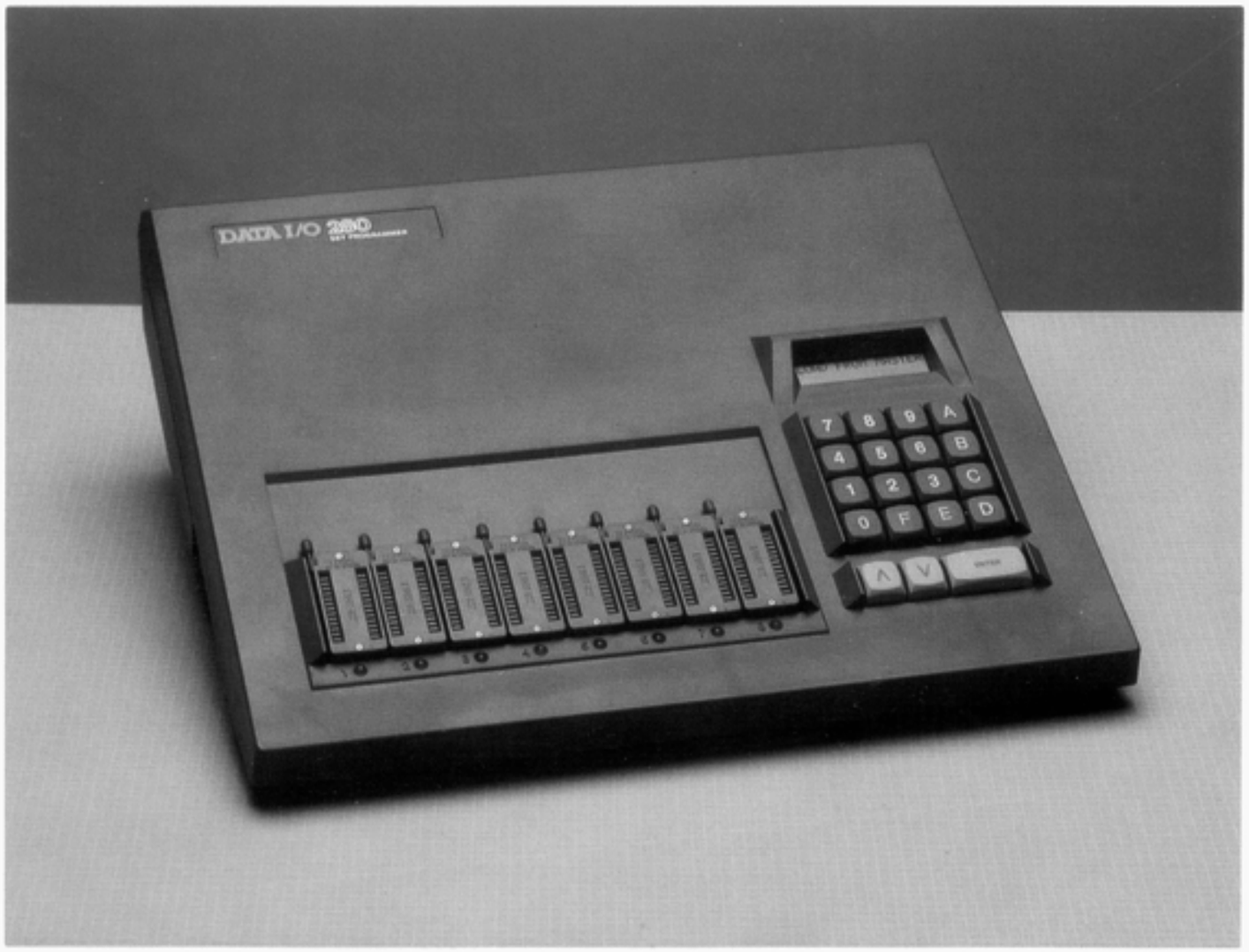
280 Set Programmer



Operator's Manual

DATA I/O

TERMINAL REMOTE CONTROL (continued)

The 280 Set Programmer is a completely self-contained gang programmer that allows you to program up to eight identical devices, or a set of up to eight devices containing different data. With the 280 Set Programmer you can program both 24 and 28 pin NMOS and CMOS EPROMs and EEPROMs without using special socket adapters or personality modules. You can operate the 280 Set Programmer "locally," using the front panel keys and 32-character display, or "remotely" using a terminal or computer and the RS232 serial I/O port.

The 280 offers several data translation formats, such as those supplied by Intel, Motorola, and Tektronix, which enable you to send files to and accept files from software development systems.

This manual contains the instructions necessary for operating the 280 Set Programmer both locally (from the front panel) and remotely.

Manual Contents

A brief description of this manual is provided below:

INTRODUCTION—This section provides a general description of the programmer, its specifications, warranty and service information and ordering information.

GETTING STARTED—This section provides instructions on how to set up the 280 Set Programmer for operation and a sample programming session to give you a general idea of how to operate the programmer.

FRONT PANEL OPERATION—This section provides detailed instructions on how to program devices using the front panel keys. How to program devices or sets is explained as well as how to download and upload data.

TERMINAL REMOTE CONTROL—This section provides instructions on how to operate the 280 Set Programmer from a remote terminal connected to the programmer through the serial interface (RS232) port.

COMPUTER REMOTE CONTROL—This section provides a description of the computer remote control command language. This command language can be used to write a software driver that allows you to operate the programmer using a host computer.

ERROR MESSAGES/DEVICE LIST—This section provides a list of the error messages displayed on the programmer. The meanings of the error messages are described as well as corrective action that should be taken. The device list provided to you should be placed in the section as well.

INDEX—This is an alphabetical guide to all major topics covered in this manual.

Specifications

Specifications for the 280 Set Programmer are listed below.

Functional Specifications

Functional specifications for the 280 are as follows:

Data RAM: 64K x 8 bytes

Translation Formats: Intel Intellec 8/MDS
Intel MCS-86 Hexadecimal Object
Motorola Exorciser
Motorola Exormax
Tektronix Hexadecimal

Keyboard: 16-key hexadecimal, 3-key functional

Display: 16 x 2 character alphanumeric display

Device Sockets: Eight 28-pin

Input/Output: Serial RS232 compatible

Remote Control: Computer Remote Control (CRC)
Terminal Remote Control (TRC)

Power Requirements

Power requirements for the 280 are as follows:

Operating Voltages: 110, 120, 220, or 240 V, +/-10%

Frequency Range: 48-63Hz, single-phase

Power Consumption: 165W nominal

Physical and Environmental

Physical and environmental requirements for the 280 are as follows:

Dimensions: 33cm x 11cm x 36cm (11.8" x 4.2" x 14.1")

Weight: 2.9 kg (6.4 lbs)

Operating Temperature Range: 5 to 45° C (41 to 113° F)

Storage Temperature Range: -40 to 70° C (-40 to 158° F)

Humidity: Up to 90%, noncondensing

Operational Altitude: To 10,000 ft

Getting Started

This section explains how to inspect the programmer, how to verify that the line voltage and fuse are correct, and how to ground the unit. Also given in this section is a brief description of the front panel keys and a sample EPROM programming session.

Connecting Power

Before connecting power to the programmer, perform the following checks:

- ☐ Make sure that the operating voltage is properly selected on the back panel of the unit.
- ☐ Make sure the correct line fuse is installed.
- ☐ Make sure the unit is properly grounded.

Verifying/Changing the Line Voltage and Fuse

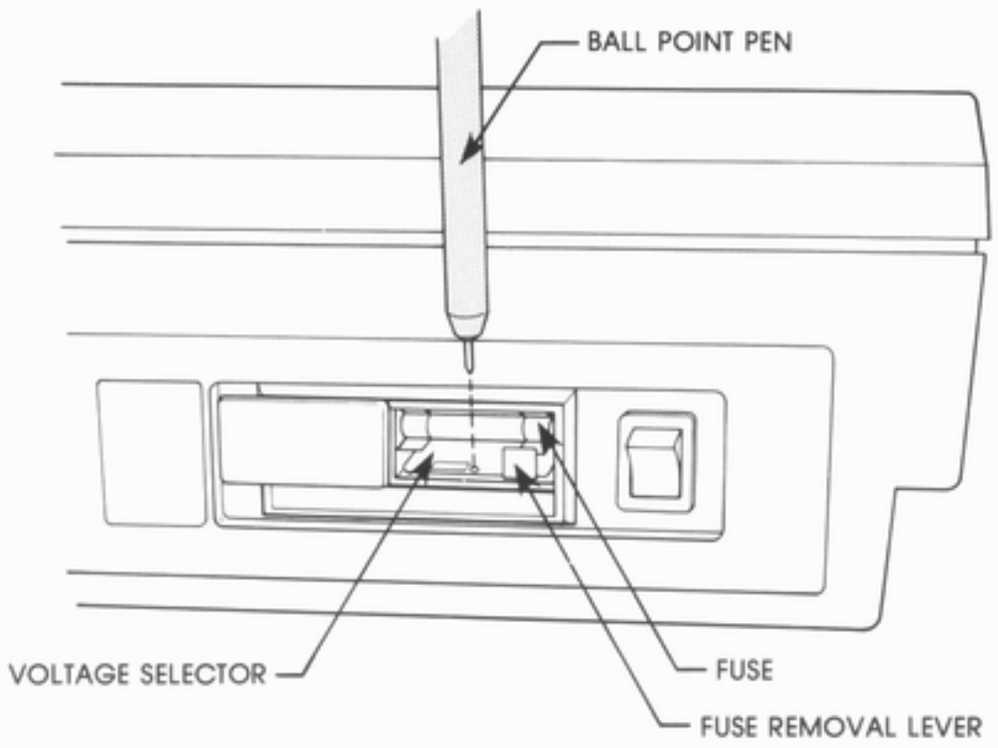
The factory has selected the proper line voltage according to your specification. A voltage label is stamped on the line voltage selector which is visible through the door that covers the voltage selector and fuse. The line voltage selector is a small printed circuit board that can be inserted into its slot in any of four different positions to accommodate 100, 120, 220, and 240 volt line voltages. Verify that the line voltage is correct. If it is not, perform the following procedures.

To change the line voltage and verify the line fuse, proceed as follows:

CAUTION

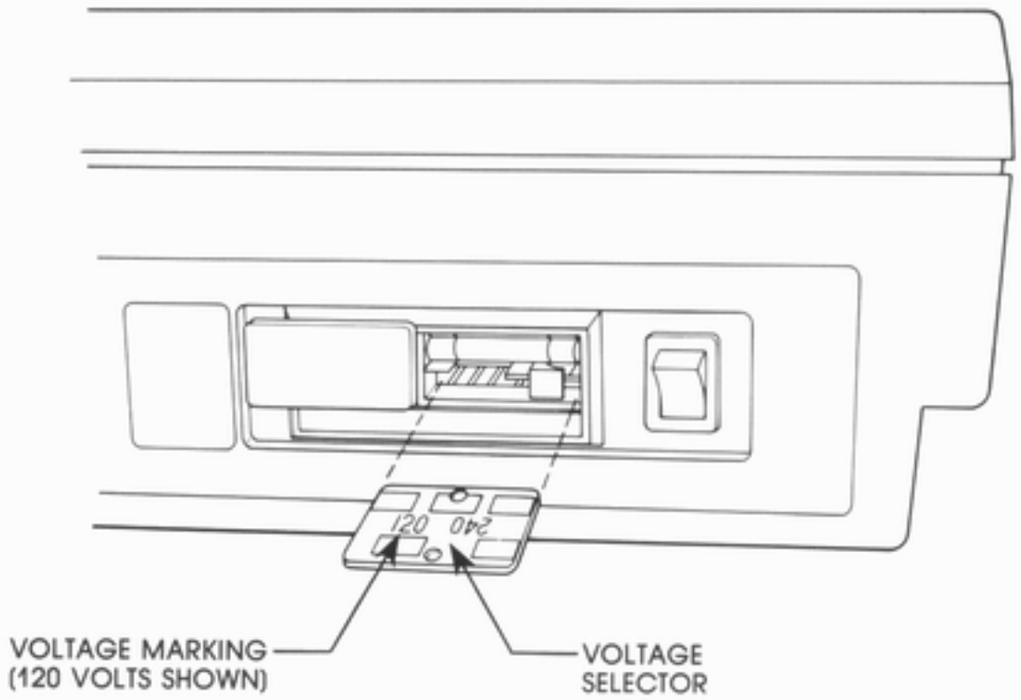
This instrument may be damaged if operated with the wrong line voltage.

1. Disconnect the power cord if plugged into the unit.
2. Slide open the clear plastic door that covers the fuse.
3. Remove the fuse by pulling on the lever labelled FUSE PULL.
4. Remove the voltage selector (small printed circuit board) by prying it out of its slot with a pointed object, such as a ballpoint pen.



5. Orient the voltage selector so that the desired voltage marking is on the top surface of the selector and to the left side as shown in the illustration.

6. Insert the voltage selector into its slot and push it firmly into place. If the selector is not inserted fully into its slot, no power connection will be made.



7. Verify that the fuse is the correct type and value per the following table; then install the correct fuse into the fuse holder.

	Line Fuse Rating			
Line Voltage	**Current**	**Voltage**	**Type***	**Data I/O Part Number**
100	1.5A	250V	Slo-blow	416-3012
120	1.5A	250V	Slo-blow	416-3012
220	0.8A	250V	Slo-blow	
240	0.8A	250V	Slo-blow	

* Littlefuse type 313, Bussman type MDA

8. Slide the door closed.

CAUTION

For continued protection against the possibility of fire, replace only with a fuse of specified voltage, current, and type ratings.

Grounding the Programmer

The Model 280 is shipped with a three-wire power cable. This cable connects the chassis of the programmer to earth ground when connected to a properly grounded three-wire ac receptacle.

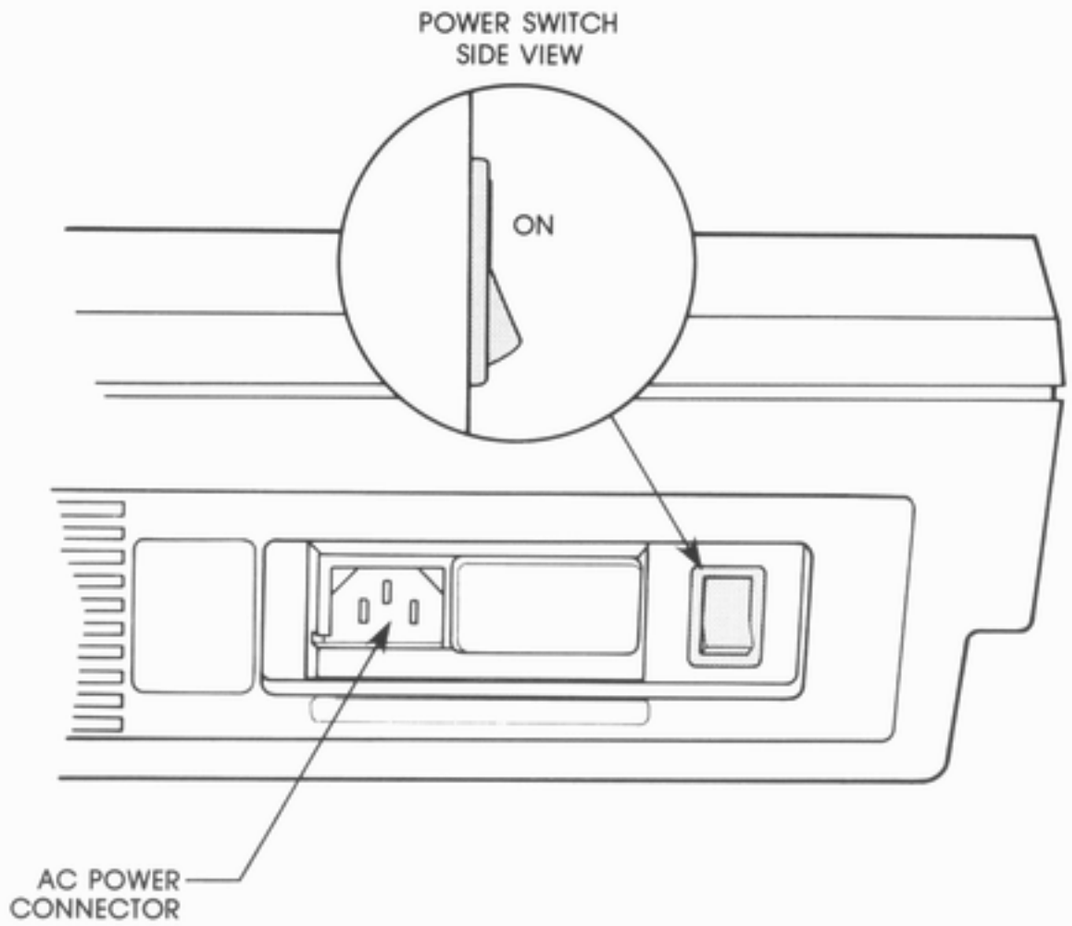
WARNING

Continuity of the grounding circuit is vital for the safe operation of the unit. Never operate this equipment with the grounding conductor disconnected.

Powering Up the Programmer

To power up the programmer, proceed as follows:

1. Make sure the sockets are empty.
2. Plug the power cable into the back panel connector and an ac power receptacle.
3. Press the power switch to the ON position (see the illustration).



When the programmer is turned on, it automatically performs a self test that verifies correct operation of the unit. During execution of the self test, the front panel display shows

```
SELF TESTING
```

with advancing decimal points on the lower portion of the display. On completion of the self test, the display shows

```
  SELF TEST OK
DATA I/O 280SP n
```

"n" is the software version number.

