Kontron

PROM PROGRAMMER
MPP-80S / EPP-80

USER REFERENCE MANUAL 5000-01

Digitized: 08/2016

-No detailed text/spell correction !
-improved Layout
-Original content + index page numbers
 do NOT fit anymore to these digitized version

KONTRON

PROM PROGRAMMER
MPP-80S / EPP-80

USER REFERENCE MANUAL 5000-01

ENGLISH VERSION

PREFACE

This manual describes the operation and use of the EPP-80 and MPP-80S PROM Programmers. It provides "how-to" information for equipment set-up, modification, and expansion, as well as instructions for local and remote programming of Programmable Read-Only Memory (PROM) devices.

The document addresses the operator/programmer whose responsibilities include loading data into PROMs or duplicating PROMs. Because this is typically not a primary responsibility, straightforward but detailed descriptions of user/operator actions are accompanied by diagrams and examples. Following a general description of PROM programmer operation, information presented is functionally organized and related to the programmer keyboard(s).

Separate sections describe use of the built-in Ultraviolet EPROM Eraser, operation of the programmer from a remote processor or terminal, and special functions associated with intelligent personality modules. An extensive index helps to make the manual an effective reference document.

An operator need have only a basic understanding of PROM handling and hexadecimal notation to successfully use this manual to load, inspect, and/or alter data in programmer RAM and then to program any of the supported PROM devices.

REVISION HTSTORY

Title	Number	Date	Notes
MPP-80S / EPP-80 PROM Programmer User Reference Manual	EPP-5000-01	7/83	First Edition Versions: MPP V 2.1 EPP V 2.3 MOD 21 V 3.3 MOD 33 V 2.3 MDM V 2.3 * IFL V 1.2 IFL Boolean V 1.2

* MDM V 2.4 in 4000 PROM

Related Publications

Kontron Module Selecting Guide

Kontron Quick Reference Guide MPP-80S
Kontron Quick Reference Guide EPP-80

CONTENTS

FIGURES AND TABLES

CHAPTER 1

INTRODUCTION

This user manual describes the Kontron EPP-80 and MPP-80S PROM programmers. They are actually independent units whose primary function is the programming of programmable read-only memory (PROM) devices. Either unit is capable of programming common EPROMs, EEPROMs, bipolar PROMs, CMOS PROMs, FPxx (IFL) logic chips, programmable array logic (PAL) chips, and EPROM microprocessors. (In this document all these devices are called "chips".)Programming is accomplished by transferring the contents of PROM programmer RAM into a blank chip attached to the programmer; data communications facilities enable commands and data to be entered at a remote station.

1.1 General

The basic functional components of the PROM programmers include:

a set of keys for operator input (function keys and data keys)

a bank of LED display cells

an ultraviolet (UV) EPROM eraser

an RS-232C/current loop interface port for communication with remote devices

The EPP-80 may also be equipped with one of: a second RS 232C serial interface, IEEE 488 communications interface, or a parallel printer interface.

Programming of a particular chip requires installation of a personality module and socket adapter set. PROM Programmers hahhnt ha operated prnperly without a personaljty modale and socket edepter installed.

A personality module attaches to sockets imbedded in the PROM programmer; each personality module supplies firmware suited to a class of PROM devices. With some personality modules, a socket adapter attaches directly to the personality module; the socket adapter provides an appropriate physical seat and electrical connections for the chip. For other personality modules (e.g., the Universal Gang Module) an "identifier" attaches to the module; this adapts the module for particular chip(s).

You should consult the current Kontron Programmers Module Selecting Guide for detailed information about appropriate personality modules, socket adapters, and identifiers to use for supported devices.

Some personality modules add special-purpose functions that you may activate via the programmer keyboard or remote execution facilities. These "intelligent" personality modules include:

the MOS Device Module, which accomodates most supported 24- and 28-pins MOS devices and provides added functionality;

the Universal Gang Module, for production-style loading of up to eight identical chips simultaneously;

the Integrated Fuse Logic module, for programming FPLA, FPLS, FPGA, and FPRP devices; optional Boolean firmware provides editors, an assembler, and additional features;

PAL	MOD 21 and MOD 33 supply special firmware suited to Programmable Array Logic chips.

