrupt Enable:* Setting this bit to a "1" enables interrupts. When this bit clears, the interface module is prevented from generating an interrupt to the computer.
0F	*GO:* Setting this bit to a "1" instructs the programming module to begin a programming cycle using the data, address and pulse width currently contained in the interface registers. When one programming cycle is complete, the module generates an interrupt to the computer if the Interrupt Enable bit is not clear.

3.3.1 PROM ADDRESS. The module interface card contains a 12-bit address register for providing address data to the PROM device that is being programmed. Before programming the device, this register must be loaded using the Register Data bits of the CRU interface with the module. Since only eight Register Data bits can be transferred with one CRU operation, loading the address register requires two CRU data transfers: one transfer to load the most significant four bits of the address register, and one transfer to load the least significant eight bits of the address register. Each transfer consists of 3 CRU operations: an LDCR instruction to transfer the address value and register select bits to the module (where it is held in an input register), an SBO instruction to set the Load Specified Register bit, and an SBZ instruction to clear the Load Specified Register bit (bit 13).

Table 3-2. PROM Programmer CRU Interface Input Bit Assignments

Base Relative Bit Number (Hexadecimal)	Description
00 (LSB) Through 07 (MSB)	*Selected Data:* Eight bits of data that provide diagnostic feedback, status indications, or read data from the addressed PROM. Contents of these input bits are determined by the Register Select Code (Output Bits 8-A) from the computer.
08 Through 0B	These bits are not currently assigned
0C	*Fault:* When set, this bit indicates an equipment failure has occurred in the PROM Programmer system. This bit is not currently implemented.
0D	*Timing Error:* When set, this bit indicates that the GO bit (output bit F) was reset during the current PROM programming cycle. Timing error clears at the end of the cycle.
0E	*Online-:* When this bit is a logic zero, it indicates that the interface module is plugged into the computer or expansion chassis and is receiving logic voltage from that chassis.
0F	*Busy:* When set, this bit indicates that the module is currently in the programming mode. When this bit is set, any attempt to clear the GO bit will not end the programming cycle, but will clear the GO bit and set the Timing Error bit.

For example, loading the 12-bit PROM address $A73_{16}$ into the module address register requires the following instruction sequence:

```
ADDRO    BYTE   >01          Initial Register Select Code
ROMAD1   BYTE   >73          Least Significant Byte of
                             ROM Address
ROMAD2   BYTE   >0A          Most Significant Byte of
                             ROM Address
PRMBA    DATA   >20          Module Base Address
START    MOV    @PRMBA,R12   Transfer Base Address to
                             Workspace Register 12
         MOVB   @ROMAD1,R1   Transfer Least Significant Byte
                             of ROM Address to R1
         SRL    R1,8         Shift ROM Address to Least
                             Significant Byte of R1
         MOVB   @ADDRO,R1    Transfer Register Select Code to
                             Most Significant Byte of R1
```

(Listing continued on next page)

```
          LDCR    R1,0          Transfer (R1) to module input
                                register
          SBO     13            Set Load bit; transfers address from
                                input register to address register
                                bits 4-11
          SBZ     13            Clear Load bit
NXTAD     MOVB    @ROMAD2,R1    Transfer Most Significant Byte of
                                ROM Address to R1
          SRL     R1,8          Shift ROM Address to Least
                                Significant Byte of R1
          CLR     R0            Change Register Select Code to 0000₁₆
          MOVB    R0,R1         Transfer Register Select Code to Most
                                Significant Byte of R1
          LDCR    R1,0          Transfer (R1) to module input register
          SBO     13            Set Load bit; transfers address
                                from input register to address
                                register bits 0-3 (bits 0-3 of
                                address byte 0A₁₆ are ignored)
          SBZ     13            Clear Load bit
```