The programmer then displays the first main menu item.

```
LOAD FROM MASTER
```

If a device is present in any of the programmer's sockets when the power is switched on, the display reads

```
DEVICE IN SOCKET
 REMOVE DEVICE
```

To correct this condition, remove the device and press ENTER. The self test will then proceed.

The Main Menu Functions

The programmer can perform six basic operations, each of which is presented on the main menu. To display each of these operations, scroll through the main menu by means of the scroll keys shown in the illustration. (Scroll forward with the right-hand key and backward with the left-hand key.) The six main menu operations are:

- ☐ load from a master device to internal memory.
- ☐ load from a set of devices to internal memory.
- ☐ program one or more devices, or sets of devices, from the contents of internal memory.
- ☐ verify the contents of one or more devices against the contents of internal memory.
- ☐ perform a blank check on one or more devices.
- ☐ utilize the RS232 port.

Each of the six operations is fully described in the Front Panel Operation section; however, to familiarize you with the basic programming operation, a sample programming session is described in the following paragraphs.

Sample Programming Session

The following steps describe how to program a blank device from a master device. (A master device is a part that has been previously programmed and is used as a "master" to program blank parts). To perform this programming session, you'll need a master device and a blank device. The master device used in the following procedure is an Intel 2716, but could be any device shown on the Device List.

In the following procedure, the blank device is assumed to be of the same type as the master device, although it is possible for the master device to differ from the blank device(s). For more details on device programming, refer to the Front Panel Operation section of this manual.

1. Make sure all the sockets of the programmer are empty.

2. Turn on the power switch and wait until the self test is complete and the display reads

   ```
   LOAD FROM MASTER
   ```

3. Press ENTER to select the LOAD FROM MASTER operation. The display reads

   ```
   F/P CODE 79/33
   ```

4. Press the down-scroll key and the display reads

   ```
   LOAD FROM MASTER
         INTEL
   ```

   If the master device you are using in this sample programming session is not an Intel part, press the down-scroll key repeatedly until the display shows the name of the appropriate manufacturer.

NOTE

If you scroll past the desired selection, use the up-scroll key to move backwards through the menu.

5. When the correct manufacturer is displayed, press ENTER to select that manufacturer. The display reads

```
LOAD FROM MASTER
  INTEL  2764
```

NOTE

If you are using a device of a manufacturer other than Intel, the name of that manufacturer and a device number appears on the display.

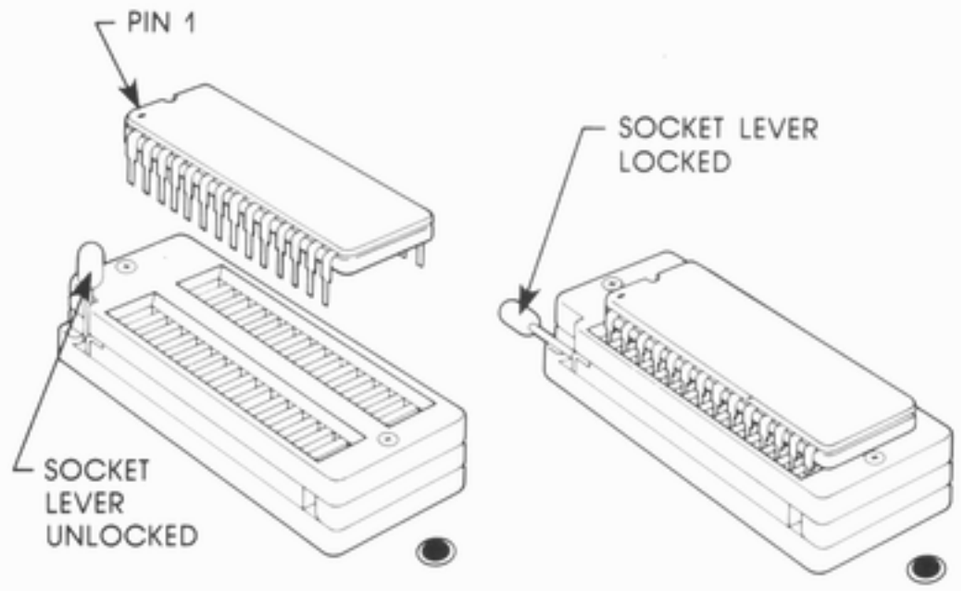
6. Press the up-scroll or down-scroll key repeatedly until the display shows the correct part number of your master device. If you are using an Intel 2716, the display should read

```
LOAD FROM MASTER
  INTEL  2716
```

7. Press ENTER to select the device type. The display reads

```
INSERT  DEVICE
 INTEL  2716
```

8. Insert the master device in the leftmost socket (socket 1) by lifting the lever and placing the device in the socket so that the bottom pins of the part are at the bottom of the socket and pin 1 is toward the top of the socket. Push the lever down to lock the device in place.



9. Press ENTER to begin the load operation. The display reads

```
LOAD FROM MASTER
.............
```

(advancing decimal points on the lower portion of the display indicate that the load operation is taking place.)

When the load operation is complete the display reads

```
LOAD FROM MASTER
 SUMCHECK = XXXX
```

where "XXXX" is the sumcheck of the device (the sum of all the data bytes in the device expressed as a hexadecimal number). Make a note of the sumcheck so that the part programmed later in this session can be verified.

10. Lift the socket lever and replace the master device with a blank device. Push the socket lever down.

11. Press the down-scroll key three times. The display reads

```
PROGRAM
```

12. Press the ENTER key to select the program function. The display reads

```
   PROGRAM
F/P CODE 19/23
```

(19/23 is the family/pinout code for the Intel 2716, and is now the default family/pinout code. If you used a master device other than an Intel 2716, the family/pinout code of the device you selected is now the default.)

Press the ENTER key to select the displayed family/pinout code. The display reads

```
INSERT  DEVICE
F/P CODE 19/23
```

13. Press the ENTER key to initiate the programming operation. During the programming operation the display reads

```
     PROGRAM
.................
```

(Advancing decimal points on the lower portion of the display indicate that the program operation is taking place.)

When the program operation is complete the display reads

```
    PROGRAM
SUMCHECK = XXXX
```

where "XXXX" is the sumcheck of the device. This hexadecimal number should match that displayed in step 9.

14. Lift up the socket lever and remove the programmed device.

This completes the sample programming session.

Front Panel Operation

Introduction

This section describes how to program and verify parts, and sets of parts, using the front panel keys and display of the 280 Set Programmer. Also provided are instructions for using the RS232 serial interface port in conjunction with front panel operations. Included in this section is the following information:

- ☐ CHECKING FOR NON-BLANK DEVICES — Describes how to verify that a device is completely blank (contains no data).
- ☐ PROGRAMMING SINGLE DEVICES OR GANGS — Describes load and program operations required to individually program and gang program devices.
- ☐ PROGRAMMING SETS OF DEVICES — Describes load and program operations required to program multi-device sets.
- ☐ VERIFYING PROGRAMMED PARTS — Describes how to verify the data programmed into a device against the contents of the programmer's memory.
- ☐ ESTABLISHING A COMMUNICATIONS LINK — Describes how to set up the programmer to transfer and receive program data through the RS232 serial port.
- ☐ DOWNLOADING DATA — Describes how to perform the transfer of data from a remote data source (computer or development system) to the programmer's memory.
- ☐ UPLOADING DATA — Describes how to perform the transfer of data from the programmer's memory to a host computer or other development system.

General Operating Notes

The following notes explain features, functions, and displays that are common to nearly all the front panel operations.

Family/Pinout Codes and Device Part Numbers

The required programming algorithm and the pin assignments of PROM and EPROM devices differs greatly. In order for the programmer to program a specific device, the device to be programmed must be selected on the programmer. Device selection may be made by selecting the manufacturer and part number from the 280 menus, or by entering a specific family/pinout code on the front panel keypad. The operating procedures presented in this section allow you to select the part by either method.

The family/pinout code for each programmable part is contained in the Device List accompanying this manual. The parts for each device manufacturer are listed in order of device size, with multiple devices of the same size being listed in numerical order.

Action Display

A special action display appears on the programmer during the execution of certain operations, such as self-test, program, and verify. This display consists of decimal points advancing across the lower portion of the front panel display and indicates that the programmer is executing the operation.

Aborting an Operation

Most operations may be aborted by pressing one of the hexadecimal keys. When an operation is halted in this manner, the programmer beeps and reverts to the currently selected mode; e.g., if a hexadecimal key is pressed during a "load from master" operation, the programmer beeps and the display returns to

```
LOAD FROM MASTER
```

If a hexadecimal key is accepted as input, press a scroll key to abort the operation.

Devices with Electronic Identifiers

To benefit from the electronic identifier feature built into many devices, an electronic identifier mode can be selected that causes the programmer to automatically select the correct programming algorithm and pin configuration. To select the electronic identifier mode when loading, programming, or verifying a device, enter FFFF when prompted for the family and pinout code.

When the electronic identifier mode is selected, the programmer determines the programming algorithm and pin configuration from the device installed in the leftmost socket. The programmer then checks to make sure that any devices installed in the other sockets can be programmed in an identical manner. If a conflict between devices exists, the programmer displays

```
ERROR 82 (invalid ID)
```

Error Indicators

If the selected device operation is successfully executed for each device in the sockets, a completion message appears such as

```
     PROGRAM
SUMCHECK = 1AE5
```

A pass signal (two beeps) sounds and the lamps below the sockets containing devices light green. However, if execution is unsuccessful for any device, a failure signal (several beeps) sounds, the lamp below the failed device(s) lights red, and a failure message appears as in the example below. (Refer to the Error Messages section at the back of the manual for a description of the error messages.)

```
     PROGRAM
BAD INSERTION 36
```

Checking for Non-blank Devices

A blank check is provided that can be used prior to programming operations to insure that no data has been previously written to the device(s). The blank check verifies that no data exists at any address of any devices installed in the programming sockets. Perform the blank check as follows:

Procedure	280 Displays
1. Scroll to the BLANK CHECK function.	BLANK CHECK
2. Press ENTER.	BLANK CHECK F/P CODE ff/pp
3. Key in the 4-digit family/pinout code for the device (e.g., 1925), and skip to step 7 or scroll to the device manufacturer (e.g., Motorola).	BLANK CHECK F/P CODE 19/25 BLANK CHECK MOTOROLA
4. Press ENTER when the correct manufacturer is displayed.	BLANK CHECK MOTOROLA 2716
5. Scroll to the part number of the device(s) (e.g., 2532).	BLANK CHECK MOTOROLA 2532

Procedure	280 Displays
6. Press ENTER to select the part number.	INSERT DEVICE MOTOROLA 2532
7. Insert the device(s), beginning at the leftmost socket and press ENTER to initiate the check.	BLANK CHECK . . . . .
If the blank check is successful for all devices installed, the socket lamps light green and the display reads	BLANK CHECK OK
If the blank check operation is unsuccessful for any devices installed, the socket lamp(s) for the failed device(s) light red and the display reads	BLANK CHECK NONBLANK 20
8. Remove the device(s) from the socket(s).	

Programming Single Devices or Gangs

The 280 Set Programmer can be used to program one or more devices with the data contained in a single master device, or with data downloaded to the programmer via the serial I/O (RS232) port. The following paragraphs describe the programming operation (for single devices and gangs) using a master device. If you choose to download the program data to the programmer's memory instead of loading from a master device, refer also to Downloading Data.

Programming a device consists of two basic operations:

- ☐ Loading the data from a master device to the programmer's memory (or downloading the data from a remote source such as a host computer — refer to Downloading Data).
- ☐ Programming the device (or devices) with the data copied to the programmer's memory.

Loading the Data from a Master Device

The first step in programming a device is to load the data from the master device. This operation transfers the programming data from the master device to the programmer's memory. When the transfer is complete, the programmer calculates and displays the sumcheck of the data. Use the following procedure to load the data from the master device.

	Procedure	280 Displays
1.	Scroll to the LOAD FROM MASTER function.	`LOAD FROM MASTER`
2.	Press ENTER.	`LOAD FROM MASTER` `F/P CODE ff/pp`

3.	If the correct family/pinout code is displayed, press ENTER and go to step 7. If the correct code is not displayed, key in the 4-digit family/ pinout code for the master device (e.g., 1925), and then skip to step 7
	
	or
	
	scroll to the device manufacturer (e.g., Motorola).
		`INSERT DEVICE`
		`F/P CODE 19/25`
		
		`LOAD FROM MASTER`
		`MOTOROLA`

4.	Press ENTER when the correct manufacturer is displayed.	`LOAD FROM MASTER` `MOTOROLA 2716`
5.	Scroll to the part number of the master device (e.g., 2532).	`LOAD FROM MASTER` `MOTOROLA 2532`

Procedure	280 Displays
6. Press ENTER to select the part number.	INSERT DEVICE MOTOROLA 2532
7. Insert the master device in the leftmost socket and press ENTER to initiate the load operation.	LOAD FROM MASTER . . . . .
8. When the load operation is complete, note the sumcheck displayed on the programmer for later verification.	LOAD FROM MASTER SUMCHECK = 9A1C
9. Lift the lever and remove the master device from the socket.	

Programming Devices (Single or Gang)

After the master data has been copied or downloaded to the programmer, use the following procedure to program one or more devices with the data. The program operation copies the data from the programmer's memory to the device(s) installed in the programming socket(s). When the programming operation is complete, the programmer calculates and displays the sumcheck of the data.

	Procedure	280 Displays
1.	Scroll to the PROGRAM function.	PROGRAM
2.	Press ENTER. If the data to be programmed was just copied from the master device and the master is the same type of device as the device(s) to be programmed, press ENTER and skip to step 7, otherwise proceed to select the device type for the parts to be programmed.	PROGRAM F/P CODE ff/pp
3.	Key in the 4-digit family/pinout code for the device (e.g., 1925), and skip to step 7 or scroll to the device manufacturer (e.g., Motorola).	INSERT DEVICE F/P CODE 19/25 PROGRAM MOTOROLA

Procedure	280 Displays
4. Press ENTER when the correct manufacturer is displayed.	PROGRAM MOTOROLA 2716
5. Scroll to the part number of the device (e.g., 2532).	PROGRAM MOTOROLA 2532
6. Press ENTER to select the part number.	INSERT DEVICE MOTOROLA 2532
7. Insert the blank devices in any socket and press ENTER to initiate the program operation.	PROGRAM . . . . .
8. When the program operation is complete, note that the sumcheck displayed on the programmer matches that displayed at the end of the load operation.	PROGRAM SUMCHECK = 9A1C
9. Remove the programmed device(s) from the socket(s).	

Programming Sets of Devices

The 280 can be used to program a set of devices (the devices must be of the same family/pinout code, refer to the Device List for the family and pinout codes) with the data contained in a set of master devices, or with data downloaded to the programmer via the serial I/O (RS232) port. The following paragraphs describe the programming operation using a set of master devices. If you choose to download the program data to the programmer's memory instead of loading from master devices, refer also to Downloading Data.

Programming a set of devices consists of two basic operations:

- ☐ Loading the data from a master set of devices to the programmer's memory (or downloading the data from a remote source such as a host computer -- refer to Downloading Data).
- ☐ Programming the device set with the data copied to the programmer's memory.

When loading data from more than one master into the programmer's memory, the data is copied into memory in the same order as the devices are installed in the sockets. The number of master devices installed in the sockets is automatically detected, and data from sockets 1 through n is copied sequentially into the programmer's memory (where n is the highest socket number with an installed device).

If any socket preceding the last device is left empty, then FF is read into memory for the address range assigned to that socket. The following table shows the maximum number of permissible masters for each device size, and the address range in memory where data is copied from each socket.

Socket Number	Device Size (no. of bits)					
	16K	32K	64K	128K	256K	512K
1	0000 07FF	0000 0FFF	0000 1FFF	0000 3FFF	0000 7FFF	0000 FFFF
2	0800 0FFF	1000 1FFF	2000 3FFF	4000 7FFF	8000 FFFF	
3	1000 17FF	2000 2FFF	4000 5FFF	8000 BFFF		
4	1800 1FFF	3000 3FFF	6000 7FFF	C000 FFFF		
5	2000 27FF	4000 4FFF	8000 9FFF			
6	2800 2FFF	5000 5FFF	A000 BFFF			
7	3000 37FF	6000 6FFF	C000 DFFF			
8	3800 3FFF	7000 7FFF	E000 FFFF			

Loading the Data from the Master Device Set

The first step in programming a device set is to load the data from the master device set. This operation transfers the programming data from the master device set to the programmer's memory according to the previous table. When the transfer is complete, the programmer calculates and displays the sumcheck of the data. Use the following procedure to load the data from the master device set.

Procedure	280 Displays
1. Scroll to the LOAD FROM SET function.	`LOAD FROM SET`
2. Press ENTER.	`LOAD FROM SET` `F/P CODE ff/pp`

3. If the correct family/pinout code is displayed, press ENTER and go to step 7. If the correct code is not displayed, key in the 4-digit family/pinout code for the master device (e.g., 1925), and then skip to step 7
 	
or
 	
scroll to the device manufacturer (e.g., Motorola).
	`INSERT DEVICE`
	`F/P CODE 19/25`
	
	`LOAD FROM SET`
	`MOTOROLA`

4. Press ENTER when the correct manufacturer is displayed.	`LOAD FROM SET` `MOTOROLA 2716`

Procedure	280 Displays
5. Scroll to the part number of the master devices (e.g., 2532).	LOAD FROM SET MOTOROLA 2532
6. Press ENTER to select the part number.	INSERT DEVICE MOTOROLA 2532
7. Insert the master device set, beginning at the leftmost socket and press ENTER to initiate the load operation.	LOAD FROM SET . . . . .
8. When the load operation is complete, note the sumcheck displayed on the programmer for later verification.	LOAD FROM SET SUMCHECK = 5D2F
9. Lift the levers and remove the master device set from the sockets.	