The UGM and IFL modules use identifiers instead of socket adapters; identifiers attach to sockets in the personality modules and provide additional chip-specific firmware. For complete descriptions of the specialized personality modules, see Chapter 6.

1.2 PHYSICAL DESCRIPTION

The EPP-80 is a bench-top unit designed for engineering and development use; the MPP-80S is a portable unit built into a sturdy case. The two programmers perform the same basic functions, which are described in this document.

A Quick Reference Guide to PROM programmer use is attached to the front panel of each PROM programmer. This document is tab-indexed by common operations and keys; it includes brief descriptions and examples of commands.

1.2.1 THE MPP-80S

The dimensions of the MPP-80S as a closed unit are as follows:

Width	Height	Depth	Weight
456 mm	130 mm	330 mm	6.5 kg

Most controls and displays are contained on the top panel, and personality modules are mounted on the front panel. The on-off switch, voltage selection switch, and communications port(s) are plainly labeled on the rear panel. The ultraviolet eraser drawer is located on the left-hand side of the top panel, just below a padded chip-holding area.

1.3 DEVICE SET-UP

When you unpack your PROM programmer,- follow the instructions and cautions on the shipping carton. Though the unit has been carefully tested, inspected, and packed to ensure that you receive it in good condition, you should inspect the unit for damage.

Modules and socket adapters are packed in dust-proof, protective plastic and foam padding. Save the packing material and the shipping carton for possible reuse.

1.3.1 POWER ON

Before you apply power to your PROM programmer, check the voltage selection switch located between the power on-off switch and the power cord receptacle. The voltage selection switch must be set for the AC power source you intend to use: 110V or 220V. You should install the selected personality module and socket adapter on the programmer next.

There must be a personality module and socket adapter installed before power is applied!

All controls and displays are contained on the front panel. Personality modules are mounted on the front panel. The onoff switch, voltage selection switch, ultraviolet eraser compartment, and serial communications port are plainly labeled on the front panel as well.

Padded storage compartments in the lid of the MPP-80S carrying case accomodate personality modules, socket adapters, spare chips, and the power cord. A personality module and a socket adapter can be left mounted on the unit when the top is closed.

1.2.2 THE EPP-80

The dimensions of the EPP-80 unit are as foliows:

Width	Height	Depth	Weight
564mm	125 mm	327mm	6.5 kg

Verify that the POWER switch (both MPP-80S and EPP-80) and UV switch (MPP-80S only) are off; these switches are off when a red circle on the front of the switch is showing. Then connect the power cord at the power cord receptacle, and plug it into the AC source.

You can now depress the POWER switch on the PROM programmer to turn it on. Immediately you will notice a line of dashes (MPP-80S) or the message "BUSY" (EPP-80) in the programmer display. Either of these indicates that the PROM programmer is performing a seif-check and initialization.

1.3.2 INITIALIZATION CHECK

Immediately upon receiving power, the PROM programmer executes a program which generates a complex checksum for its operating programs (stored in EPROMs), and compares that checksum with an expected value. If the checksum does not match the expected value, the error message " E CrC " is displayed; the programmer is not usable under these circumstances.

During normal PROM programmer operation, a checksum (CrC) is computed and saved each time data in user RAM memory is changed. When this user RAM data is subsequently accessed (e.g., for programming or check operations) a new CrC value is generated and compared to the old CrC value to verify the memory contents. This checking prevents the transference of bad data to chips via defective memory components.

1.4. GETTING STARTED

If you are unfamiliar with MPP-80S/EPP-80 PROM programmer operation, you should carefully examine Chapter 2 before attempting to use your programmer. For use of the basic PROM programmer, see Chapter 3; study the explanations and examples before attempting operations new to you. Before you use a PROM programmer with one of the special-purpose personality modules, examine the appropriate subsection of Chapter 6.

One of the simplest operations you may be responsible for is making a copy of a chip. For this purpose you will need only the LOAD, PROgram, and CHecK functions, which are described in 3.1.