3.3.2 **DATA TRANSFER.** Once the ROM address has been set-up in the module, the data register must be loaded with the byte of data to be stored in the PROM circuit. This sequence is similar to transferring the address, except that the register select code bits are different. The following instruction sequence transfers the value FF_{16} from a location in memory to the module for storage in an EPROM device. Because each EPROM word must be programmed repeatedly, this sequence must be repeated several times for each word. Refer to EPROM data sheets for specific requirements.

```
MEMADD    EQU     $             Establish memory address for data
          DATA    >00FF         Enter data in memory location
                                MEMADD
DATAIN    BYTE    >03           Register Select Code to load
                                data
START     MOV     @MEMADD,R1    Transfer data from memory
                                address MEMADD to R1
          MOVB    @DATAIN,R1    Transfer Select Code (Left byte)
                                to R1 (left byte)
          LDCR    R1,0          Transfer (R1) to module input
                                register
          SBO     13            Set Load bit, transfers data
                                from input register to PROM
                                Data register.
          SBZ     13            Clear Load bit
```

TTL PROMs must be programmed one bit at a time and, therefore, require a masking operation prior to transferring data to the PROM Programmer. Data masking is discussed later in this section.

3.3.3 **PULSE WIDTH.** The duration of the programming pulse required for a particular PROM device is specified by the manufacturer of the device being used. To accomodate a variety of PROM devices, the module allows the user to select one of six available pulse widths to be used during the PROM programming cycle. The selection of pulse width is accomplished by setting one of the bits in the Pulse Width Register of the module. To ensure that the correct bit is set, load all of the bits in the Pulse Width Register each time. Table 3-3 lists the hexadecimal values that can be loaded into the register and the corresponding pulse widths produced.

The instruction sequence to transfer the selected register value to the module is similar to the previous two transfers. Again, the register select code value is different. The following sequence transfers the value to select a pulse width of 2.0 milliseconds to the module:

```
PSWD    EQU     $            Establish memory address for pulse
                             width value
        DATA    >0004        Enter value corresponding to 2.0
                             milliseconds
PWD     BYTE    >0500        Register Select Code for
                             pulse width
START   MOV     @PSWD,R1     Transfer value from memory
                             address PSWD to R1.
        MOVB    @PWD,R1      Transfer select code (left byte)
                             to R1 (left byte)
        LDCR    R1,0         Transfer (R1) to module input
                             register
        SBO     13           Set Load bit; transfers data
                             from input register to Pulse
                             Width Register
        SBZ     13           Clear Load bit.
```

Table 3-3. Programming Pulse Widths

Value in Memory (Hexadecimal)	Pulse Width (Milliseconds)
0001	0.5
0002	1.0
0004	2.0
0008	4.0
0010	8.0
0020	16.0

3.3.4 INITIATE PROGRAMMING CYCLE. Once all of the module registers contain the desired parameters, the programming cycle can begin. Before starting, however, the status of the module should be checked to ensure that it is capable of performing the operation. In addition, the interrupt line from the module should be disabled unless the software interface is designed to handle the interrupt. The following instruction sequence checks module status and then instructs the module to begin the programming cycle:

```
TB      15          Check Online- bit
JEQ     ONLERR      If device is not online (Online- =1),
                    then jump to user defined routine,
                    ONLERR
SBZ     14          Disable interrupts
SBZ     15          Ensure that the Go bit is clear
SBO     15          Set Go bit; starts programming
                    cycle
```

3.3.5 DUTY CYCLE TIMING. TTL PROM devices require a rest period between programming cycles to allow their internal circuits to cool and stabilize. Failure to allow for this rest period will produce unreliable results. EPROM circuits require no extended rest period. The duration of the rest period is determined by the duty cycle parameter specified for the particular device. Duty cycle is expressed as a percentage of the programming cycle time to the total of programming and rest times. A duty cycle of 25 percent, for example, indicates that there must be a rest period 3 times longer than the programming cycle between each programming cycle. The software interface for the module must ensure that the duty cycle requirement is met.