Programming a Set of Devices

The second step in programming a device set is to program the blank device set with the data copied to the programmer's memory. This operation copies the programming data from the programmer's memory to the device(s) installed in the programming socket(s), according to the previous table. When the programming operation is complete, the programmer calculates and displays the sumcheck of the data. Use the following procedure to copy the data from the programmer's memory to the set of devices installed in the programming sockets.

Procedure	280 Displays
1. Scroll to the PROGRAM function.	PROGRAM
2. Press ENTER. If the data to be programmed was just copied from the master device set and the master set contains the same type of devices as the set to be programmed, press ENTER and skip to step 7, otherwise proceed to select the device type for the parts to be programmed.	PROGRAM F/P CODE ff/pp
3. Key in the 4-digit family/pinout code for the devices (e.g., 1925), and skip to step 7 or scroll to the device manufacturer (e.g., Motorola).	PROGRAM F/P CODE 19/25 PROGRAM MOTOROLA

Procedure	280 Displays
4. Press ENTER when the correct manufacturer is displayed.	PROGRAM MOTOROLA 2716
5. Scroll to the part number of the devices (e.g. 2532).	PROGRAM MOTOROLA 2532
6. Press ENTER to select the part number.	INSERT DEVICE MOTOROLA 2532
7. Insert the master devices, beginning at the leftmost socket and press ENTER to initiate the program operation.	PROGRAM . . . . .
8. When the program operation is complete, make sure that the sumcheck displayed on the programmer matches that displayed at the end of the load operation.	PROGRAM SUMCHECK = 9A1C
9. Lift the levers and remove the programmed set of devices from the sockets.	

Verifying Programmed Parts

After the programmer has completed a programming operation, you can check the programmed device(s) to make sure that the data in the device(s) matches the data in the programmer's memory. The verify function compares device data with the data read into the programmer's memory during the preceding load or download operation.

The verify operation uses the VccH and VccL levels and the number of passes recommended by the manufacturer of the device being verified. VccH and VccL levels can be selected specifically when using Terminal Remote Control or Computer Remote Control.

If a single master device was specified in the preceding "load from master" (or download) operation, then verification compares the data in each programmed device with the same memory data. (The verify operation is repeated for each device installed in the programmer's sockets.)

If more than one master device was specified in the preceding "load from set" (or download) operation, then verification compares each device in the set with the data stored in the associated address range of the programmer's memory.

Performing the Verify Operation

Before you can verify the data within a device, a gang of devices, or a set of devices, you must first make sure that the master data is contained in the programmer's memory. You can load the master data to the programmer's memory by using one of the following methods:

- ☐ perform the previously described Load from Master operation.
- ☐ perform the previously described Load from Set operation.
- ☐ perform a download operation (described later in this section).

With the master data contained in the programmer's memory, perform the verify operation as follows:

	Procedure	280 Displays
1.	Scroll to the VERIFY function.	VERIFY
2.	Press ENTER. If the data to be verified was just copied from the master device or device set, and the master device(s) is the same type of device as the device(s) to be verified, press ENTER and skip to step 7, otherwise proceed to select the device type for the part(s) to be verifed.	VERIFY F/P CODE ff/pp
3.	Key in the 4-digit family/pinout code for the devices (e.g., 1925), and skip to step 7 or scroll to the device manufacturer (e.g., Motorola).	VERIFY F/P CODE 19/25 VERIFY MOTOROLA
4.	Press ENTER when the correct manufacturer is displayed.	VERIFY MOTOROLA 2716
5.	Scroll to the part number of the device (e.g., 2532).	VERIFY MOTOROLA 2532

Procedure	280 Displays
6. Press ENTER to select the part number.	INSERT DEVICE MOTOROLA 2532
7. Insert the device(s), beginning at the leftmost socket and press ENTER to initiate the verify operation.	VERIFY ..... then VERIFY SUMCHECK = 9A1C
If the verify operation is unsuccessful, the display reads	VERIFY VERIFY ERROR 29
10. Lift the levers and remove the verified device(s) from the socket(s).	

Establishing a Serial Communications Link

The programmer is equipped with a single serial RS232 compatible I/O port that can be used for the following programmer functions once a communications link is established:

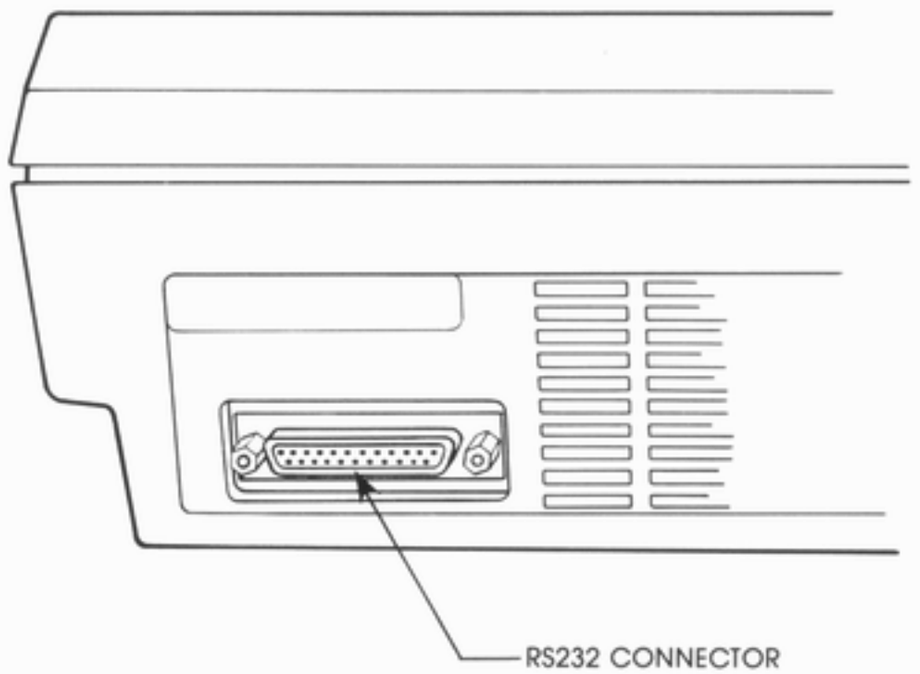
- ☐ downloading (receiving) programming data from a development system or host computer
- ☐ uploading (transferring) programming data to a host computer, development system, or remote terminal.
- ☐ interfacing with a remote terminal for control of the programmer.
- ☐ interfacing with a host computer for control of the programmer.

Cable Connections

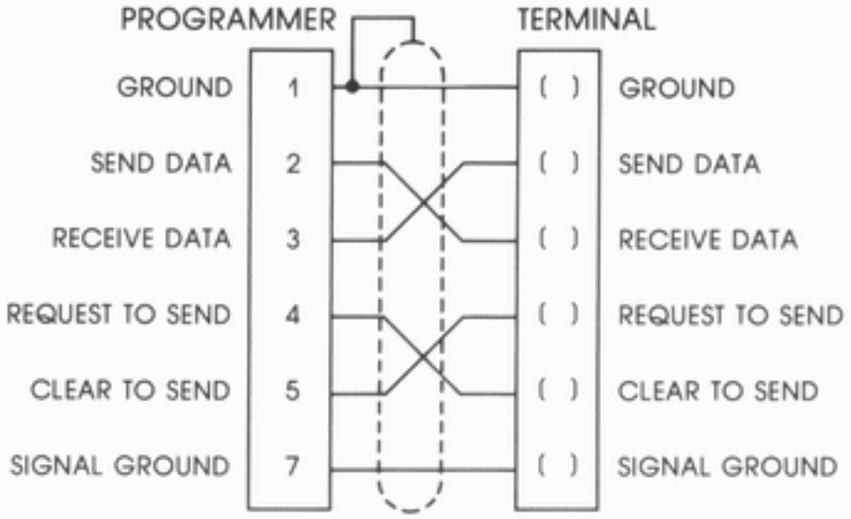
The communications link uses the serial I/O port (RS232 compatible) on the back panel of the programmer in a handshake or non-handshake mode. The cable connections required to the RS232 connector for each mode are shown in the accompanying illustrations.

NOTE

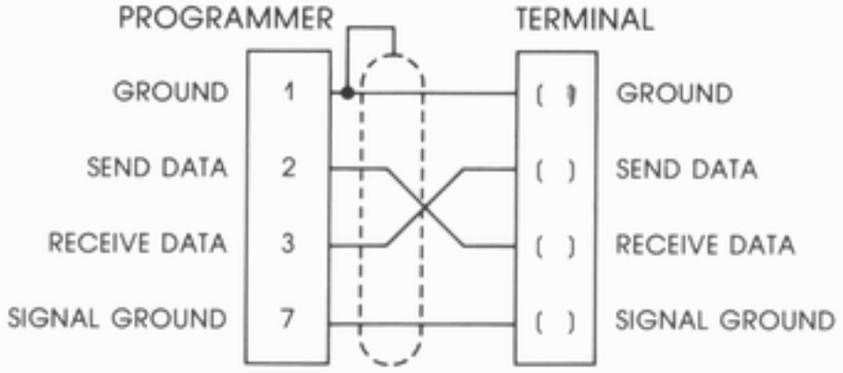
To reduce electromagnetic interference (EMI), we recommend using a shielded cable.



RS232 Connection



Half/Full Duplex, With Handshake



Half/Full Duplex, Without Handshake

NOTES

1. *All signals are named with respect to the originating unit.*
2. *All undesignated pins are to be left open.*
3. *For applications that do not require handshaking, the programmer's clear to send line is pulled up internally.*
4. *Host system's pin numbers may differ.*

Setting Communications Protocol

The programmer can be made to select the correct communications protocol automatically (see Terminal Remote Control or Computer Remote Control) or you can change the communications protocol from the factory default values by using the following procedure. (The factory default values are baud rate = 9600, parity check = none, number of data bits = 8, and the number of stop bits = 1.)

	Procedure	280 Displays
1.	Scroll to the RS232 PORT function.	RS232 PORT
2.	Press ENTER.	RS232 PORT COMPUTER CONTROL
3.	Scroll to the PORT SETTING function.	RS232 PORT PORT SETTING
4.	Press ENTER.	PORT SETTING BAUD RATE = 9600
5.	Scroll to the desired data transfer speed and press ENTER.	PORT SETTING PARITY CHK.=NONE
6.	Scroll to the desired parity and press ENTER.	PORT SETTING DATA BITS = 8

Procedure	280 Displays
7. Scroll to the desired number of data bits and press ENTER.	PORT SETTING STOP BIT = 1
8. Scroll to the desired number of stop bits and press ENTER.	RS232 PORT

This completes setting of the communications protocol for the serial I/O port.

Downloading Data

The 280 can receive data transferred from a host computer to its memory over a serial communications link. (Refer to Establishing a Communications Link for information on serial I/O connections and setup.) This feature may be used to download data from a computer, or other software development system, to the programmer in preparation for programming parts. The programmer can accept the data in any of five data translation formats: Intel Intellec 8/MDS, Intel MCS-86 Hexadecimal Object, Motorola Exorciser, Motorola Exormax, and Tektronix Hexadecimal.

When initiating a download operation, the programmer prompts for the entry of three parameters in addition to the data translation format. These paramters are:

- ☐ the beginning address
- ☐ the block size
- ☐ the offset address

Selection of the beginning address allows you to make transfers of data to a specified start address in the programmer's memory. For example, if you specify a beginning address of 1000H (hexadecimal), the data transferred from the external data source will be placed in the programmer'memory starting at location 1000H instead of 0000H.

Selection of the block size allows you to specify how much data is to be transferred to the programmer's memory (starting at the specified beginning address). For example, if you specify a beginning address of 1000H and a block size of 2000H, the subsequent data transfer operation would transfer 8192 (2000H) bytes of data to the programmer's memory, with the first byte written to location 1000H.

The offset address allows you to select the beginning address, in the host computer memory, of the file to be downloaded. The download operation adds the specified offset address to the specified start address to determine the address in the computer memory where downloading will begin (i.e., the file address). For example, if you specify a beginning address of 1000H and an offset address of 2000H, then the first byte of data downloaded will be from address 3000H in the computer memory.

Perform a download operation as follows:

Procedure	280 Displays
1. Scroll to the RS232 PORT function.	RS232 PORT
2. Press ENTER.	RS232 COMPUTER CONTROL
3. Scroll to the DOWNLOAD function.	RS232 PORT DOWNLOAD
4. Press ENTER.	DOWNLOAD FORMAT INTEL INTELLEC 8
5. Scroll to the desired data translation format (e.g., Motorola Exormax) and press ENTER.	DOWNLOAD EXORMAX BEG RAM ADD=0000
6. Press ENTER to select the displayed beginning RAM address (the first programmer memory address that the downloaded data will be transferred to) or key in the hexadecimal beginning address and press ENTER.	DOWNLOAD EXORMAX BLOCK SIZE =0000
7. Press ENTER to select the displayed block size, or key in the block size of the data to be downloaded and press ENTER.	DOWNLOAD EXORMAX OFFSET =FFFFFFFF

Procedure	280 Displays
8. Press ENTER to select the displayed offset address or key in the offset address required to specify the file address in the host computer (e.g., 2000). (The default offset address, FFFFFFFF, is the first address received by the programmer.)	DOWNLOAD EXORMAX OFFSET =00002000
9. Press the ENTER key to ready the programmer to receive the downloaded data. You can now initiate the download operation from the host computer.	LOADING PORT

Uploading Data

The 280 can be used to transfer device data from its memory to a computer over a serial communications link. (Refer to Establishing a Communications Link for information on serial I/O connections and setup.) This feature may be used to upload data from the programmer's memory for the purpose of editing the data and then downloading it back to the programmer memory. The programmer can transfer the data in any of five data translation formats: Intel Intellec 8/MDS, Intel MCS-86 Hexadecimal Object, Motorola Exorciser, Motorola Exormax, and Tektronix Hexadecimal.

When initiating an upload operation, the programmer prompts for the entry of three parameters in addition to the data translation format. These paramters are:

- ☐ the beginning address
- ☐ the block size
- ☐ the offset address

Selection of the beginning address allows you to make transfers that use only a portion of data contained in the programmer's memory. For example, if you specify a beginning address of 8000H (hexadecimal) for data copied from a 64K x 8 device, the data transfer operation will begin at address 8000H (32,768 decimal).

Selection of the block size allows you to specify how much of the data contained in the programmer's memory is to be transferred, starting at the specified beginning address. For example, if you specify a beginning address of 8000H and a block size of 1000H for data copied from a 64K x 8 device, the subsequent data transfer operation would transfer only the data contained in the address range 8000H to 9000H.

Selection of the offset address allows you to select the address in the computer memory where the data will be uploaded. Unlike the download operation, the specified offset address is subtracted from the start address to determine the address in the computer memory where uploading will begin. For example, if you specify a beginning address of 8000H and an offset address of 2000H, then the first address of device data will be uploaded to address 6000H in the host computer.

Perform an upload operation as follows:

Procedure	280 Displays
1. Scroll to the RS232 PORT function.	RS232 PORT
2. Press ENTER.	RS232 PORT COMPUTER CONTROL
3. Scroll to the UPLOAD function.	RS232 PORT UPLOAD
4. Press ENTER.	UPLOAD FORMAT INTEL INTELLEC 8
5. Scroll to the desired data translation format (e.g., Tektronix Hexadecimal) and press ENTER.	UPLOAD TEKTRONIX BEG RAM ADD=0000

Procedure	280 Displays
6. Press ENTER to select the displayed beginning RAM address, or key in the hexadecimal beginning address of the data in the programmer's memory to be uploaded and press ENTER.	UPLOAD TEKTRONIX BLOCK SIZE =0000
7. Press ENTER to select the displayed block size, or key in the block size of the data to be uploaded and press ENTER.	UPLOAD TEKTRONIX OFFSET =FFFFFFFF
8. Press ENTER to select the displayed offset address or, key in the offset address for the data. (The default offset address, FFFFFFFF, is 0.)	UPLOAD TEKTRONIX OFFSET =FFFFFFFF
9. Press the ENTER key to initiate the upload operation.	SENDING PORT ..... then SENDING COMPLETE SUMCHECK = xxxx

Terminal Remote Control

Introduction

This section explains how to operate the 280 Set Programmer from a remote terminal. All of the operations that can be performed from the front panel are available in Terminal Remote control mode, as well as editing commands, and commands for programming with data in 16-bit-wide word format and 32-bit-wide word format.

A summary of all of the commands used to operate the programmer from a remote terminal is presented immediately following this introduction and some general operating notes follow the command summary. Read the general operating notes before attempting any of the Terminal Remote Control commands.

The commands for operation from a remote terminal are grouped into the following subsections:

- ☐ ENTERING AND EXITING TERMINAL REMOTE CONTROL MODE–This subsection explains how to transfer control of the programmer to a remote terminal and how to return control to the front panel of the programmer.
- ☐ ON-LINE HELP–This subsection explains how to display the on-line help screen and how to read the help screen symbols.
- ☐ SELECTING A DEVICE TYPE–This subsection explains how to select the device type. Selecting the device type tells the programmer which algorithm and voltages it should use to blank check, load, program, and verify a part.
- ☐ PROGRAMMING OPERATIONS–This subsection explains how to perform a blank check on devices prior to programming them, load the master data into RAM from previously programmed devices, and program blank devices with the master data.