CHAPTER 2

CONCEPTS

This chapter provides background information related to PROM programmer operation. New users should examine this information before attempting to operate the programmer. Detailed operating procedures are described in following chapters

2.1 Programmer Differences

Although the PROM programmers are nearly identical, there are some noteworthy differences:

The EPP-80 includes a bank of twelve keys (the Special Function keys) that the MPP-80S does not. Some of these correspond to functions that are not available on the MPP-80S.

The EPP-80 display panels present characters in a dotmatrix format which provides much greater fiexibility of form than the seven-segment display panels on the MPP-80S. By comparison, the MPP-80S displays are limited; in particular, some alphabetic characters in MPP-80S displays can only be presented in lower case, or must be approximated (e.g., "1 " for "T"). Most examples in this document present MPP-80S displays; corresponding displays on the EPP-80 are more nearly self-explanatory.

For data communications, the EPP-80S includes an RS-232C (serial) port and an optional second interface which can be a parallel (printer) port, an IEEE-488 port, or another RS-232C port. By contrast, the MPP-80S can include only one communications interface: the RS-232C port.

You should keep these differences in mind as you use this book.

2.2 STANDBY MODE

When the PROM programmer is inactive and ready to accept a command sequence, it is said to be in standby mode. This is indicated in the EPP-80 display by

```
   READY FOR     xxxx
```

and in the MPP-80S display by

```
  Fc             xxxx
```

where "xxxx" is the device type number for which the programmer is prepared (i.e., equipped). The PROM programmer determines the type number automatically, from the personality module/socket adapter set that is installed.

2.3 OPERATIONAL SEOUENCES

To perform a particular task with the programmer, you press a series of keys. The first key typically identifies the function to be performed (e.g., LOAD). Then you may indicate one or more parameters. You press "ENTER" to indicate that you are ready for the next step; the display indicates what you should do next.

The keys "ENTER" and minus (" - ") can serve as go and stop indicators, respectively. You frequently press "ENTER" to proceed to the next step in an operation; - " can cancel some operations and return the programmer to standby mode. For some operations, "ENTER" increments a counter (e.g. for the next RAM address) and " - " decrements the counter.

PROM programmer displays include abbreviated indications of the current operation at the left (e.g., "Pr" for a Program operation). Look for these in examples presented in later chapters.

Many displays include a field (e.g., an address) which you can change; with the EPP-80 / MPP-80 S and MDM personality module only, the current contents of this variable field can be set to flash on and off. You may accept the current contents (i.e., the default value) of the field by pressing "ENTER", or you may enter a new value for the field on the data keypad (the four-by-four set of data keys "0" through "F" to the right of the "ENTER" key).

2.4 Operating The Mpp-80s Ultra-Violet Eraser

The ultraviolet (UV) eraser is located at the left end of the programmer front panel. This section describes manual operation of the MPP-80S eraser. Operation of the eraser on the EPP-80 is controlled via built-in software and the "UV" Special Function key, as described in 3.4.12.

The UV eraser accepts as many as five 24-pin EPROMs or one 40-pin LSI device with on-board EPROM. Follow these steps to erase EPROMs:

1. Insert the EPROMs into the eraser cavity window-side down and close the cover.

2. Switch programmer power on (if necessary) and then set the UV switch to ON. The lamp in the switch should light within 5 seconds.

3. See manufacturer's specifications to determine the proper length of time to erase your EPROMs; the intensity of the UV lamp is 5mW/cmE2 when it is new.

The eraser lamp automatically shuts off when the eraser cover is opened. Note that you need to.manually set the switch to OFF when the Operation has been completed.

To replace the UV eraser lamp, remove the front panel of the PROM programmer. You can then unscrew the lamp from the holder and insert a new lamp.