A set of counting loops can ensure that a sufficient rest period is allowed. Begin the first counting loop when the Go bit is set, and then monitor the Busy bit to determine when the programming cycle stops. When Busy clears, the count from the first loop can be multiplied by a factor (determined by the duty cycle parameters) to arrive at the number of counts required to satisfy the rest period. The following sequence satisfies the time keeping requirements to ensure dependable PROM programming results:

```
MAXCT   EQU   65000        Define upper bound for count
                           operation
        CLR   R14          Clear R14 for use as counting
                           register
LOOP    EQU   $            Begin timing loop
        TB    15           Check Busy bit
        JNE   CONTIN       If not busy, exit loop
                           to CONTIN
        INC   R14          Add one to counting register,
                           R14
        CI    R14,MAXCT    Compare count with upper
                           bound
        JGT   HWERR        If R14 exceeds upper bound, jump
                           to error routine HWERR
        JMP   LOOP         Return to start of count
                           loop
CONTIN  EQU   $            Begin rest period time calculation
        SBZ   15           Clear Go bit
```

(Listing continued on next page)

```
DTYCY   EQU   0025          Define a 25% duty cycle
        LI    R11,DTYCY     Transfer duty cycle value
                            to R11
        LI    R0,100        Load a value of 100 into R0
        S     R11,R0        Subtract duty cycle value from 100;
                            remainder stored in R0
        MOV   R0,R1         Right justify remainder in R0 and R1
        CLR   R1
        DIV   R11,R0        Divide value in R0 and R1 by duty cycle
                            value; result stored in R0, remainder
                            in R1
        MOV   R1,R1         Compare remainder to zero
        JEQ   GOON          If remainder is zero, jump to GOON
        INC   R0            If remainder is not zero, add one to
                            R0 to round up the count.
GOON    EQU   $
        MPY   R14,R0        Multiply programming time (R14) by rest time
                            multiplier contained in R0 and R1;
                            answer stored in R0 and R1, right
                            justified.
        CI    R1,0          Compare R1 to zero
        JEQ   OUTLP         If R1=0, jump to next comparison
NXTLP   EQU   $             If R1≠0, begin rest time count loop
        DEC   R1            Subtract 1 from value in R1
        NOP                 No-op instructions to match delay time
        .                   in first loop
        .
        .
        NOP
        JH    NXTLP         If R1 is greater than zero, repeat loop
OUTLP   EQU   $
        CI    R0,0          Compare R0 to zero
        JEQ   OUT           If R0 is zero, jump to exit
                            routine, OUT
        DEC   R0            If R0 is not zero, decrement
                            R0
        JMP   NXTLP         Repeat loop (until R0
                            and R1 are zero)
```

3.4 DATA MASKING

TTL PROM devices must be programmed in a bit-by-bit fashion. Attempts to program more than one bit at a time produce unreliable results. Therefore, when programming the PROM Programmer for TTL PROMs, all but one of the PROM word bits must be masked to the inactive level during a particular programming cycle. This restriction requires four separate programming cycles to program each 4-bit PROM word (or eight cycles to program an 8-bit PROM word). The following instruction sequence illustrates the data manipulation within the computer that is required to program a 4-bit PROM word. The example assumes that the PROM Programmer stores ones in the PROM (initial state of the PROM is all zeros). The example further assumes that all preliminary parameters (address and pulse width) have been previously setup in the programmer, and that the data to be programmed is right-justified in memory location MEMAD1.

```
MASK     EQU    >FFFE       Define mask
MEMAD1   EQU    $           Establish memory address for data
         DATA   >3456       Enter data in memory location
                            MEMAD1
STRING   EQU    4           Define ROM word size
SETUP    EQU    $           Define starting point of setup
                            sequence for programming a
                            PROM word
DATAIN   EQU    >0300       Register select code to load
                            data
         LI     R0,DATAIN   Load register select code into
                            R0
         LDCR   R0,0        Transfer Register Select Code to module
                            input register
         LI     R0,MASK     Transfer FFFE to R0
         MOV    @MEMAD1,R1  Transfer 3456 to R1
         LI     R2,STRING   Transfer ROM word size to R2
         MOV    R1,R3       Copy data in R3
CYCLE    EQU    $           Define start of word programming
                            loop
         SZC    R0,R1       Use mask (R0) to set all but one bit
                            in R1 to zeros (inactive)
         SWPB   R1          Put least significant byte of R1
                            in most significant position
         LDCR   R1,8        Transfer masked data to module input
                            register
         .
         .                  Insert sequences to initiate programming
                            cycle and to allow for duty cycle rest
                            period
         .
         SRC    R0,15       Change mask value to enable next
                            significant bit position
         MOV    R3,R1       Restore correct data to R1
         DEC    R2          Decrement word size counter
         JNE    CYCLE       If word size has not decremented to
                            zero, repeat masking cycle.
NEWAD    EQU    $           If word size has decremented to zero, enter
         .                  segment to select new address and data
         .                  pattern.
         .
```