- ☐ VERIFYING PROGRAMMED DEVICES–This subsection explains how to verify devices after they have been programmed. Devices are automatically verified during the programming operation, but this command allows you to verify the devices at another time.

- ☐ EDITING MEMORY–This subsection explains how to edit the contents of the programmer's data memory, or RAM.

- ☐ SUMCHECKING DATA–This subsection explains how to perform a sumcheck and a checksum on the data stored in the programmer's RAM.

Symbols and Conventions

The following is a list of symbols and conventions used in the Terminal Remote Control section of this manual:

<CR> This symbol represents the carriage return key.

<ESC> This symbol represents the escape key.

CTRL-S CTRL- followed by a character means that you should hold down the CONTROL (or CTRL) key and press the character once. For example, to type CTRL-S, you would hold down the CONTROL key and press the S key once.

H Any value followed by an upper-case H means that the value is in hexadecimal form.

<lower-case> Lower-case letters enclosed in angle brackets describe the kind of parameter that you should fill in for the command. Replace the words and the brackets with the appropriate parameter value for the operation you are performing. For example, the Memory Modify command format is M <start address>. You would enter M 6 to modify memory starting at address 6.

[<lower-case>] Words or characters enclosed in square braces represent optional entries.

Terminal Remote Control Command Summary

Command Format	Description	Command Format	Description
M	Enter TRC mode	P [a1 a2 a3]	Program device(s)
R	Exit TRC (Return)	PS	Program Set
CARRIAGE RETURN	Execute command	PW	Program Word
BACKSPACE	Backspace	PL	Program Long word
ESCAPE	Abort command		
CTRL-S	Suspend display	CT a1 a2 [a3]	Sumcheck (Total)
CTRL-Q	Continue display	CX a1 a2 [a3]	Checksum (EXOR)
?	Display help screen	V [VccADJ]	Verify single device
		VS [VccADJ]	Verify Set
A?	Select device type	VW [VccADJ]	Verify Word
		VL [VccADJ]	Verify Long word
B	Blank check devices		
		M a1 a2	Display Memory
L	Load single master	M a1	Modify Memory
LS [master #]	Load Set of masters	M a1 a2 d1 [d2...d8]	Fill Memory
LW [master #]	Load Word	I a1 d1 [d2...d8]	Insert data
LL [master #]	Load Long word	D a1 [12]	Delete data
		T a1 a2 a3	Transfer (copy)
		MS a1 a2 d1 [d2...d8]	Search Memory

a1 = start address
a2 = end address
a3 = destination address

d1...d8 = data bytes 1 through 8
master # = number of masters
VccADJ = Vcc selection (1 or 2)

General Operating Notes

This subsection contains important information on how to type in the Terminal Remote Control (TRC) commands, as well as information on how to use some special keys to control command execution, what the default parameter values are, and how to read the device error indicators.

Entering Commands

Below are some general rules for entering Terminal Remote Control commands.

- ☐ Each command must be entered at the > prompt and must be followed by a carriage return.
- ☐ All commands must be typed in upper-case letters.
- ☐ The first parameter (such as a memory address) that follows the command letter(s) can either be separated from the command letter(s) by a space or it can be typed immediately following the command letter(s) with no space. For example, the Load Set command could be typed LS4 or LS 4 to load a set of four masters.
- ☐ Each parameter following the first parameter must be separated from the preceding parameter by a space. For example, the Display Memory command must be typed M 0 F or M0 F to display memory addresses 0 through F.
- ☐ You do not need to enter leading zeros when entering command parameters, such as memory addresses. For example, 0010 is the same as 10.
- ☐ All command parameters are entered as hexadecimal numbers, unless otherwise specified.

Controlling Command Execution

The following keys are special keys which can be used to control the execution of TRC commands.

BACKSPACE Key (CTRL-H)

While you are entering TRC commands, you can use the BACKSPACE key (or the CTRL-H, if your terminal keyboard has no BACKSPACE key) to delete the previously entered character. You can use this key repeatedly until you have deleted the entire command line.

ESCAPE Key

You can abort the command currently being executed or the command you are currently typing by pressing the ESCAPE key. If you press the ESCAPE key while typing in a command (before pressing carriage return), the programmer will ignore the line you were typing, display the > prompt on a new line, and await your input. If you press the ESCAPE key while a command is being executed, the programmer will abort the command, display the > prompt, and await your input.

CTRL-S

Typing a CTRL-S will suspend the displaying of data on the terminal screen. The command that was being executed when the CTRL-S was typed is temporarily halted, but not aborted. To continue the display, type CTRL-Q.

CTRL-Q

Typing a CTRL-Q continues the displaying of data on the terminal screen exactly where it was suspended by the CTRL-S command.

Action Display

While an operation is being executed in TRC mode, advancing dots will appear on the bottom line of the programmer display, indicating that the programmer is operating.

Error Indicators

The lights (LEDs) below each of the sockets are used to indicate successful or unsuccessful programming operations. Many Terminal Remote Control commands also cause the programmer to display a table showing the results of the operation performed on each socket (see the example below).

```
DEVICE: 1 2 3 4 5 6 7 8
        P P P F X X X X
```

If an operation is completed successfully on a device, a "P" (which stands for pass) will appear below the socket number (see example above) containing that device and the socket lamp will light green.

If an operation is not completed successfully on a device, an "F" (which stands for fail) will appear below the socket number (see example above) containing that device and the socket lamp will light red.

If a socket is left empty during an operation, an "X" will appear below the socket number (see example above) of the empty socket and the socket lamp will remain off.

A "SYNTAX ERROR" will occur when the format of a command is incorrect or when illegal characters are entered in a command. A caret (^) will appear below the first unrecognized character encountered by the programmer.

Default Parameter Values

Many of the Terminal Remote Control commands can be accompanied by optional parameters, such as starting and ending memory addresses. The default values of these optional parameters are listed below.

<start address>	0000H
<end address>	1FFFH
<destination>	0000H
<number of masters>	
for a set	1
for a word-wide set	2
for a long-word-wide set	4
Verify <Vcc selection>	0 (manufacturer's recommendation)

Entering and Exiting Terminal Remote Control (TRC) Mode

The following is an explanation of how to transfer control of the programmer to the remote terminal (enter TRC mode) and how to return control to the front panel of the programmer (exit TRC mode).

Entering Terminal Remote Control Mode

Procedure	280 Displays
1. Connect the programmer to the terminal as described in Establishing a Serial Communications Link in the Front Panel Operation section of this manual.	
2. Power up the programmer by plugging the power cord into the back of the programmer and into a power outlet and flipping the power switch on the back of the programmer up to the ON position.	SELF TESTING . . . . .
3. When the self-test is complete and the display reads LOAD FROM MASTER, scroll through the main menu to RS232 PORT and press ENTER.	RS232 PORT COMPUTER CONTROL
4. Scroll through the RS232 PORT menu to TERMINAL CONTROL and press ENTER.	TERMINAL CONTROL

Procedure	280 Displays
5. To establish communications between the programmer and the terminal and automatically select the proper port (I/O) settings, type M<*CR*> on the terminal keyboard. You can set the I/O (or port) settings manually, by executing the PORT SETTING function (see the Front Panel Operation section). If the port settings are correct when you enter TRC mode, simply press any terminal key to establish communication. When communication is established, a command prompt (>) will appear on the terminal screen. The programmer will display TERMINAL CONTROL until control of the programmer is returned to the front panel.	TERMINAL CONTROL

Exiting Terminal Remote Control Mode

You can exit Terminal Remote Control mode by using either the terminal keys or the programmer keys.

To exit TRC using the terminal keys, type

R<*CR*>

(The R command stands for Return to local.)

or

To exit TRC using the programmer keys, press any key.

The programmer will display

```
RS232 PORT
```

You can now continue operating the programmer using the front panel keys.

On-Line Help

A help screen displaying the currently selected device type and a list of all of the TRC commands and their formats can be displayed on the terminal screen by typing

?<CR>

at the command prompt (>). The help screen is shown below:

```
                ***   280 SET PROGRAMMER COMMAND HELP   ***
-------------------------------------------------------------------------------

SELECT DEVICE TYPE  A?              \   SUMCHECK (TOTAL)  CT A1 A2[ A3]
BLANK CHECK         B               \   CHECKSUM (EXOR)   CX A1 A2[ A3]
LOAD SINGLE         L               \   MEMORY DISPLAY    M A1 A2
LOAD SET            LS [MASTER #]   \   MEMORY MODIFY     M A1
LOAD WORD           LW [MASTER #]   \   MEMORY FILL       M A1 A2 D1[..D8]
LOAD LONG WORD      LL [MASTER #]   \   INSERT            I A1 D1[..D8]
PROGRAM SINGLE      P [A1 A2 A3]    \   DELETE            D A1[ A2]
PROGRAM SET         PS              \   TRANSFER          T A1 A2 A3
PROGRAM WORD        PW              \   MEMORY SEARCH     MS A1 A2 D1[..D8]
PROGRAM LONG WORD   PL              \   RETURN TO LOCAL   R
VERIFY SINGLE       V [VCCADJ]      \   HELP              ?
VERIFY SET          VS [VCCADJ]     \
VERIFY WORD         VW [VCCADJ]     \
VERIFY LONG WORD    VL [VCCADJ]     \

DEVICE TYPE:  INTEL  2764       F/P: 79/33     MASTER #: 01
>
```

A1 = start address
A2 = end address
A3 = destination address

D1..D8 = data bytes 1 through 8
MASTER # = number of masters
VCCADJ = Vcc selection (1 or 2)

Selecting a Device Type

Use the Select Device Type command to select the manufacturer and part number of the device you are loading from, programming, blank checking or verifying. After you have selected the device's manufacturer and part number, the 280 Set Programmer will automatically supply the proper family and pinout codes for the device. The family and pinout codes tell the programmer which algorithm and voltages to use to program the part. You must select the correct device type before you can blank check a device, load from a device, program a device, or verify a device. The default device type is Intel 2764.

Command Format: A?

Procedure	Example Key Sequence
1. Type in the Select Device Type command letters, A?, and press carriage return.	A?<*CR*>
2. After the list of device manufacturers is displayed on the screen, type in the number that corresponds to the manufacturer of the device you are loading, programming or verifying and press carriage return. If you press carriage return without entering a number, device manufacturer 0 will be selected.	7<*CR*>
3. After the list of device part numbers is displayed on the screen, type in the number that corresponds to the correct device part number and press carriage return. If you press carriage return without entering a number, part number 0 will be selected. The device type selected will remain the default device type until a new device type is selected. The power-up default device is Intel 2764.	1<*CR*>

Example

Select Intel device 2816:

```
>A?
DEVICE MANUFACTURERS

0.AMD                 7.HITACHI          14.NCR             21.SIGNETICS
1.CYPRESS             8.INTEL            15.NEC             22.TEXAS INSTRU.
2.ELECTRONIC ARRAY    9.LATTICE          16.OKI             23.TOSHIBA
3.EUROTECHNIQUE      10.MITSUBISHI       17.RICOH           24.VTI
4.EXEL               11.MOSTEK           18.ROCKWELL        25.XICOR
5.FUJITSU            12.MOTOROLA         19.SEEQ
6.GENERAL INSTRU.    13.NATIONAL         20.SGS-ATES

ENTER MANUFACTURER NUMBER AND RETURN 8<CR>

INTEL DEVICES

0. 2716               7. 2764           13. 27128A
1. 2816               8. 2764A          14. P27128A
2. 2816A              9. 27C64A         15. 27256
3. 2732              10. P2764          16. 27C256
4. 2732A             11. P2764A         17. 27512
5. 2732B             12. 27128          18. 27513
6. P2732A

ENTER PART NUMBER AND RETURN 1<CR>

FAMILY AND PINOUT CODE IS 37/23
>
```

NOTE

The above list is only an example. The number of devices available may be different at the time you purchase your programmer.

Programming Operations

The following commands allow you to perform programming operations from the remote terminal. With these commands you can check to make sure devices are blank, load data into RAM from master devices, and program blank devices with the master data. The commands are presented in the following order:

Command Name	Command Format
Blank Check	B
Load Single	L
Load Set	LS [<number of masters>]
Load Word	LW [<number of masters>]
Load Long Word	LL [<number of masters>]
Program Single	P [<start address> <end address> <destination address>]
Program Set	PS
Program Word	PW
Program Long Word	PL

Checking for Non-blank Devices

The Blank Check command allows you to check a device prior to programming it to make sure that it is blank.

Command Format: B

	Procedure	Example Key Sequence
1.	Insert the device or devices to be checked in any socket or sockets and lock them into place by pushing the socket levers down.	
2.	Type in the Blank Check command letter, B, and press carriage return.	B<*CR*>
3.	Check the screen display for the letters that appear below the sockets containing devices to be checked. A "P" (for "pass") appears below any socket containing a blank device, and an "F" (for "fail") appears below any socket containing a non-blank device.	
4.	Lift the socket levers and remove the devices.	

Example

Check eight devices for non-blank bits:

```
>B<CR>
DEVICE: 1 2 3 4 5 6 7 8
        P P F P P P P P
>
```

NOTE : Device three failed the blank check.

Loading Device Data Into Memory

The following commands allow you to load data from a single master (a previously programmed device that is used as a "master" to program other devices) or sets of master devices into the programmer's data memory, or RAM. The sets of devices can be loaded into RAM in 8-bit-wide word format, 16-bit- wide word format, or 32-bit-wide word format. Once the data has been loaded into RAM, the data can then be edited, programmed into blank devices, or uploaded to a computer for further editing and development on the computer system. Data can also be downloaded into programmer RAM from a remote computer, instead of being loaded from a master (see the Front Panel Operation section or the Computer Remote Control section for instructions on downloading).

Loading a Single Master

Use the following command to load (or copy) data from a single master device into the programmer's data RAM. Loading the master data into RAM is the first step in programming devices. The second step is to copy the data from RAM to a blank device using a program command. The format of the Load Single command is shown below, followed by the procedures for loading a single device.

Command Format: L

Procedure	Example Key Sequence
1. Insert the master device into the leftmost socket, socket 1, and push the socket lever down to lock the device into place.	
2. Type the Load Single command letter, L, and press carriage return.	L<CR>
3. Check the screen display for the letter that appears below device socket 1 (see example). If the device was loaded successfully, a "P" will appear below socket 1 and the socket lamp for socket 1 will light green. If the device was not loaded successfully, an "F" will appear below socket 1 and the socket lamp for socket 1 will light red. Also note the sumcheck total which is displayed below the device information. The sumcheck is the hexadecimal total of all data loaded. The sumcheck should match the sumcheck of any device programmed with this data.	
4. Lift the socket lever and remove the device.	

Example

Load data into memory from a single master device:

```
>L<CR>
DEVICE: 1 2 3 4 5 6 7 8
        P X X X X X X X

THE SUMCHECK (TOTAL) IS:  E000

>
```

Loading a Set of Masters

Use the following command to load data from a set of master devices into programmer RAM for subsequent editing, programming into blank devices, or uploading. Each device's data is placed into a continuous block of programmer RAM and the RAM blocks are arranged in the same order as the devices in the sockets. You can use this command if you want to produce sets of devices identical to the masters or you can use this command if you want to merge the data loaded from multiple devices into larger devices.

Command Format: LS [<number of masters>]

Procedure	Example Key Sequence
1. Insert the set of masters into consecutive sockets, beginning at socket 1 (the leftmost socket), and push down the socket levers. (See the table following the example for the maximum number of devices that can be loaded at once and the programmer RAM locations assigned to each device.) The data will be loaded into RAM in the same order as the devices are placed in the sockets.	
2. Type the Load Set command letters, LS.	LS
3. Type in the number of masters in the set to be loaded and press carriage return, or to load the default number of masters, press carriage return without entering a number of masters. The power-up default number of masters is one. (The space after "LS" is optional.) If a socket to the left of the last master is left empty, FF will be loaded into the RAM block assigned to that socket.	LS 4<*CR*> or LS<*CR*>

Procedure	Example Key Sequence
4. Check the screen display for the letters that appear below the sockets containing devices. A "P" will appear below any socket containing a device that was loaded successfully and an "F" will appear below any socket containing a device that was not loaded successfully. Also note the sumcheck total which appears below the device information. The sumcheck total should match the sumcheck of any set programmed with this data.	
5. Lift the socket levers and remove the devices.	

Example

Load a set of four devices:

```
>LS 4<CR>
DEVICE: 1 2 3 4 5 6 7 8
        P P P P X X X X

THE SUMCHECK (TOTAL) IS:  15E7

>
```

Programmer RAM blocks assigned to set devices being loaded:

Socket Number	Device Size (no. of bits)					
	16K	32K	64K	128K	256K	512K
1	0000 07FF	0000 0FFF	0000 1FFF	0000 3FFF	0000 7FFF	0000 FFFF
2	0800 0FFF	1000 1FFF	2000 3FFF	4000 7FFF	8000 FFFF	
3	1000 17FF	2000 2FFF	4000 5FFF	8000 BFFF		
4	1800 1FFF	3000 3FFF	6000 7FFF	C000 FFFF		
5	2000 27FF	4000 4FFF	8000 9FFF			
6	2800 2FFF	5000 5FFF	A000 BFFF			
7	3000 37FF	6000 6FFF	C000 DFFF			
8	3800 3FFF	7000 7FFF	E000 FFFF			

Loading Word-Wide Masters

Use the Load Word command to load pairs of devices that are set up to be read as 16-bit-wide words. This command loads the data from a pair of 8-bit-wide devices into a 16-bit-wide word format, reading the first address of the first device and then the first address of the second device into consecutive RAM locations. The addresses are loaded in pairs, as described above, consecutively from the first pair to the last pair (see illustration). You can load more than one pair of 8-bit-wide devices, as long as the total size of all the devices does not exceed 512K bits. See the following table for the maximum number of pairs that can be loaded at once.