CHAPTER 3

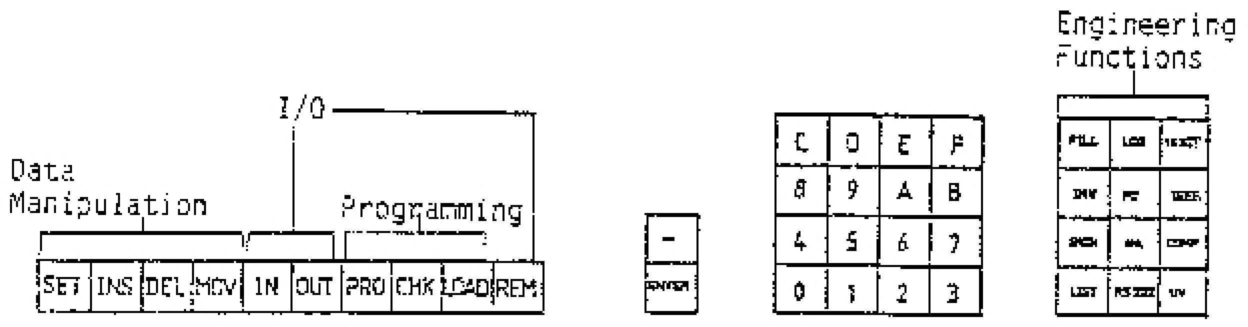
LOCAL OPERATION

This chapter includes specific operating instructions for the EPP-80 and MPP-80S PROM programmers. Local Operation of these units is controlled from the built-in keyboard; the programmer keyboards are identical, except that the EPP-80 includes 12 additional special-function keys. Remote operation of the programmers is described in Chapter 4; however, functional descriptions presented in this chapter apply to remote operation as well.

Figure 3-1 depicts the EPP-80 keyboard and functional groups of keys as they are discussed in this chapter; the MPP-80S keyboard is similar, but without a bank of Special Function keys. Keys are color-coded by function for ease of use, as follows:

Functional	COLOUR	
Universal Programming-- PROgram, CHecK, LOAD	Red	3.1
Data Manipulation-- SET, INSert, DELete, MOVe	Yellow	3.2
Input/Output-- INput, OUTput, REMote	White	3.3
Engineering Functions-- FILL, LOG, 16 BIT, INVert, etc.	Gray	3.4

Figure 3-1. EPP-80 Keyboard



3.1 UNIVERSAL PROGRAMMING FUNCTIONS

The programming functions LOAD, CHecK, and PROgram represent the primary tasks of the PROM programmer: copying data (i.e., coded programs) into PROM programmer Random-Access Memory (RAM), transferring data from programmer RAM into a chip, and verifying contents of RAM and the PROM. You do not need other functions to copy data from one chip to another.

3.1.1 LOAD

When you press the LOAD key, you initiate a sequence of steps to transfer the contents of all or part of the chip currently in the socket into programmer RAM. You are given the opportunity to select a RAM starting address (SA), or you can accept the default starting address (0000) that is displayed. You may also select the ending address (EA) in RAM, or accept the displayed default -- which corresponds to the entire chip being copied.

Data from 4-bit chips can be copied into low or high halfbytes (nibbles) of PROM programmer RAM, as indicated in Example 2 following.

LOAD Example 1:
Copy contents of a pair of 2716 chips into programmer RAM.

Press	The display is	and it means
	Fc 2716	Programmer is in standby mode; it is equipped with personality module MOD 1 and socket adapter SA 1-1 for the 2716 EPROM device.
LOAD	Ld SA *0000*	Default starting address in RAM is *0000;* a different hex address can be entered via the data keyboard
ENTER	Ld EA 07FF	Default ending address; this depends upon the entered starting address and it accomodates the entire chip. A different hex address can be entered via the data keyboard.
ENTER	Ld P-I	PROM to be copied from should be mounted in the socket at this time, if it is not already mounted.
ENTER	--------------	Loading Operation in progress.
	Fc 2716	Loading successfully completed; programmer is in standby mode.

LOAD Example 1 (continued):

Following paragraphs indicate how to continue with Example 1 to copy the contents of a second 2716 chip into programmer RAM, immediately following the data from tue first 2716 chip
(already loaded). Data from the two chips can then be
programmed into a single 2732 chip, using a different personality module.