3.5 READING A ROM

The module can also be used to verify data contained in an existing ROM circuit or in a PROM that has just been programmed with the module. The process is similar to those already described for the programming operation. The address register must be loaded with the address of the ROM word to be read. However, since no data will be transmitted to the ROM, the module data register does not need to be loaded. Also, the register select code bits must all be set to select the read data operation. Addressing a Store CRU operation to the module then transfers the data from the ROM to the specified computer location. From that location it can be read and printed on a printer or CRT for verification. The following instruction sequence illustrates the steps required to read a ROM word into computer memory (at address $A73_{16}$):

```
ADDRO    EQU    >0100         Register Select Code
DOUT     EQU    >0700         Register Select Code for
                              reading ROM
ROMAD1   BYTE   >73           Least Significant Byte of ROM
                              Address
ROMAD2   BYTE   >0A           Most Significant Byte of ROM
                              Address
         MOVB   @ROMAD1,R1    Transfer Least Significant Byte
                              to R1
         SRL    R1,8          Shift address to right half of R1
         LI     R0,ADDRO      Load Register Select Code into
                              R0
         MOVB   R0,R1         Transfer code to left half of R1
         LDCR   R1,0          Transfer (R1) to module input register
         SBO    13            Set Load bit
         SBZ    13            Clear Load bit
NXTAD    MOVB   @ROMAD2,R1    Transfer Most Significant Byte
                              to R1
         SRL    R1,8          Shift address to right half of R1
         CLR    R0            Change Register Select Code to
                              0000 (16)
         MOVB   R0,R1         Transfer Register Select Code
                              to R0
         LDCR   R1,0          Transfer (R1) to module input register
         SBO    13            Set Load bit
         SBZ    13            Clear Load bit
         LI     R1,DOUT       Load Register Select Code for reading
                              ROM into R1
         LDCR   R1,0          Transfer (R1) to module input register
                              (enables ROM data input to CRU)
         STCR   R1,8          Load the 8 ROM data bits from the
                              module into R1
```

3.6 READ MODULE REGISTERS

The three module registers (Address, Data and Pulse Width) can be examined at any time in the program sequence by addressing successive STCR instructions to the module while changing the Register Select Code. The STCR instruction will read the contents of the 8 bits indicated by the Register Select Code and store them in a specified memory area. Therefore, if the Register Select Code already indicates the register that needs to be checked, only an STCR instruction is required. This feature allows each register to be checked immediately after it is loaded merely by using an STCR instruction, since the Register Select Code used to load the register is the same as that used to read from the register. To illustrate the register monitoring process, the following instruction sequence reads the contents of the registers into defined areas of memory so that they can be displayed on a system terminal:

```
ROMAD1   BYTE   00            Memory for least significant byte
                              of address register
ROMAD2   BYTE   00            Memory for most significant byte
                              of address register
```

(Listing continued on next page)

```
ROMDAT   BYTE   00          Memory for module data register
                            contents
PULSWD   BYTE   00          Memory for Pulse Width contents
CHK      EQU    $
         CLR    R1
         LDCR   R1,0        Clear module input register (Register
                            Select Code = 000₂)
         STCR   ROMAD2,8    Store most significant byte of
                            address register in memory
                            ROMAD2
         SBO    8           Change Register Select Code
         STCR   ROMAD1,8    Store least significant byte of
                            address register in memory ROMAD1
         SBO    9           Change Register Select Code
         STCR   ROMDAT,8    Store Data Register contents in
                            memory ROMDAT
         SBZ    9
         SBO    10          Change Register Select Code
         STCR   PULSWD,8    Store Pulse Width value in memory
                            PULSWD
```

SECTION IV

OPERATION

4.1 GENERAL

Operation of the PROM Programming module is controlled by the software interface that is resident in the Model 990 Computer. This interface may be supplied by the user (see Section III of this manual), or it can be ordered from Texas Instruments Incorporated as part of the Prototyping System Software. Because operator interaction with either of these interfaces varies, the operator should consult the documentation for the interface to determine the exact steps required to operate the system. The Prototyping System Software information may be found in the *Model 990 Computer Prototyping System Operation Guide*. This section provides information for using the module regardless of the controlling software interface.