Command Format: LW [<number of masters>]

Procedure	Example Key Sequence
1. Place the pair or pairs of 8-bit-wide devices into the sockets, starting with socket 1, and lock them into place. Place the devices of each pair in adjacent sockets, with the device containing the high-order data to the right of the device with the low-order data for the pair.	
2. Type in the Load Word command letters, LW.	LW
3. Type in the number of devices being loaded and press carriage return, or to load the default number of masters, press carriage return without entering the number of masters. The power-up default number of masters is 2. (The space after "LW" is optional.)	LW 4<CR> or LW<CR>

Procedure	Example Key Sequence
4. Check the screen display for the letters that appear below the sockets containing devices. A "P" will appear below any socket containing a device that was loaded successfully and an "F" will appear below any socket containing a device that was not loaded successfully. Also note the sumcheck total which appears below the device information. The sumcheck should match the sumcheck of any set of devices programmed with this data.	
5. Lift the socket levers and remove the devices.	

Device Size	Maximum Number of Pairs
16K	4
32K	4
64K	4
128K	2
256K	1

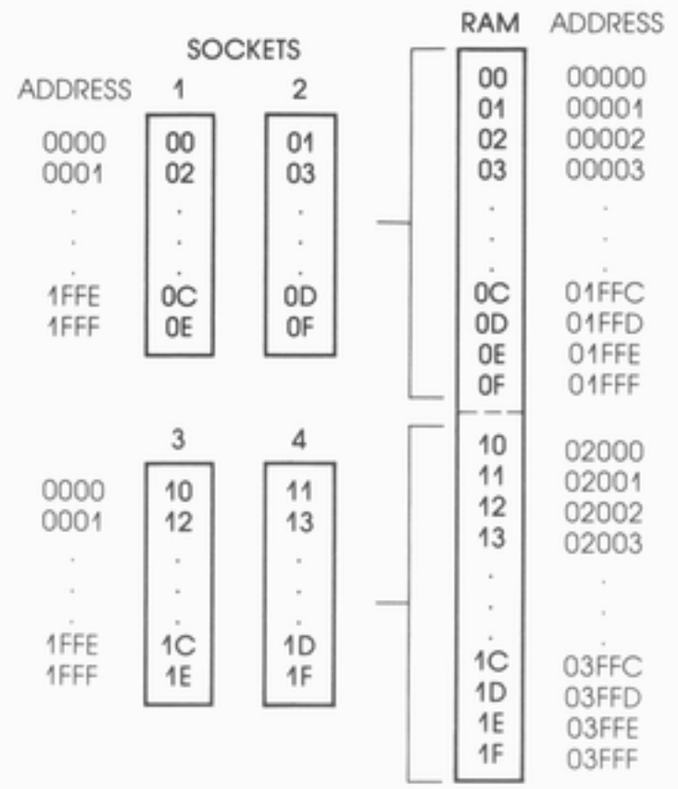
Example

Load two pairs of devices (four devices total):

```
>LW 4<CR>
DEVICE: 1 2 3 4 5 6 7 8
        P P P P X X X X

THE SUMCHECK (TOTAL) IS:  168B

>
```



NOTE

The above illustration represents the loading of two pairs of 64K devices.

Loading Long-Word-Wide Masters

Use the Load Long Word command to load devices that are set up to be read as 32-bit-wide words. This command loads the data from a set of four 8-bit-wide devices into a 32-bit-wide word format. See the following illustration for the order that device data is loaded into RAM. You can load more than one long-word-wide set of 8-bit-wide devices, as long as the total size of all of the devices does not exceed 512K bits. See the following table for the maximum number of pairs that can be loaded at once.

Command Format: LL [<number of masters>]

Procedure	Example Key Sequence
1. Place the sets of 8-bit-wide devices into the sockets, starting with socket 1, and lock them into place. Place the devices of a set in order in adjacent sockets, with the lowest-order device for the set furthest to the left.	
2. Type in the Load Long Word command letters, LL.	LL
3. Type in the number of devices being loaded and press carriage return, or to load the default number of masters, press carriage return without entering the number of masters. The power-up default number of masters is four. (The space after "LL" is optional.)	LL 4<*CR*> or LL<*CR*>

Procedure	Example Key Sequence
4. Check the screen display for the letters that appear below the sockets containing devices. A "P" will appear below any socket containing a device that was loaded successfully and an "F" will appear below any socket containing a device that was not loaded successfully. Also note the sumcheck total which appears below the device information. The sumcheck total should match the sumcheck of any set of devices programmed with this data.	
5. Lift the socket levers and remove the devices.	

Device Size	Maximum Number of Pairs
16K	2
32K	2
64K	2
128K	1

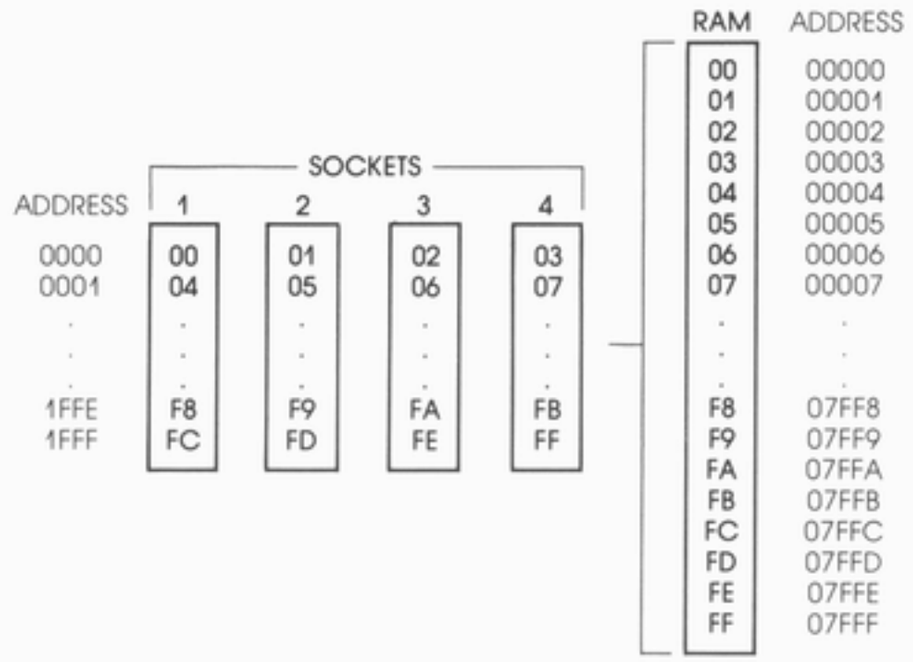
Example

Load one set of devices (four devices total):

```
>LL 4<CR>
DEVICE: 1 2 3 4 5 6 7 8
        P P P P X X X X

THE SUMCHECK (TOTAL) IS:  168B

>
```



NOTE

The above illustration represents the loading of a set of 64K devices.

Programming Devices

The following commands allow you to program devices singly, up to eight at a time, or in sets. The sets can be programmed in eight-bit-wide word format, 16-bit-wide word format or 32-bit-wide word format. Before programming any devices, you must load the data to be programmed into the devices into the programmer's RAM. You can load the data into the programmer's RAM by using a load command to load the data from a master device or by using a download command to download the data from a computer. Instructions on using all of the TRC load commands are presented preceding this subsection and instructions on downloading are presented in the Front Panel Operation section.

Programming Devices from a Single Master

Use the Program Single command to program from one to eight devices with the same data. You can program the devices with programmer RAM starting with the first byte of programmer RAM, or you can specify another range of programmer RAM to be programmed into the devices. You can also select the device address that the first data byte will be written to.

Command Format: P [<start address> <end address> <destination address>]

Procedure	Example Key Sequence
1. Insert the blank device or devices in any of the eight sockets and lock each device into place.	
2. Type in the Program Single command letter, P.	P
3. Type in the first address of RAM that you want programmed into each device, or, to program each device starting with programmer RAM address 0 and device address 0, go to step 6.	P 0
4. Type a space and then type in the last address of RAM that you want programmed into each device. If you entered a start address, you must enter an end address.	P 0 1FFF

Procedure	Example Key Sequence
5. Type a space and then type in the address of the device memory location that you want the first byte of data to be written to (destination address). The power-up default destination address is 0.	P 0 1FFF 0
6. Press carriage return to execute the command. Each device is programmed with the same range of data.	P<*CR*> or P 0 1FFF 0<*CR*>
7. Check the screen display for the letters that appear below the sockets containing devices. A "P" will appear below any socket containing a device that was programmed successfully and an "F" will appear below any socket containing a device that was not programmed successfully. Also note the sumcheck total which appears below the device information. The sumcheck should match the sumcheck of the data in the programmer's RAM.	
8. Lift the socket levers and remove the devices.	

Example

Program six devices with RAM addresses 0 through 1FFF, starting at device address 0:

```
>P 0 1FFF 0<CR>
PROGRAMMING
DEVICE: 1 2 3 4 5 6 7 8
        X P P P X P P P

THE SUMCHECK (TOTAL) IS:  E000

>
```

NOTE

All six devices (placed in sockets 2, 3, 4, 6, 7 and 8) were programmed successfully.

Programming a Set of Devices

Use the Program Set command to program a set of devices with the contents of the programmer's RAM. This command fills each device with successive blocks of programmer RAM, starting with RAM address 0 and device address 0. The programmer fills the leftmost device with the first block of RAM and the next device with the next (consecutive) block of RAM, and so on. See the following table for memory to device data mapping. More than one set of devices can be programmed at once, if more than one set of device data was loaded into RAM.

Command Format: PS

Procedure	Example Key Sequence
1. Insert the set or sets of blank devices into consecutive sockets, starting with socket 1, and lock them into place. (See the table following the example to determine which devices will be programmed with which blocks of RAM.)	
2. Type the Program Set command letters, PS, and press carriage return.	PS<*CR*>

Procedure	Example Key Sequence
3. Check the screen display for the letters that appear below the sockets containing devices. A "P" will appear below any socket containing a device that was programmed successfully and an "F" will appear below any socket containing a device that was not programmed successfully. Also note the sumcheck total which appears below the device information. The sumcheck should match the sumcheck of the data in the programmer's RAM.	
4. Lift the socket levers and remove the devices.	

Example

Program two 128K devices with successive 128K blocks of RAM:

```
>PS<CR>
PROGRAMMING
DEVICE: 1 2 3 4 5 6 7 8
        P P X X X X X X

THE SUMCHECK (TOTAL) IS: 96D4

>
```

Programmer RAM blocks programmed into set devices:

Socket Number	Device Size (no. of bits)					
	16K	32K	64K	128K	256K	512K
1	0000 07FF	0000 0FFF	0000 1FFF	0000 3FFF	0000 7FFF	0000 FFFF
2	0800 0FFF	1000 1FFF	2000 3FFF	4000 7FFF	8000 FFFF	0000 FFFF
3	1000 17FF	2000 2FFF	4000 5FFF	8000 BFFF	0000 7FFF	0000 FFFF
4	1800 1FFF	3000 3FFF	6000 7FFF	C000 FFFF	8000 FFFF	0000 FFFF
5	2000 27FF	4000 4FFF	8000 9FFF	0000 3FFF	0000 7FFF	0000 FFFF
6	2800 2FFF	5000 5FFF	A000 BFFF	4000 7FFF	8000 FFFF	0000 FFFF
7	3000 3FFF	6000 6FFF	C000 DFFF	8000 BFFF	0000 7FFF	0000 FFFF
8	3800 3FFF	7000 7FFF	3000 FFFF	C000 FFFF	8000 FFFF	0000 FFFF

Programming Devices Using a Word-Wide Format

Use the Program Word command to program 8-bit-wide devices using a 16-bit-wide word format. This command treats two 8-bit-wide devices as a single 16-bit- wide device. The programmer programs address 0 of the first device with the first byte of RAM and then address 0 of the second device with the second byte of RAM. It then continues on to program address 1 of the first device with the third byte of RAM, and address 1 of the second device with the fourth byte of RAM, and so on. You can program up to four pairs of word-wide devices using one Program Word command, provided the programmer memory has been loaded with data for all of the pairs. The programmer fills the next pair of devices starting with the next RAM address following the last address copied into the previous pair.

Command Format: PW

Procedure	Example Key Sequence
1. Insert the device pairs into the sockets and lock them into place. The devices in sockets 1 and 2 will be programmed as a 16-bit-wide word pair. Each successive pair of devices will be programmed as a 16-bit-wide pair with the next block of data.	
2. Type the Program Word command letters, PW, and press carriage return.	PW<*CR*>

Procedure	Example Key Sequence
3. Check the screen display for the letters that appear below the sockets containing devices. A "P" will appear below any socket containing a device that was programmed successfully and an "F" will appear below any socket containing a device that was not programmed successfully. Also note the sumcheck total which appears below the device information. The sumcheck should match the sumcheck of the data in the programmer's RAM.	
4. Lift the socket levers and remove the devices.	

Example

Program four sets of 64K devices in 16-bit-wide word format:

```
>PW<CR>
PROGRAMMING
DEVICE: 1 2 3 4 5 6 7 8
        P P P P P P P P

THE SUMCHECK (TOTAL) IS:  96D4

>
```

Programming Devices Using a Long-Word-Wide Format

Use the Program Long Word command to program eight-bit-wide devices using a 32-bit-wide word format. This command treats four 8-bit-wide devices as a single 32-bit-wide device. The programmer programs address 0 of each of the four devices with the first four successive bytes of memory, and then programs address 1 of each of the four devices with the next four successive bytes of memory, and so on. You can program two sets of long-word-wide devices using one Program Long Word command, provided you have loaded two sets of data into the programmer's memory. The second set of devices will be programmed with the block of memory following the last byte of memory that was programmed into the first set of devices.

Command Format: PL

Procedure	Example Key Sequence
1. Insert the sets of devices into the sockets and lock them into place. The devices in sockets 1, 2, 3, and 4 will be programmed as one 32-bit-wide word set and devices in sockets 5, 6, 7, and 8 will be programmed as a second 32-bit-wide word set.	
2. Type the Program Long Word command letters, PL, and press carriage return.	PL<*CR*>

Procedure	Example Key Sequence
3. Check the screen display for the letters that appear below the sockets containing devices. A "P" will appear below any socket containing a device that was programmed successfully and an "F" will appear below any socket containing a device that was not programmed successfully. Also note the sumcheck total which appears below the device information. The sumcheck should match the sumcheck of the data in the programmer's RAM.	
4. Lift the socket levers and remove the devices.	

Example

Program one set of 128K devices in 32-bit-wide word format:

```
>PL<CR>
PROGRAMMING
DEVICE: 1 2 3 4 5 6 7 8
        P P P P X X X X

THE SUMCHECK (TOTAL) IS:  96D4

>
```

Verifying Programmed Devices

Use the following commands to verify device data against data loaded into the programmer's RAM. Each device is automatically verified after it is programmed; however, if you wish to verify devices at another time, you can use these commands. Before you use any of the following verify commands, you must load the programmer's RAM with data which the devices are to be verified against using either a load or a download command. The verify commands are presented in the following order.

Command Name	Command Format
Verify Single	V [<Vcc Selection>]
Verify Set	V S [<Vcc Selection>]
Verify Word	V W [<Vcc Selection>]
Verify Long Word	V L [<Vcc Selection>]

Verifying Devices Against a Single Master

The Verify Single command allows you to verify a single device or up to eight identical devices simultaneously against the master data loaded into memory.

Command Format: V [<Vcc Selection>]

Procedure	Example Key Sequence
1. Load the master data into the programmer`s RAM using either a load or a download command.	
2. Lock the device or devices to be verified into any of the sockets.	
3. Type in the Verify Single command letter, V.	V
4. Select the Vcc high voltage and low voltage setting from the following table and press carriage return. The power-up default is selection 0.	V<*CR*> or V 2<*CR*>

Selection	High Voltage	Low Voltage
0	(manufacturer's recommendation)	
1	5.2 V	4.8 V
2	5.5 V	4.5 V

Procedure	Example Key Sequence
5. Check the screen display for the letters that appear below the sockets containing devices. A "P" will appear below any socket containing a device that was verified successfully and an "F" will appear below any socket containing a device that was not verified successfully. Also note the sumcheck total which appears below the device information. The sumcheck should match the sumcheck produced when the device was programmed.	
6. Lift the socket levers and remove the devices.	

Example

Verify the contents of eight devices:

```
>V 2<CR>
DEVICE: 1 2 3 4 5 6 7 8
        P P P P F P P P

THE SUMCHECK (TOTAL) IS:  E000

>
```

NOTE

The device in socket 5 did not verify successfully against the master data.

Verifying Sets of Devices

The Verify Set command allows you to verify sets of devices (each containing different data) against the master set data stored in the programmer's RAM.