With the second 2716 chip, repeat the sequence of steps previously listed, but specify *0800* as the Start Address before you press ENTER (second step). The programmer automatically uses *OFFF* as the End Address (corresponding to
the third step). When you press "ENTER" following the display:

```
             Ld P-1
```

data from the second 2716 chip is loaded into RAM locations *0800-OFFF;* previously loaded data in RAM locations *0000-07FF*
is unaffected. Data from the two EPROMS has been
concatenated into a single "file" in RAM locations *0000- OFFF.*

To program the RAM data into a 2732 chip, you should wait until the programmer returns to standby mode (following the second load operation) and then remove the MOD 1/SA 1-1 assembly; replace these with MOD 9/SA 12. For details about
the programming operation, see 3.1.3. In order for the
complex copy Operation to be successful, the following conditions must be met:

1. The programmer <u>must</u> be in standby mode while the personality modules are exchanged.

2. Programmer power must not be interrupted.

3. No keys should be pressed while there is no personality module installed in the programmer.

4. You should press "ENTER" once immediately after the new module is installed; the new module type appears in the standby display, indicating that the new personality module is operational.

5. This procedure is not valid with the MDM or IFL special personality modules.

LOAD Example 2:

Load two 3605 PROMs into low and high half-bytes of programmer RAM, for later programming of a single 8-bit chip.

Press	The display is	and it means
	Fc 36--	Programmer is in standby mode and equipped with personality module MOD 17 and socket adapter SA 24 (for 3605 PROM).
LOAD	Ld SA 0000	Load operation is indicated; default Starting Address (SA) is 0000.
ENTER	Ld EA 03FF	Default ending address corresponds to all of the chip.
ENTER	Ld Lo 03FF	Indicates that data is to be loaded into the lower 4-bit nibble of each byte in programmer RAM. See note (*).
ENTER *	Ld P-1	PROM to be copied from should be mounted in the socket at this time (if it is not already mounted).
ENTER	--------------	Loading in progress. Data is loaded into the low half-byte of addresses *0000* through 03FF.
	Fc 36--	Loading Operation is complete, programmer is in standby mode; PROM may be removed from socket. See the following note (*) to continue with second PROM.

* for the second 3605 PROM, repeat this entire sequence of steps, except press the " - " key at this step to load data into the high nibble of addresses *0000* through 03FF.

3.1.2 CHECK

Check operations verify contents of programmer RAM and/or PROM devices; data is not altered during a check. You can initiate the following check operations:

CHECK *0*	Blank Check -- verifies that a PROM in the socket adapter is in the proper blank state: all bits are zero or all bits are one, depending upon the chip. You do not need to know which is the appropriate state, personality module firmware handles it.
CHECK 1	Verification Check -- compares the contents of programmer RAM with the PROM and reports differences in the programmer display.
CHECK 2	Illegal Bit Check -- verifies that the PROM in the socket adapter (which may be partially programmed) can accept the data contents of RAM, and reports discrepancies via programmer display. Because the programmer can only set or reset PROM bits (not both for the same chip, EEPROMs excluded), some partially programmed PROMs cannot be loaded with certain bit patterns.
CHECK 3	PROM Checksum calculates a unique hex value that represents the contents of the PROM in the socket, and displays that value. See Notes below.
CHECK 4	RAM Checksum calculates a unique hex value representing the contents of programmer RAM (using the same algorithm as that for Check 3) and displays the value. See Notes below.

Additional Check operations are possible with the MDM personality module; these operations are described in Chapter 6.

Notes regarding CHecK operations:

1. Two different checksum algorithms are available; they are described in detail in Appendix E.

2. CHECKS 3 and 4 are not available with the IFL personality module, which is described in 6.5.

To execute a CHecK operation, press the "CHecK" key. The display shows the default check number *0* (for blank check); enter another number an the data keypad to change to another check operation. Press "ENTER" to register the displayed CHecK number.

Start Address and End Address prompts are presented, similar to those in the LOAD examples (3.1.1). You may respond to these by pressing only "ENTER", if the displayed default address limits are appropriate; alternatively, you may enter different address(es) from the data keypad, each followed by "ENTER".

The programmer displays

```
          CH P-1
```

to indicate that the CHecK Operation is ready to proceed. When you next press "ENTER" the operation actually begins; the display is also a reminder to insert the PROM in the socket, if you have not already done so. Press the " - " key instead of a number or ENTER to abort the check operation and return to standby mode.