4.2 CONTROLS AND INDICATORS

The module contains two LED indicators and a key switch that are mounted on the front panel of the module. The personality card for bi-polar PROM devices has a toggle switch located to the right of the three PROM sockets; the personality card for EPROM devices has a toggle switch next to its single socket. Figure 4-1 illustrates the position of each of these components; table 4-1 defines their nomenclature and function.

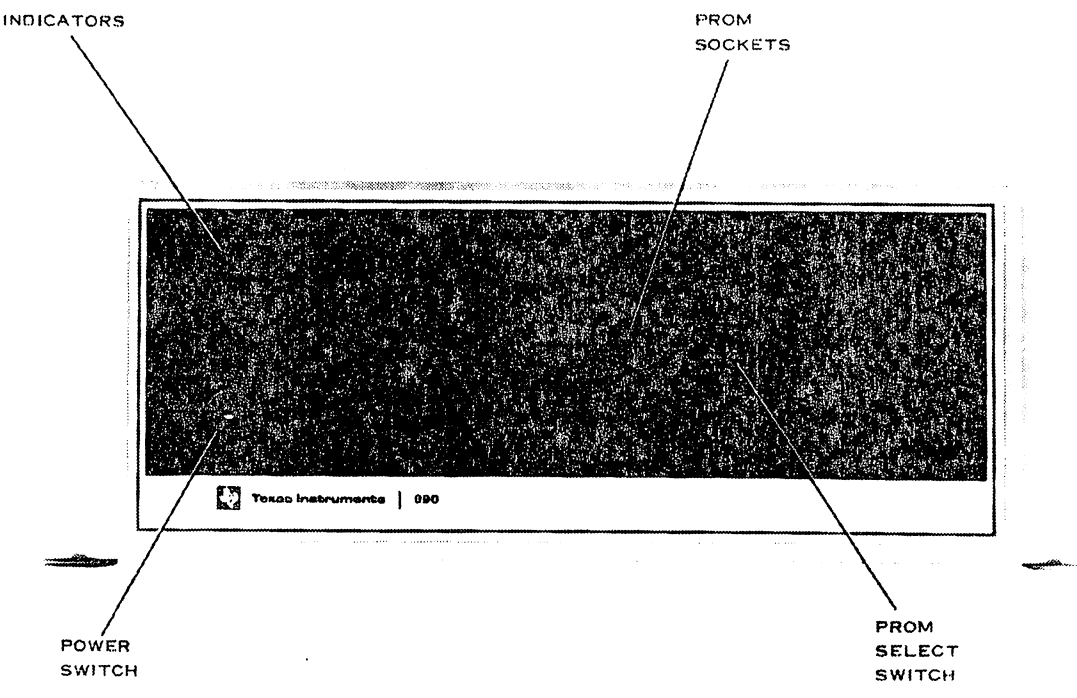


Figure 4-1. PROM Programming Module Front Panel (Bi-polar PROMs)

Table 4-1. PROM Programming Module Operator Components

Nomenclature	Device	Function
POWER	Indicator	Lights to indicate that the module power supply is furnishing logic power to the module.
PROGRAMMING	Indicator	Lights to indicate that the module is executing a programming cycle and is applying the programming pulse to the PROM device sockets.
ON/OFF	Key Switch	Enables and disables ac power to the power supply. When in ON position, ac power is applied to the supply to generate all voltage levels required by the module. When in OFF position, ac power is removed from the supply to disable operation. Removing the key from the switch prevents the position from being changed.
C/D*	Toggle Switch	Modifies function of rightmost PROM socket. When set to C, the socket can be used to program 256 x 8 PROMs, device types 74S470 and 74S471. When set to D, the socket can be used to program 512 x 8 PROMs, device types 74S472 and 74S473.
A*	DIP Socket	Mounts and holds 32 x 8 PROM devices for programming (device types 74188A, 74S188 and 74S288).
B*	DIP Socket	Mounts and holds 256 x 4 PROM devices for programming (device types 74S287 and 74S387).
C/D*	DIP Socket	Mounts and holds 256 or 512 x 8 PROM devices for programming (device types 74S470, 74S471, 74S472 and 74S473).
EPROM**	DIP Socket	Mounts and holds EPROM devices for programming (Intel devices 2704 or 2708).
A/B**	Toggle Switch	Modifies function of EPROM DIP socket. When set to B, the socket can be used to program 1024 x 8 EPROM devices (Intel 2708); when set to A, the socket can be used to program 512 x 8 EPROM (Intel 2704).