Command Format: VS [<Vcc Selection>]

Procedure	Example Key Sequence
1. Load the master set data into the programmer's RAM using either a load or a download command.	
2. Insert the set(s) of devices in the sockets in the same order that they were programmed originally, starting with socket 1. (See Programming a Set of Devices for the table showing which RAM addresses correspond to which devices.) You may verify a partial set, but each device must be inserted in the socket corresponding to the same RAM data range that the device was programmed with originally.	
3. Type in the Verify Set command letters, VS.	VS
4. Select the Vcc high voltage and low voltage setting from the following table and press carriage return. The power-up default is selection 0.	VS<CR> or VS 2<CR>

Selection	High Voltage	Low Voltage
0	(manufacturer's recommendation)	
1	5.2 V	4.8 V
2	5.5 V	4.5 V

Procedure	Example Key Sequence
5. Check the screen display for the letters that appear below the sockets containing devices. A "P" will appear below any socket containing a device that was verified successfully and an "F" will appear below any socket containing a device that was not verified successfully. Also note the sumcheck total which appears below the device information. The sumcheck should match the sumcheck produced when the devices were programmed. The sumcheck will always be the total for the entire set, even if you verified only a partial set.	
6. Lift the socket levers and remove the devices.	

Example

Verify the contents of a set of four devices:

```
>VS 2<CR>
DEVICE: 1 2 3 4 5 6 7 8
        P P P P X X X X

THE SUMCHECK (TOTAL) IS:  1357

>
```

Verifying Word-Wide Devices

Use the Verify Word command to verify devices against data that is loaded into RAM in a 16-bit-wide word format. This command compares two 8-bit-wide devices to 16-bit-wide data in the programmer's RAM. Address 0 of the first device is compared with the first byte of RAM, address 0 of the second device with the second byte of RAM, address 1 of the first device with the third byte of RAM, and so on. More than one pair of devices can be compared, provided the master data for all pairs to be verified is loaded into the programmer's RAM.

Command Format: VW [<Vcc Selection>]

Procedure	Example Key Sequence
1. Load the master data into the programmer's memory using either a load or a download command.	
2. Insert the pairs of devices (up to 4) in the sockets, starting with socket 1. Place the devices in the sockets in the same order that they were programmed (with the high-order device to the right of the low-order device for the pair).	
3. Type in the Verify Word command letters, VW.	VW

Procedure	Example Key Sequence
4. Select the Vcc high voltage and low voltage setting from the following table and press carriage return. The power-up default is selection 0.	VW<CR> or VW 2<CR>

Selection	High Voltage	Low Voltage
0	(manufacturer's recommendation)	
1	5.2 V	4.8 V
2	5.5 V	4.5 V

5. Check the screen display for the letters that appear below the sockets containing devices. A "P" will appear below any socket containing a device that was verified successfully and an "F" will appear below any socket containing a device that was not verified successfully. Also note the sumcheck total which appears below the device information. The sumcheck should match the sumcheck produced when the devices were programmed.

6. Lift the socket levers and remove the devices.

Example

Verify the contents of three pairs of devices:

```
>VW 2<CR>
DEVICE: 1 2 3 4 5 6 7 8
        P P P P P P X X

THE SUMCHECK (TOTAL) IS:  1357

>
```

Verifying Long-Word-Wide Devices

Use the Verify Long Word command to verify device data against data that is loaded into RAM in a long word (32-bit-wide) format. This command compares four 8-bit-wide devices to 32-bit-wide data in the programmer's RAM. Address 0 of the first device is compared with RAM address 0, address 0 of the second device is compared with RAM address 1, and so on. Two sets of devices can be verified simultaneously, provided the data for all devices is loaded into the programmer's RAM.

Command Format: VL [<Vcc Selection>]

Procedure	Example Key Sequence
1. Load the master data into the programmer's RAM using either a load or a download command.	
2. Insert the set or sets of devices into the sockets, starting with socket1. Place the devices of each set in order, with the device containing the lowest-order bits furthest to the left for each set.	
3. Type in the Verify Long Word command letters, VL.	VL

Procedure	Example Key Sequence
4. Select the Vcc high voltage and low voltage setting from the following table and press carriage return. The power-up default is selection 0.	VL<*CR*> or VL 2<*CR*>

Selection	High Voltage	Low Voltage
0	(manufacturer's recommendation)	
1	5.2 V	4.8 V
2	5.5 V	4.5 V

5. Check the screen display for the letters that appear below the sockets containing devices. A"P" will appear below any socket containing a device that was verified successfully and an "F" will appear below any socket containing a device that was not verified successfully. Also note the sumcheck total which appears below the device information. The sumcheck should match the sumcheck produced when the devices were programmed.

6. Lift the socket levers and remove the devices.

Example

Verify the contents of one set of four devices:

```
>VL 4<CR>
DEVICE: 1 2 3 4 5 6 7 8
        P P P P X X X X

THE SUMCHECK (TOTAL) IS:  1357

>
```

Editing Memory

The following commands allow you to edit data that has been loaded into the 280 Set Programmer's RAM. With these commands, you can display the data, edit individual addresses, fill a segment of memory with data, insert new data, delete data, copy data, and search memory for up to 16 characters of data. The commands are presented in the following order:

Command Name	Command Format
Memory Display	M <start address> <end address>
Memory Modify	M <start address>
Memory Fill	M <start address> <end address> <data1> [<data2...data8>]
Insert	I <start address> <data1> [<data2...data8>]
Delete	D <start address> <end address>
Transfer (Copy)	T <start address> <end address> <destination address> Memory
Search	MS <start address> <end address> <data1> [<data2...data8>]

You may also want to use the Sumcheck (Total) command to calculate a sumcheck of the data stored in the programmer's memory after you have finished editing and are ready to program devices. The Sumcheck (Total) command is explained following this subsection.

Displaying Memory

Use the Memory Display command to display the contents of RAM on the terminal screen. The RAM addresses are arranged on the screen in a table format, with each row containing 16 bytes of consecutive data (see example). To determine the address of a byte of RAM, add the 5-character RAM address shown at the very left of the line containing the data to the 2-character address shown at the top of the column containing the data. The ASCII-CODE column on the far right contains the ASCII equivalent of the hexadecimal data on each line.

Command Format: M <start address> <end address>

Procedure	Example Key Sequence
1. Type the Memory Display command letter, M.	M
2. Type in the address of the first byte of RAM that you want to display on the screen.	M 0
3. Type a space, then the address of the last byte of RAM that you want to display on the screen followed by a carriage return.	M 0 1F<*CR*>
4. Type CTRL-S to halt the display and CTRL-Q to resume the display.	

Example

Display the contents of RAM addresses 0 through 1F:

```
>M 0 1F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
00000 32 1A 5B 47 43 52 03 4B CB 20 54 33 4A 11 20 52 2.[GCR.K. T3J. R
00010 52 20 11 4A 33 54 20 CB 4B 03 AA 52 43 47 4B 1A R.J3T .K..RCGK..
>
```

NOTE

To determine the memory address of a data byte, add the column heading of the data byte column to the row heading (5-character hexadecimal number at the far left of each line).

A period under the ASCII-CODE heading represents hexadecimal data without an ASCII equivalent.

Modifying Single Memory Locations

Use the Memory Modify command to display and modify the contents of RAM locations one-by-one. The Memory Modify command causes the programmer to enter an editing mode which allows you to display the contents of RAM addresses one-by-one and type in new data for the address displayed. To exit Memory Modify mode and return to the > prompt, press *<ESC>*.

Command Format: M <start address>*<CR>*
<new data>*<CR>*
.
.
.
<ESC>

Procedure	Example Key Sequence
1. Type the Memory Modify command letter, M.	M
2. Type in the address of the first memory location you wish to modify and press carriage return.	M 6*<CR>*
3. After the RAM address and its contents have been displayed, type in the new data and press carriage return to accept it. The new data will appear to the right of the old data. If you do not wish to modify the address displayed, but wish to display the next RAM address for modification, press carriage return without entering new data.	E5*<CR>* or *<CR>*

Procedure	Example Key Sequence
4. To modify the next memory location, return to step 3.	
5. To exit Memory Modify mode and return to the > prompt, press <*ESC*>. If you press <*ESC*> after typing in new data but before pressing carriage return to accept the new data, the new data for the last address displayed will be ignored.	

Example

Enter Memory Modify mode starting at address 6 and type in new data for addresses 6, 7, and 8:

```
>M 6<CR>
00006  F1   E5<CR>
00007  FF   21<CR>
00008  00   21<CR>
00009  43   <ESC>
>
```

Filling a Segment of Memory

Use the Memory Fill command to fill (replace) a segment of RAM with up to eight bytes of new data. The original data in these RAM locations are overwritten with the new data.

Command Format: M <start address> <end address> <data1> [<data2>...<data8>]

Procedure	Example Key Sequence
1. Type the Memory Fill command letter, M.	M
2. Type in the address of the first byte of RAM you want to fill with new data.	M 20
3. Type a space and then type in the address of the last byte of RAM you want to fill with new data.	M 20 2F
4. Type a space and then type in the data to be input into the specified memory range. Separate each hexadecimal byte with a space. You can insert up to eight hexadecimal bytes (or eight ASCII characters) with one Memory Fill command. If you input more data than can fit in the range specified, the excess data is ignored.	M 20 2F 11 22<*CR*>

Example

Display memory contents before fill:

```
>M 20 3F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
00020 FF FF FF FF EE EE EE EE DD DD DD DD CC CC CC CC ................
00030 BB BB BB BB AA AA AA AA 99 99 99 99 88 88 88 88 ................
```

Fill memory:

```
>M 20 2F 11 22<CR>
```

Display memory contents after fill:

```
>M 20 3F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
00020 11 22 11 22 11 22 11 22 11 22 11 22 11 22 11 22 ."."."."."."."."
00030 BB BB BB BB AA AA AA AA 99 99 99 99 88 88 88 88 ................
```

Inserting New Data

Use the Insert command to insert new data into RAM. The original data is not overwritten by this command, as it is with the Memory Modify and the Memory Fill commands, but is shifted down to accommodate the new data.

Command Format: I <start address> <data1> [<data2>...<data8>]

Procedure	Example Key Sequence
1. Type the Insert command letter, I.	I
2. Type in the starting address where you want the new data to be inserted.	I 5
3. Type a space and then type in the new data followed by a carriage return. Separate each hexadecimal byte with a space. You can insert up to eight hexadecimal bytes (or eight ASCII characters) with one Insert command.	I 5 11 11 11 11<*CR*>

Example

Display memory contents before insertion of new data:

```
>M 0 1F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
00000 FF FF FF FF EE EE EE EE DD DD DD DD CC CC CC CC  ................
00010 BB BB BB BB AA AA AA AA 99 99 99 99 88 88 88 88  ................
```

Insert new data:

```
>I 5 11 11 11<CR>
```

Display memory contents after insertion of new data:

```
>M 0 1F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
00000 FF FF FF FF EE 11 11 11 EE EE EE DD DD DD DD CC  ................
00010 CC CC CC BB BB BB BB AA AA AA AA 99 99 99 99 88  ................
```

Deleting Data

Use the Delete command to delete data bytes from memory. Data following the deleted addresses are shifted up to fill in the deleted addresses. "FF" is filled in the vacated addresses at the end of RAM.

Command Format: D <start address> [<end address>]

Procedure	Example Key Sequence
1. Type the Delete command letter, D.	D
2. Type in the address where you want to begin deleting data.	D 14
3. To delete a single byte of data, press carriage return without entering an end address. To delete a range of data, type a space, then the address of the last memory byte that you want to delete followed by a carriage return.	D 4<*CR*> or D 4 7<*CR*>

Transferring (Copying) Memory

Use the Transfer command to copy a block of RAM data from one RAM location to another. This command copies the block specified by the start address and end address to the destination address. The original data starting at the destination address is overwritten by the copied block.

Command Format: T <start address> <end address> <destination address>

Procedure	Example Key Sequence
1. Type the Transfer command letter, T.	T
2. Type in the address of the first byte of RAM to be copied.	T 0
3. Type a space and then type in the address of the last byte of RAM to be copied.	T 0 F
4. Type a space and then type in the address that you want the first byte of the specified data range to be copied to, or the destination address, and press carriage return. The data will be copied into consecutive RAM addresses, starting with the destination address.	T 0 F 25<*CR*>

Example

Display memory contents before transfer:

```
>M 0 3F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
00000 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99  ................
00010 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77  WWWWWWWWWWWWWWWW
00020 66 66 66 66 66 44 44 44 44 44 44 44 44 44 44 44  FFFFFDDDDDDDDDDD
00030 44 44 44 44 44 11 11 11 11 11 11 11 11 11 11 11  DDDDD...........
```

Transfer data:

```
>T 0 F 25<CR>
```

Display memory contents after transfer:

```
M 0 3F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
00000 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99  ................
00010 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77  WWWWWWWWWWWWWWWW
00020 66 66 66 66 66 99 99 99 99 99 99 99 99 99 99 99  FFFFF...........
00030 99 99 99 99 99 11 11 11 11 11 11 11 11 11 11 11  ................
```

Searching Memory

Use the Memory Search command to find a block of data up to eight hexadecimal bytes long.

Command Format: MS <start address> <end address> <data1> [<data2>...<data8>]

	Procedure	Example Key Sequence
1.	Type the Memory Search command letters, MS.	MS
2.	Type in the address of the memory location where you want the search to begin.	MS 0
3.	Type a space and then type in the memory address where you want the search to end.	MS 0 3F
4.	Type a space and then type in the data that you want to search for and press carriage return. Separate each hexadecimal byte with a space. You can type in up to eight hexadecimal bytes (eight ASCII characters).	MS 0 3F 12 34<CR>

Example

Search memory block 0 through 3F for the hexadecimal data 12 34:

```
>MS 0 3F 12 34<CR>
MEMORY MATCH AT ADDRESS 000038
>
```

Display memory block searched:

```
>M 0 3F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
00000 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77   WWWWWWWWWWWWWWWW
00010 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77   WWWWWWWWWWWWWWWW
00020 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77   WWWWWWWWWWWWWWWW
00030 77 77 77 77 77 77 77 77 12 34 77 77 77 77 77 77   WWWWWWWW.4WWWWWW
```

Sumchecking Data

Use the following commands to cause the programmer to sumcheck (total) or checksum (EXOR) the data stored in the programmer's RAM. These commands are useful for verifying that the data stored in RAM is the same as data programmed into a device or data downloaded through the serial port.

Performing a Sumcheck (Total)

Use the Sumcheck (Total) command to calculate a hexadecimal summation of data stored in the programmer's RAM. This sumcheck can be used to verify that data was downloaded to the programmer correctly, to sumcheck only a portion of memory, or to produce a current sumcheck after you have edited memory. A sumcheck is automatically performed after most programming operations, so you should not need to use this command during standard production.

Command Format: CT <start address> <end address> [<destination address>]

	Procecure	Example Key Sequence
1.	Type in the Sumcheck (Total) command letters, CT.	`CT`
2.	Type in the address of the first byte of RAM to be checked by the sumcheck operation.	`CT 0`
3.	Type a space and then type in the address of the last byte of RAM to be checked.	`CT 0 3FFF`

Procedure	Example Key Sequence

4. To perform the sumcheck without storing the sumcheck value in RAM, press carriage return, or to store the sumcheck value in RAM, type a space and then type in the RAM address that you want the sumcheck value to be stored in and press carriage return. If you store the sumcheck value in RAM, it will overwrite any data previously stored in the RAM locations which the sumcheck is written to. Make sure that there is no data that you want to retain located in the addresses that the sumcheck will be written to.
 	
The sumcheck (total) is calculated and then displayed on the screen. The sumcheck is also stored in memory if you specified a destination address.
	`CT 0 3FFF<CR>`
	or
	`CT 0 3FFF 4000<CR>`

Example

Calculate the sumcheck total of memory addresses 0 through 3FFF and store the sumcheck value in address 4000:

```
>CT 0 3FFF 4000<CR>
THE SUMCHECK (TOTAL) IS:  0201
>
```

NOTE

The two-byte sumcheck is stored in memory addresses 4000 and 4001, with the low-byte (01 in above example) stored in 4000 and the high-byte (02 in above example) stored in 4001.

Performing a Checksum (EXOR)

Use the Checksum (EXOR) command to calculate an exclusive-OR checksum of RAM. You may wish to use this command instead of the sumcheck (total) command if data being downloaded to the programmer includes an exclusive-OR checksum, or if you prefer to work with an exclusive-OR summation. The exclusive-OR checksum is a 1-byte (two's complement) hexadecimal value representing the sum of the data range specified. Each bit is added separately and the carry for each bit is discarded. For example,

```
  0000 0001
+1111 1111
-----------
  1111 1110
```

Command Format: CX <start address> <end address> [<destination address>]

Procedure	Example Key Sequence
1. Type in the Checksum (EXOR) command letters, CX.	CX↵
2. Type in the address of the first byte of RAM to be checked by the checksum operation.	CX 0
3. Type a space and then type in the address of the last byte of RAM to be checked by the checksum operation.	CX 0 3FFF

Procedure	Example Key Sequence
4. To perform the checksum operation without storing the checksum value in RAM, press carriage return. To store the checksum in RAM, type a space, then the RAM address that you want the checksum to be stored in followed by a carriage return. If you store the checksum value in RAM, it will overwrite any data previously stored in that RAM location. The checksum (EXOR) value is calculated and then displayed on the screen. The checksum is also stored in RAM if you specified a destination address.	CX 0 3FFF<*CR*> or CX 0 3FFF 4000<*CR*>

Example

Calculate the checksum of memory addresses 0 through 3FFF and store the checksum value in address 4000:

```
>CX 0 3FFF 4000<CR>
THE CHECKSUM (EXOR) IS:  FE
>
```

Computer Remote Control

Introduction

The 280 Set Programmer can be controlled via a remote host computer using the Computer Remote Control (CRC) commands described in this section. The Computer Remote Control commands were designed to be incorporated into a software program, or driver, which would allow an operator to control the 280 Set Programmer using the software program. The driver generates and sends commands to the programmer which executes the commands and returns a response character, and in some cases also data, which the driver then reacts to and uses to generate messages and prompts for the operator.