Errors encountered during CHecK operations are reported in the programmer display panel. For example, the following display indicates that PROM contents and RAM data do not match at address *0101:*

```
 E   C H 1   0 1 0 1    F F     62
         _________________________________
 !    !         !        !      !
 !    !         !        !      !
 !    !         !        !     RAM contains hex value 62.
 !    !         !        !
 !    !         !       PROM contains hex value FF.
 !    !         !
 !    !        Hex address for which comparison fails.
 !    !
 !    Check 1 has failed.
 !
 Error Indicator.
```

3.1.3 PROGRAM

When you press the PROgram key, you initiate an Operation which transfers the contents of all or part of programmer RAM into the chip mounted an the socket adapter. You may accept the displayed Starting Address default (0000) or enter a different starting address; likewise, you may accept the Ending Address default (which corresponds to the Starting Address plus the size of the chip) or enter a different Ending Address.

When you press ENTER after establishing the ending address, transference of data actually begins. This is accomplished in three steps, as follows:

1. The PROM is automatically checked for programmability (Check 2, as described in 3.1.2).

2. Data is transferred into the PROM.

3. The new contents of the PROM are compared to programmer RAM (Check 1).

PRO Example:
Program part of a 7620 PROM (512 x 4 bits).

	Fc 76--	Programmer is in standby mode.
PROgram	Pr SA *0000*	Default Start Address is displayed.
1 *0* 0	Pr SA 0100	New Start Address selected.
ENTER	Pr EA 02FF	Default Ending Address displayed; this accomodates the entire chip from the entered Start Address.
0 1 F F	Pr EA 01FF	New Ending Address selected.
ENTER	Pr Lo 01FF	Program from lower nibble of RAM range (this is the default).
ENTER Or -	Pr P-I Pr P-I	Accepts default (i.e., program from lower nibble); alternatively, Program from upper nibble of RAM range. Following either of these, insert the PROM in the programming socket.
ENTER		Programming in process.
	Pr PI-	Programming complete; remove PROM.
ENTER	Fc 76--	Programmer returned to standby.

3.2 DATA MANIPULATION

You can inspect and manipulate PROM programmer RAM with the yellow keys SET, INSert, DELete, and MOVe. Memory addresses and data are expressed using hexadecimal digits.

Note that SET, INSert, DELete, and MOVe (as well as FILL and INVert) are not applicable to the data storage format employed by the IFL personality module. The IFL module utilizes the named keys for entirely different functions; see 6.5 for complete descriptions of the IFL functions.

3.2.1 SET

When you press the SET key, the programmer display presents the 4-digit hex address *0000* in the Address field. You can view the value stored at that address by pressing the ENTER key; or you can enter a different address an the data keypad.

When you have selected the desired starting address, press ENTER to view the data stored at that RAM address; the value is displayed in the RAM field. To change the value from that displayed, press appropriate data keys; the new values are displayed in the PROM Input field. The new value replaces that in the RAM field after you press ENTER again, and the next memory location is displayed.

When you are using the SET function, the ENTER key advances to the next sequential address, and the " - " key returns to the preceding address. You can step through memory to examine its contents by repeatedly pressing ENTER. To view the contents of addresses numerically distant from that currently displayed, you might find it expedient to press the SET key and begin the Operation again, with a new starting address.

SET Example:
Inspect the contents of RAM addresses *0000* through *0002.* Modify them, if necessary, to the values 02, 45, and 56 respectively.

	Fc 2708	Programmer is in standby mode, and equipped with personality module MOD 2 and socket adapter SA 2-1 (for 2708 EPROM).
SET	SE SA *0000*	Set operation is in progress,default starting address is hex *0000*
ENTER	SE SA *0000* FF	Value currently stored at address *0000* is hex FF.

SET Example: (continued)

0 2	SE SA *0000 02* FF	Potential new value for this address is hex 02.
ENTER	SE SA *0001* 45	New value for address *0000* is accepted; the current value for address *0001* is hex 45.
	SE SA *0000 02*	Going back to address *0000*,the current value stored there is 02.
ENTER	SE SA *0001* 45	Current value at address *0001* is 45.
ENTER	SE SA *0002* 56	Current value at address *0002* is 56.

Any other function key terminates the SET function and activates the new selected function.