*Bi-polar Personality Card
**EPROM Personality Card

4.3 MOUNTING AND REMOVING A PROM

In order to program a PROM, it must be properly installed in the correct socket on the front panel of the PROM Programming Module. Perform the following steps to ensure that the PROM is correctly installed:

1. Ensure that the PROGRAMMING indicator is not lighted. If it is lighted, wait until it goes out before continuing with procedure.

2. Select the socket on the module that corresponds to the device to be programmed. If the wrong personality card is currently installed on module, install the correct personality card using the procedure described later in this section.

3. Position the locking lever of the selected socket so that it points out from the panel. This releases the locking mechanism in the socket.

4. Insert the PROM into the socket with pin 1 of the device in the upper left corner of the socket (next to the locking lever).

5. Reposition the locking lever so that it points up. This locks the PROM securely in place.

6. Ensure that the ON/OFF switch is in the ON position, the POWER indicator is lighted, and that the toggle switch is set properly.

7. Initiate the programming cycle (consult the software interface documentation to determine the required preparation and steps to perform the programming cycle).

8. When the programming cycle is complete, remove the PROM by positioning the locking lever to release the PROM and lift the PROM from the socket.

4.4 EXCHANGING PERSONALITY CARDS

Each personality card comes in a complete front panel assembly for quick and easy exchange to switch from bi-polar ROMs to erasable PROMs (EPROMs). The following procedure describes the steps required to exchange the cards:

NOTE

When the front panel of the module is removed, an interlock switch removes power from the unit to prevent an electrical shock hazard.

1. Grasp the front panel at either side and pull out to remove the panel from the module. The table-top model may have to be held down to remove the front panel.

2. Disconnect the ribbon cable connector, the logic power connector and the ac power connector from the front panel assembly.

3. Connect the ribbon cable connector to the new personality card and front panel assembly. Ensure that the triangle embossed on the cable connector aligns with the triangle embossed on the personality card connector. The triangle indicates the position of pin 1 for each connector.

4. Connect the ac power cable connector to the corresponding connector on the personality card. The shape of the connector pin sleeves prevents them from mating in the wrong orientation. If the connectors will not mate properly, rotate one connector 180 degrees and try again.

5. Connect the logic power connector to the corresponding connector on the personality card.

6. Ensure that the cables are not caught between the panel assembly and the chassis of the module, and press the new front panel assembly in place on the chassis.

ALPHABETICAL INDEX

INTRODUCTION

The following index lists key words and concepts from the subject material of the manual together with the area(s) in the manual that supply major coverage of the listed concept. The numbers along the right side of the listing reference the following manual areas:

- Sections - References to Sections of the manual appear as "Section x" with the symbol x representing any numeric quantity.

- Appendixes - References to Appendixes of the manual appear as "Appendix y" with the symbol y representing any capital letter.

- Paragraphs - References to paragraphs of the manual appear as a series of alphanumeric or numeric characters punctuated with decimal points. Only the first character of the string may be a letter; all subsequent characters are numbers. The first character refers to the section or appendix of the manual in which the paragraph is found.

- Tables - References to tables in the manual are represented by the capital letter T followed immediately by another alphanumeric character (representing the section or appendix of the manual containing the table). The second character is followed by a dash (-) and a number:

  Tx-yy

- Figures - References to figures in the manual are represented by the capital letter F followed immediately by another alphanumeric character (representing the section or appendix of the manual containing the figure). The second character is followed by a dash (-) and a number:

  Fx-yy

- Other entries in the Index - References to other entries in the index are preceded by the word "See" followed by the referenced entry.