The CRC, or driver, commands are ASCII characters which are summarized in the command summary table and described in more detail in the pages that follow. The commands are grouped into the following subsections:

- ☐ ENTERING AND EXITING COMPUTER REMOTE CONTROL MODE—This subsection explains how to transfer control of the programmer to a remote host computer and return control to the front panel of the programmer.

- ☐ VERIFYING PROPER COMMUNICATION—This subsection describes the commands used to verify that proper communication has been established between the computer and the programmer.

- ☐ PROGRAMMING OPERATIONS—This subsection describes the commands used to set parameters prior to performing programming operations, load data into RAM, program blank devices, test devices and verify that devices were programmed properly.

- ☐ TRANSFERRING DATA—This subsection describes the commands used to upload or download data to or from the remote host computer.

- ☐ INQUIRING ABOUT OPERATING AND ERROR STATUS—This subsection describes the commands used to inquire about parameters and options selected and error status.
- ☐ DATA TRANSLATION FORMATS—This subsection describes the five data translation formats available for the 280 Set Programmer.

Symbols and Conventions

The following is a list of symbols and conventions used in this section of the manual.

A	Capital letters in a command must be sent to the programmer through the serial port to execute the command.
h	A lower-case h represents a hexadecimal digit.
n	A lower-case n represents a decimal digit.
ff	Two lower-case fs represent the family code of of the device.
pp	Two lower-case ps represent the pinout code of the device.
<CR>	This symbol represents a carriage return, which must follow each command entry.
<ESC>	This symbol represents the escape key.

Computer Remote Control Command Summary

Command Format	Description	Command Format	Description
M	Enter CRC mode (automatic setup)	n23]	Set verify pass number
Z	Exit CRC mode	hhll18]	Set Vcc level
<CR>	Execute command	nV	Verify device(s)
<ESC>	Abort operation		
		n=	Enable timeout
hhU	Set nulls	hhhhhhhhW	Set I/O address offset
H	No operation	nnA	Select data translation format
		I	Input data
ffpp@	Set family and pinout codes	O	Output data
hhhhhh:	Set begin device address	C	Compare data
hhhhhh<	Set begin RAM address		
hhhhhh;	Set block size	[	Family and pinout codes inquiry
		CC]	Electronic ID family and pinout codes inquiry
nn22]	Convert to 16-bit mode		
		CD]	Electronic ID code inquiry
L	Load master device(s)	R	Respond with device parameters
hh^	Clear RAM with data hh		
		/	Number of devices failed
nP	Program device or set	X	Error code inquiry
nT	Test for illegal bits	G	Software configuration
B	Blank check all devices		
nS	Perform sumcheck		

Response Characters

The response characters, summarized in the table below, are characters that the programmer sends to the host computer after attempting to execute a command. The programmer's response to a command will always contain a response character followed by a carriage return. In addition, the response may contain data and a line feed and nulls (ASCII character 00). Whether or not the response contains a line feed or nulls is dependent upon the null count setting. How to set the null count is described in the Verifying Proper Communication subsection.

Programmer Response Characters

Character	Name	Description
>	Prompt	Sent upon entering Computer Remote Control mode, when *<ESC>* halts a command, or when the programmer successfully executes a command. The programmer then transmits a carriage return.
F	Fail	Sent when the programmer fails to execute a command. The programmer then transmits a carriage return.
?	Question	Sent when the programmer does not understand a command or the command was invalid. The programmer then transmits a carriage return.

Using PROMlink™ to Operate the 280

If you are using PROMlink™ as the driver program to operate the 280, you must set the programmer's serial port (I/O) settings to match PROMlink's™ I/O settings before entering CRC mode. You cannot use the command that will automatically select the proper communications protocol (M<CR>) when interfacing with PROMlink™. Refer to the Front Panel Operation section for instructions on how to set the 280's serial port (I/O) settings.

If your version of PROMlink™ does not support the 280 Set Programmer, you can still use PROMlink™ to operate the 280 by selecting the Model 120/121 as the programmer type.

Aborting an Operation

Any of the Computer Remote Control commands can be aborted by pressing *<ESC>*.

Entering and Exiting Computer Remote Control Mode

The following procedures describe how to enter Computer Remote Control mode and return control of the programmer to the front panel.

Entering Computer Remote Control Mode

Procedure	280 Displays
1. Connect the programmer to the computer as described in Establishing a Serial Communications Link in the Front Panel Operation section of this manual.	
2. Power up the programmer by flipping the power switch on the back of the programmer up to the ON position.	SELF TESTING . . . . .
3. When the self test is complete and the display reads LOAD FROM MASTER, set the communications protocol of the programmer to match the communications protocol of the host computer (see the Front Panel Operation section for instructions on setting the 280 port settings).	
4. Scroll through the main menu to RS232 PORT (if necessary) and press ENTER.	RS232 PORT COMPUTER CONTROL

Procedure	280 Displays
5. Scroll through the RS232 PORT menu (if necessary) to COMPUTER CONTROL and press ENTER.	COMPUTER CONTROL
6. If you are using PROMlink™, press any key on the computer keyboard to establish communication with the programmer. If you are using a driver other than PROMlink™, see the driver instructions for entering the driver program. A command prompt will appear if the port settings were set correctly and the programmer and computer are communicating. The programmer will display COMPUTER CONTROL until control of the programmer is returned to the front panel.	COMPUTER CONTROL

NOTE

If you are using the CRC commands interactively, you can type M<CR> on the computer keyboard to cause the programmer to automatically select the correct communications protocol and establish communications. If you use this command, you do not have to set the programmer port settings before entering CRC mode. If you are writing your own driver, you may wish to take advantage of this command; however, you cannot use this command to enter CRC mode using PROMlink™.

Exiting Computer Remote Control Mode

You can exit Computer Remote Control mode by using the appropriate exit command specified by the CRC driver program, or you can press any programmer key. If you are using the CRC commands interactively, you can type Z<CR> on the computer keyboard to exit CRC mode. If you are writing your own driver, the driver's exit command must send a "Z<CR>" to the programmer to cause it to exit CRC mode. The user should still be able to exit CRC mode by pressing a programmer key when using a driver.

After exiting CRC mode, the programmer displays

```
RS232 PORT
```

You can then continue to operate the programmer using the front panel keys.

Verifying Proper Communication

The first thing that the driver program should do is verify that the computer and the programmer are communicating properly. To do this, use the following commands.

Command Format	Command Name	Data Output to Host	Description
hhU	Set Nulls	none	Sets the number of nulls (hh) sent to the host computer after the response character and carriage return, and enables line feeds. Values 0 through FE enable line feeds and cause nulls to be sent. The default value, FF, causes no nulls and no line feeds to be sent.
H	No Operation	none	Causes the programmer to return a prompt (>), indicating that communications are established.

Programming Operations

Use the following commands to set the family and pinout code, set programming parameters and options, load device data into memory, program devices, test devices and verify programmed devices.

Command Format	Command Name	Data Output to Host	Description
Setting Programming Parameters and Options			
ffpp@	Set Family and Pinout Codes	none	Sets the family (ff) and pinout (pp) codes for the devices to be loaded, programmed or verified.
hhhhhh:	Set Begin Device Address	none	Sets the first device address from which or to which data is to be transferred. The default begin device address is 0.
hhhhhh<	Set begin RAM Address	none	Sets the first RAM address to be used for data transfers. The default begin RAM address is 0.
hhhhhh;	Set Block Size	none	Sets the block size (number of bytes) to be used in data transfers. The default block size is the word limit of the device times the set size. To reset the block size to the default, enter 0.
nn22]	Convert to 16-bit Mode	none	Causes the 280 to load, program and verify each pair of 8-bit-wide devices using a 16-bit-wide word format when nn=16. To return the programmer to 8-bit-wide mode, set nn equal to 8. See the Terminal Remote Control section for an explanation of memory mapping when using 16-bit wide (word wide) devices.

Command Format	Command Name	Data Output to Host	Description
Loading Master and Clearing RAM			
L	Load Master	none	Loads data into RAM from a master device or set of masters.
hh^	Clear RAM	none	Clears (fills) all of RAM with data hh. The default clear data is 00.
Programming Devices			
nP	Program Device	none	Programs and verifies a single device or gangs of identical devices if n=1, or programs and verifies sets of n devices if n is 2 through 8. The program sequence in CRC mode does not include a blank check or an illegal bit test. The default set size is one device.
Testing Devices			
nT	Test for Illegal Bits	none	Tests single or gangs of devices (if n=1) or sets of n devices for illegal bits (programmed bits that do not exist in RAM). The default set size is one device (single or gang test).
B	Blank Check	none	Checks the installed devices for programmed bits.
nS	Sumcheck	hhhh	Calculates the sumcheck of the device in socket n; if n=0, calculates the sumcheck of RAM data from begin RAM address to either the word limit of the device type selected or the block size (if selected).

Command Format	Command Name	Data Output to Host	Description
Verifying Devices			
n23]	Set Verify Pass Number	none	Sets the number of verify passes (n) to be made. The number n can be either 1 or 2.
hhll18]	Set Vcc Level	none	Sets the high-pass (hh) and low-pass (ll) verify Vcc levels in tenths of volts. The allowable high-pass range is 50 to 60 and the allowable low-pass range is 40 to 50.
nV	Verify Device	none	Compares RAM data with programmed devices or sets of n devices. The default set size is one device.

Transferring Data

Use the following commands to input data to the programmer from the host computer or output data from the programmer to the host computer.

Command Format	Command Name	Data Output to Host	Description
n=	Enable Timeout	none	Enables the I/O timeout if n is set to any value other than 0. Timeout is disabled when n=0, which is the default.
hhhhhhhhW	Set I/O Address Offset	none	Specifies an address to be subtracted from all addresses input to the programmer and added to all addresses output from the programmer. This allows the adjusting of RAM addresses to the address range of larger memories. The default address offset is the first address received (input).
cnnA	Select Data Translation Format	none	Selects the instrument control code (c) and data translation format (nn) used for input and output of data. See the Data Translation Formats subsection at the end of the CRC section for details on the instrument control code and data translation formats and a list of available data translation formats and their codes. The default format is Intel Intellec 8/MDS, format 83, and the default control code is 0.

Command Format	Command Name	Data Output to Host	Description
I	Input Data	none	Instructs the programmer to accept formatted data from the host computer.
O	Output Data	hhhhh...	Translates data into the selected format and outputs this data to the computer.
C	Compare Data	none	Compares data in RAM with the data sent through the serial port from the host computer.

Inquiring About Operating and Error Status

Use the following commands to cause the programmer to output the status of the current device(s), the electronic ID settings, error status, and the programmer software configuration.

Command Format	Command Name	Data Output to Host	Description
[	Family and Pinout Codes Inquiry	ffpp	Outputs the family (ff) and pinout (pp) codes currently in effect.
CC	Electronic ID Family and Pinout Codes Inquiry	ffpp	Ouputs the electronic ID family (ff) and pinout (pp) codes currently in effect.
CD	Electronic ID Code Inquiry	hh	Ouputs the electronic identifier code of the installed device.
R	Respond	wl/ws/n	Outputs the status of the device selected by the current family and pinout codes. wl = word limit (3 or 4 hexadecimal digits) ws = word size (4 or 8 bits) n = VOL or VOH (1=VOL and 0=VOH)
/	Number of Devices Failed	xx-yy	Ouputs the number of devices that failed (xx) and the number of devices in the sockets (yy).

Command Format	Command Name	Data Output to Host	Description
X	Error Code Inquiry	h1...h16	Outputs the last 16 error codes which occurred. This command also clears the error codes from memory. See the Error Messages section for explanations of the meanings of the error codes.
G	Software Configuration	hhhh	Outputs the configuration number of the programmer's software.

Data Translation Formats

Introduction

This subsection defines the data translation formats available for the 280 Set Programmer. The 280 Set Programmer is capable of interfacing with most RS232 serial equipment employing a data translation format described in this subsection.

Any of the five available data translation formats can be selected from the front panel for upload and download operations. In Computer Remote Control mode a 2-digit code assigned to each data translation format must be entered into the programmer through the serial port to send or receive data in that format. In addition to the data translation format code, there is a 1-digit instrument control code that must be entered in CRC mode. This instrument control code specifies control characters to be transmitted to, or received from, peripheral instruments.

Available Data Translation Formats

The following is a quick reference list of data translation formats available for the 280 Set Programmer and their corresponding codes. The formats are listed in alphabetical order. Each format is described in detail in the pages that follow.

Format	Code
Intel Intellec 8/MDS	83
Intel MCS-86 Hexadecimal Object	88
Motorola Exorciser	82
Motorola Exormax	87
Tektronix Hexadecimal	86

Data Verification

For data verification, the 280 calculates a sumcheck of all data sent to or from the programmer. At the end of a successful input operation, the programmer will display the sumcheck of all data transferred. It will also compare any received sumcheck fields with its own calculation. If the two agree, the programmer will display the sumcheck; a mismatch will produce an error message. Output data is always followed by a sumcheck field which may be printed on disk or tape for use in subsequent input operations.

Description of Format Codes

Each format is assigned a 2-digit data translation format code which you must enter when transferring data in CRC mode to tell the programmer which format to use. In addition to this code, a 1-digit instrument control code may be used to specify control characters for peripheral equipment. The codes must be formatted as follows: xyy, where "x" is the instrument control code and "yy," the format code. If no codes are entered into the programmer, the current default values will be in effect.

The following list shows the instrument control codes, with the corresponding 280 action.

Control Code	Programmer Action
0	Sends data immediately and continuously until acknowledging a "reader off" code. It will then stop sending data until receiving a "reader on" code. Sending no control codes results in normal, uninterrupted transmission.
1	Sends "reader on" (ASCII DC1/Hex 11) when ready to receive data, and "reader off" (ASCII DC3/Hex 13) when all data is received. Also sends "punch on" (ASCII DC2/Hex 12) before sending data, and "punch off" (ASCII DC4/Hex 14) after sending data.
2	Sends data after acknowledging a "reader on" (ASCII DC1/Hex 11), and stops sending data after acknowledging a "reader off" ASCII DC3/Hex 13).

Leader and Null Output

A leader is a string of characters that is attached to the beginning and end of a data file. It is used to separate different files from one another and allows extra room which may be necessary for loading and unloading the data medium to or from equipment. For the 280 Set Programmer, the leader is sent at the beginning and end of a data output operation. With one exception, this leader will always be comprised of a carriage return, a line feed, and 50 nulls in succession.

Null count is the number of null characters in the string of characters between each record or line within a file. What actually comprises a data record depends upon the format that is being used. Records and lines can basically be thought of as separations of data within a file.

Null count is a parameter which can be defined by the 280 user for use with printers with a slow carriage return response time. The number of nulls can be set to any value from zero to 254 decimal (FE hexadecimal). The string of characters actually sent between each and every record or line of the file includes a carriage return, a line feed, and the number of nulls defined by the null count, with one exception. When the user defines the null count to be equal to the value of "FF" hexadecimal (or 255 decimal), the leader is made up of a solitary carriage return (no line feed and no nulls). Also, the string separating the records of the file is a carriage return (no line feeds and no nulls).

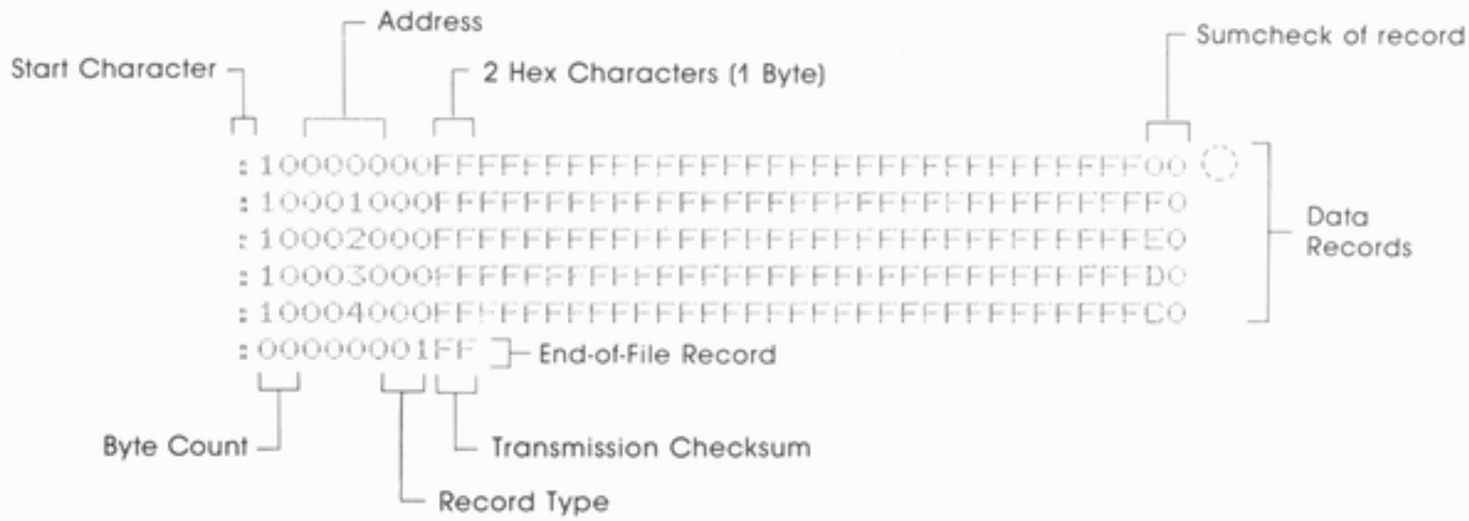
Parity for the beginning and end leader is the same as the parity for the data within the file. The same is true for the carriage return, line feed and nulls separating the records or lines of the file. They have the same parity as the data.