(At this point, the REMote key can be pressed twice in succession to effect an escape from the SET operation, as follows:)

REM	---------------	Programmer is changing to remote operating mode; all keys except REMote are locked.
	rE 2708	Programmer is in remote operating mode.
REM	Fc 2708	Programmer is returned to local operating mode, in a standby (ready) condition.

3.2.2 INSERT

You can insert a value at a specified memory location after pressing the INSert key. The value previously stored at that location is moved to the next sequential address, as are all values stored at higher addresses. The value stored in the highest RAM address (e.g., address 1FFF) is lost.

After you press the INSert key, the default Starting Address *(0000)* is displayed. You can enter a different Starting Address an the data keypad; proceed with the insert operation by pressing ENTER.

Current value for the displayed address is presented when ENTER is pressed. You enter the value to be inserted an the data keypad, and it is simultaneously displayed in the PROM Input field. Press ENTER to accept the new value into memory; the next address and its current value are displayed. Repeat the process to insert additional values at this location. Pressing ENTER without a new data value advances the display to the next sequential memory location and inserts the last value again.

To inspect inserted data, press the " - " key to step backwards through memory from the end of the Insertion to the original Starting Address. Press the REMote key twice to end the Insert Operation and return the programmer to standby mode.

INS Example:
Insert the values 65 and 20 into programmer RAM, beginning with location *07F0.*

	Fc 2716	Programmer is in standby mode, equipped for 2716 PROM.
INSert	In SA *0000*	Ready for the insert operation; default Starting Address is *0000.*
7 F *0*	In SA *07F0*	Starting Address is now *07F0.*
ENTER	In SA *07F0* 8F	Current value at location 07F0 is 8F.
6 5	In SA *07F0* 65 8F	New value to be inserted is 65.
ENTER	In SA 07F1 8F	New value has been inserted. Current value at the next memory location (07F1) is 8F.
2 *0*	In SA 07F1 20 8F	New value to be inserted is 20.
ENTER	In SA 07F2 8F	New value has been inserted. Current value at the next memory location (07F2) is 8F.
	In SA 07F1 20	Current value for location 07F1 is 20; location 07F2 still contains the data value 8F because the " - " key does not cause data to be inserted.
	In SA *07F0* 65	Current value for location *07F0* is 65--just checking!

3.2.3 DELETE

A delete operation, initiated when you press the DELete key, deletes data from programmer RAM. After you specify the Start Address, one data byte is deleted each time you press "ENTER". Remaining data is relocated to lower memory addresses, and the last locations in memory are filled with blanks (either *00* or FF, depending upon the chip for which the programmer is equipped).

DEL Example:
Delete the contents of memory locations *0110* through 011A; this is actually 11 data values.

	Fc 2532	Programmer is in standby mode, equipped for 2532 EPROM (i.e., equipped with personality module MOD 7 and socket adapter SA 1).
DELete	dE SA *0000*	Delete Operation in progress; default Starting Address is *0000.*
1 1 0	dE SA *0110*	Starting Address is changed to *0110.*
ENTER	dE SA *0110 0F*	Current value at location 0110 is 0F.
ENTER	dE SA *0110* 56	The value *OF* has been deleted. Current value at location *0110* is 56; this is the value previously stored in location 0111.
ENTER	dE SA *0110* 32	The value 56 has been deleted. Current value at location *0110* is 32; this is the value previously stored in location 0112, then moved to 0111 after the first deletion.
ENTER	dE SA *0110* xx	The current value is deleted each time you press ENTER and the next value (xx) appears in the RAM field.
	(8 more times)	
REMote	rE 2532	Programmer is resetting for remote Operation. Programmer is in remote mode; all keys other than REMote are locked.
REMote	Fc 2532	Programmer is reset to local mode, in a standby (ready) condition.

3.2.4 MOVE

You can copy one or more data values within programmer RAM with a Move Operation. After you press the MOVe key, you establish a destination address (dE) for the first byte to be moved, and specify the Source address (So) of the data to be moved, and then enter a byte Count (bC) that represents the number of data values to be moved.

Moved data is inserted at the indicated destination address, overwriting data that is currently in memory. Locations that formerly contained the moved data are not changed; the Move operation might rightly be called a Copy Operation.