Intel Intellec 8/MDS Format, Code 83

Intel data records begin with a 9-character prefix and end with a 2-character suffix. The byte count must equal the number of data bytes in the record.

The figure represents a series of valid data records. Each record begins with a colon, which is followed by a 2-character byte count. The four digits following the byte count give the address of the first data byte. Each data byte is represented by two hexadecimal digits; the number of data bytes in each record must equal the byte count. Following the data bytes of each record is the sumcheck (the two's complement, in binary, of the preceding bytes, including the byte count, address and data bytes) expressed in hexadecimal.

The end-of-file record consists of the "colon" start character, the byte count (equal to "00"), the address, the record type (equal to "01") and the sumcheck of the record.



LEGEND

◌ Nonprinting Carriage Return, line feed, and nulls

Intel MCS-86 Hexadecimal Object, Code 88

The Intel 16-bit Hexadecimal Object file record format has a 9-character (4-field) prefix that defines the start of record, byte count, load address, and record type and a 2-character sumcheck suffix. The figure illustrates the sample records of this format.

The four record types are:

00 = data record
01 = end record (signals end of file)
02 = extended address record (added to the offset to determine the absolute destination address)
03 = start record (ignored during input and not sent during output by Data I/O translator firmware)

Record type 00, data record, begins with the colon start character. The colon is followed by the byte count (in hexadecimal notation), the address of the first data byte, and the record type (equal to "00"). These parameters are followed by the data bytes. The sumcheck follows the data bytes and is the two's complement (in binary) of the preceding bytes in the record, including the byte count, address and data bytes.

Record type 01, the end-of-file record, also begins with the colon start character. The colon is followed with the byte count (equal to "00"), the address (equal to "0000"), the record type ("01") and the sumcheck, "FF".

Record type 02, the extended address record, defines bits 4 to 19 of the segment base address. It can appear randomly anywhere within the object file and in any order; i.e., it can be defined such that the data bytes at high addresses are sent before the bytes at lower addresses. The following example illustrates how the extended address is used to determine a byte address.

LEGEND

◌ Nonprinting Carriage Return, line feed, and nulls

Problem: Find the address for the first data byte for the following file.

```
: 02 0000 02 1230 BA
: 10 0045 00 55AA FF .....BC
```

Solution:

Step 1: Find the record address for the byte. The first data byte is 55. Its record address is 0045 from above.

Step 2: Find the offset address. The offset address is 1230 from above.

Step 3: Shift the offset address one place left, then add it to the record address, like this:

```
  offset address    1230     (upper 16 bits)
+ record address    0045     (lower 16 bits)
                    12345    (20-bit address)
```

The address for the first data byte is therefore 12345.

NOTE

Always specify the address offset when using this format, even when the offset is zero.

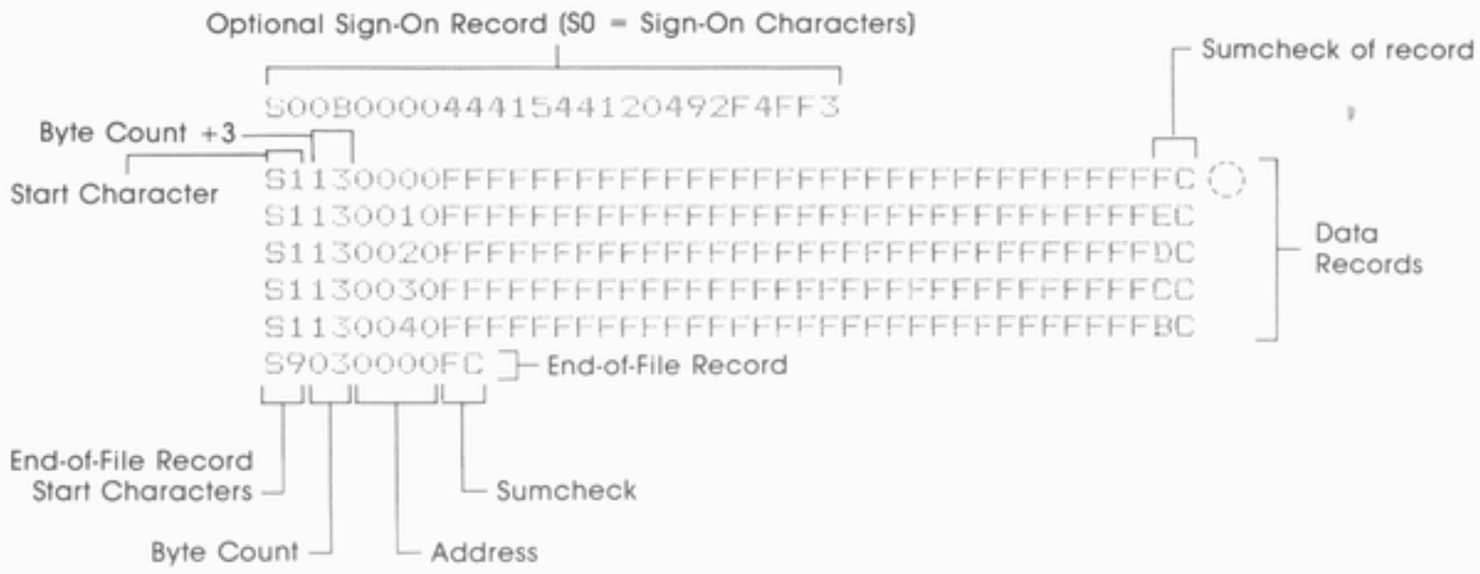
During output translation, the firmware will force the record size to 16 (decimal) if the record size is specified greater than 16. There is no such limitation for record sizes specified less than 16.

Motorola Exorciser Format, Code 82

Motorola Exorciser data files may begin with an optional sign-on record, which is initiated by the start characters "S0." Valid data records start with an 8-character prefix and end with a 2-character suffix. The figure shows a series of valid Motorola data records.

Each data record begins with the start characters "S1"; the programmer will ignore all earlier characters. The third and fourth characters represent the byte count (which expresses the number of data bytes), followed by the address and sumcheck bytes in the record. The address of the first data byte in the record is expressed by the last four characters of the prefix. Data bytes follow the address; each data byte is represented by two hexadecimal characters. The number of data bytes occurring must be three less than the byte count. The suffix is a 2-character sumcheck, which equals the one's complement of the binary summation of the byte count, address and data bytes.

The end-of-file record consists of the start characters "S9," the byte count (equal to "03"), the address (in hexadecimal) and a sumcheck.



LEGEND

◌ Nonprinting Carriage Return, line feed, and nulls

Motorola Exormax Format, Code 87

Motorola data files may begin with an optional sign-on record, initiated by the start characters "S0." Data records start with an 8- or 10-character prefix and end with a 2-character suffix. The figure shows a series of Motorola Exormax data records.

Each data record begins with the start characters "S1" or "S2"; "S1" if the following address field has four characters, S2 if it has six characters. The third and fourth characters represent the byte count (which expresses the number of data bytes), followed by the address and sumcheck bytes in the record. The address of the first data byte in the record is expressed by the last four characters of the prefix (six characters for addresses above hexadecimal FFFF). Data bytes follow the prefix; each data byte is represented by two hexadecimal characters. The number of data bytes occurring must be three or four less than the byte count. The suffix is a 2-character sumcheck, the one's complement (in binary) of the preceding bytes in the record, including byte count, address and data bytes.

The end-of-file record begins with either an "S8" or "S9" start character. The start character must be "S9" if the previous data record started with an "S1"; otherwise, either "S8" or "S9" may be used. Following the start characters are the byte count (equal to "03"), the address (equal to "0000") and a sumcheck.

LEGEND

◌ Nonprinting Carriage Return, line feed, and nulls

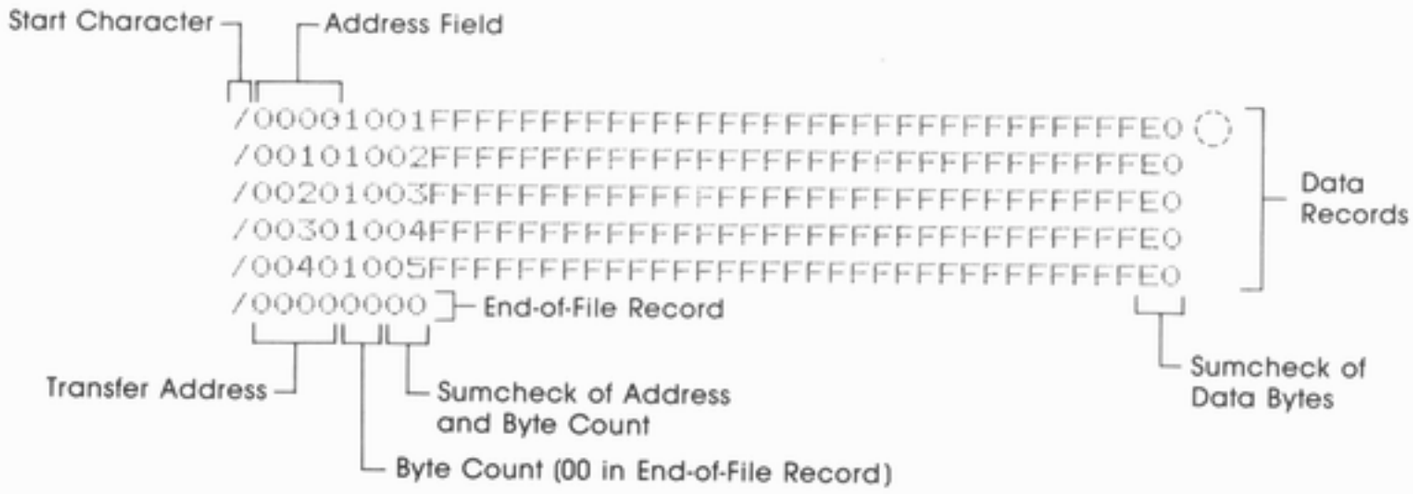
Tektronix Hexadecimal Format, Code 86

The figure illustrates a valid Tektronix data file. The data in each record is sandwiched between the start character (a slash) and a 2-character sumcheck. Following the start character, the next four characters of the prefix express the address of the first data byte. The address is followed by a byte count, which represents the number of data bytes in the record, and by a sumcheck of the address and byte count. Data bytes follow, represented by pairs of hexadecimal characters. Succeeding the data bytes is their sumcheck, an 8-bit sum, modulo 256, of the 4-bit hexadecimal values of the digits making up the data bytes. All records are followed by a carriage return.

Data is output from the programmer starting at the first RAM address and continuing until the number of bytes in the specified block have been transmitted. The programmer divides output data into records prefaced by a start character and an address field for the first byte in the record.

The end-of-file record consists of a start character (slash), followed by the transfer address, the byte count (equal to "00"), and the sumcheck of the transfer address and byte count.

An optional abort record contains two start characters (slashes), followed by an arbitrary string of ASCII characters.



LEGEND

◌ Nonprinting Carriage Return, line feed, and nulls

Error Messages

The following is a list of error codes and their corresponding messages. The circumstances which cause an error message to be displayed are described in the Description column and corrective action to take upon receiving the error message is explained in the right-hand column. The error codes are listed in numerical order.

Code	Name	Description	Corrective Action
15	DRAM ERROR	The self-test shows a dynamic RAM read/write verification error.	Contact your local Data I/O Service Center.
17	BANK ERROR	The bank selector circuit is unable to switch banks.	Contact your local Data I/O Service Center.
18	VCC ERROR	When 5 V is applied to pin 26/28, the voltage drops by an unacceptable level.	Contact your local Data I/O Service Center.
19	VPP ERROR	The programming power supplies cannot be set at the proper levels.	Contact your local Data I/O Service Center.
20	NONBLANK DEVICE	Device failed blank test.	Press ENTER to override.
21	ILLEGAL BIT	Unable to program device due to already programmed bit of incorrect polarity.	Erase the device if possible or discard it.

Error Messages

Code	Name	Description	Corrective Action
22	EEPROM FAILURE	EEPROM device failed to program properly.	Replace device if faulty.
25	BAD DEVICE CODE	An incorrect family and pinout code was entered.	Consult the Device List for the correct family and pinout codes and enter the correct code.
26	WRONG SOCKET	A load operation from a single master was specified, but the device is not in socket 1.	Remove the device and re-install it in socket 1.
27	BLOCK LIMIT ERROR	PROM or RAM size is insufficient to perform the current operation using the begin RAM address, block size, or set size specified.	Check begin RAM address, block size, or set size to be sure they are set correctly. You may need to have your RAM serviced.
29	VERIFY ERROR	The programmed PROM data failed to verify against the master data in RAM.	Reprogram the device, if possible, or, try another device.
31	NOT TRISTATE™	Device failed to TRISTATE™ all data pins.	Indicates that the device is faulty. Replace the device.
32	DEVICE NOT ENABLED	All the device's data pins are not driving when the chip is enabled.	Indicates that the device is faulty. Replace the device.
33	DATA SHORT	The device contains a short on its data line.	Indicates that the device is faulty. Replace the device.

Code	Name	Description	Corrective Action
35	DATA BUS FAIL	Indicates that the part data bus may be damaged or the contact to ZIF socket is faulty.	Replace the faulty device or contact your local Data I/O Service Center.
36	BAD INSERTION	The device was inserted in the socket incorrectly.	Align the bottom-most pin of the device with the bottom of the socket. Pin 1 should be at the top of the device.
38	OVERCURRENT	The device to be programmed is drawing excessive current.	Indicates that the device is faulty. Replace the device.
41	FRAME ERROR	The serial interface detected a start bit but the stop bit was in the wrong position.	Check the current baud rate and stop bit settings.
42	OVERRUN ERROR	The serial interface received characters when the programmer was unable to accept them.	Check the serial port connections. Make sure handshake lines are properly connected and try again.
43	PARITY ERROR	The incoming data has incorrect parity.	Check the parity setting and re-attempt transmission.

Error Messages

Code	Name	Description	Corrective Action
46	I/O TIMEOUT	No characters, or only nulls and rubouts, were received upon serial input for 25 seconds after pressing the ENTER key; or, no characters could be transmitted for a period of 25 seconds due to the state of the handshake lines.	Check all connections and attempt transmission again.
51	I/O FORMAT ERROR	The programmer received an invalid address field.	Check all connections, check the data format and data source and then try again.
52	I/O VERIFY ERROR	The data from the serial port does not match the data in RAM.	Reload data to RAM. If the problem persists, service the programmer or contact your local Data I/O Service Center.
55	I/O FORMAT ERROR	The sumcheck field received by the programmer does not agree with its own calculated sumcheck.	Reload data to RAM. If the problem persists, service the programmer or contact your local Data I/O Service Center.
57	I/O FORMAT ERROR	Data sent to the programmer are non-hexadecimal characters. Input data is in the incorrect format.	Specify the correct data translation format for the data being transferred, or correct the data in the file.

Code	Name	Description	Corrective Action
71	PIN 1 ERROR	The self-test indicates that the socket pin 1driver is at an inaccurate level.	Contact your local Data I/O Service Center.
73	PIN 20 ERROR	The self-test indicates that the socket pin 20 driver is at an inaccurate level.	Contact your local Data I/O Service Center.
74	PIN 21 ERROR	The self-test indicates that the socket pin 21 driver is at an inaccurate level.	Contact your local Data I/O Service Center.
75	PIN 22 ERROR	The self-test indicates that the socket pin 22 driver is at an inaccurate level.	Contact your local Data I/O Service Center.
76	PIN 23 ERROR	The self-test indicates that the socket pin 23 driver is at an inaccurate level.	Contact your local Data I/O Service Center.
77	PIN 26 ERROR	The self-test indicates that the socket pin 26 driver is at an inaccurate level.	Contact your local Data I/O Service Center.
78	PIN 24 ERROR	The self-test indicates that the socket pin 24 driver is at an inaccurate level.	Contact your local Data I/O Service Center.

Error Messages

Code	Name	Description	Corrective Action
79	PIN 28 ERROR	The self-test indicates that the socket pin 28 driver is at an inaccurate level.	Contact your local Data I/O Service Center.
81	NO ID FOUND	The installed device has no electronic ID.	Check the device type. Consult the Device List for the correct family and pinout codes.
82	INVALID ID	The device's electronic ID is incompatible with the family and pinout codes selected.	Consult the Device List for the correct family and pinout codes and make sure the correct codes are entered. Programming with incorrect codes could damage a part.
90	VCC FAIL	The Vcc supply circuits are not working properly.	Contact your local Data I/O Service Center.
91	DAC FAIL	The digital-analog converter circuit failed.	Contact your local Data I/O Service Center.
92	5.6 V FAIL	The 5.6 V power supply circuit failed.	Contact your local Data I/O Service Center.
93	ID VOLT FAIL	The electronic ID power supply circuit failed.	Contact your local Data I/O Service Center.
94	VPP FAIL	The programming pulse power supply circuit failed.	Contact your local Data I/O Service Center.

Index

Index

Index

Index