NOTE:
if a block of data is to be moved upwards in memory (i.e., to a higher RAM address range), the address ranges of the source and destination blocks must not overlap; if they do overlap, some of the source block will be overwritten before it can be copied. To accomplish such a move, first move the source block to an intermediate destination block that is completely above the source, then use another MOVE Operation to go from this intermediate block to the intended (original) destination.

MOV Example:

Copy the 8 bytes beginning at location 0200 to addresses 0B00 through 0807

	Fc 2716	Programmer is in ready condition and equipped for 2716 EPROM (i.e., with personality module MOD 1 and socket adapter SA 1-1).
MOVe	Mo dE *0000*	Move Operation is in progress; default Destination address is *0000.*
B00	Mo dE *0b00*	Destination address is now *0B00.*
ENTER	Mo So *0000*	Default Source address is*0000.*
2 *0 0*	Mo So *0200*	Source address is now *0200.*
ENTER	Mo bC *0000*	Byte count (i.e., number of data bytes to move) is zero by default. If you press ENTER now, the Operation will terminate without any data being copied.
8	Mo bC *0008*	Number of bytes to move is 8.
ENTER	-----------	Move in process;
	Fc 2716	move completed and programmer in standby condition.

3.3 INPUT/OUTPUT FUNCTIONS

You transfer data to or from the PROM programmer via the INput and OUTput keys. You can transfer control of programmer operations to or from a remote device with the REMote key.

3.3.1 INPUT

When you press the INput key, you inititiate a series of steps that leads to reception of data by the PROM programmer over the built-in serial interface. This data may be sent from a remote computer or terminal, using any of a wide variety of transmission protocols.

After pressing INput, you must enter a 2-digit transmission format code number; supported formats are listed in Table 3-1 and described in Appendix B. Then you typically enter a Starting Address (i.e., the location in RAM where the first data byte is to be stored) and an Ending Address (for the last data byte). Some transmission protocols include a Starting Address in the transmitted data stream; for some of these an address offset is requested instead of (or in addition to) a Starting Address. The offset is added to the transmitted Starting Address to determine the RAM location for received data.

Table 3-1. Communication Formats

Code	Name	Code	Name
10	BPNF	41	MOTOROLA EXORCISOR
11	BHLF	42	RCA COSMAC
12	B1OF	43	F-8 FORMULATOR
20	ASCII HEX SPACE	44	SIGNETICS ABSOLUTE
21	ASCII HEX PERCENT	45	TEKTRONIX HEXADECIMAL
22	ASCII HEX APOSTROPHE	46	MOS TECHNOLOGY
23	ASCII HEX COMMA	47	LIST 8 BYTES/LINE **
30	BINARY	48	LIST 16 BYTES/LINE **
31	DEC BIANRY	60	TRANSKON W/ACK
32	INTEL NON INTELLEC	70	12-BIT BINARY
33	7-BIT ASCII W/ECHO	71	12-BIT BINARY DUMP**
34	7-BIT ASCII W/OUT ECHO	80	INTEL 8086 HEX*
35	7-BIT ASCII LOADED TO DISPLAY (EPP-80 ONLY)		
40	INTEL HEX		

* for Input only   ** for Output only

IN Example:

Receive data in ASCII Hex format with commas as delimiters between bytes (protocol code 23), and subtract hex 100 from the transmitted address before storing that data in RAM.

Press	Diplay is...	And it means
	Fc 2758	Programmer is in ready condition and equipped for 2758 EPROM (i.e., with personality module MOD 1 and socket adapter SA 1-2).
INput	In 00	Input command accepted; input code is set to 00 (not a valid protocol).
2 3	In 23	Input code now set to 23 (ASCII Hex with comma separators).
ENTER	OFFSE 0000	Because Starting Address is transmitted along with data in protocol 23, an address offset is requested.
1 0 0	OFFSE 0100	Address offset is hex 100.
-	In Po	The offset is designated negative with the " " key; hex 100 is automatically subtracted from the transmitted Starting Address to determine actual location for incoming data in memory. The display indicates that this is your opportunity to select the Input Polarity.
ENTER	--------------	The ENTER key designates positive polarity (i.e., received data is not inverted); the " - key would result in the inversion of received data before it is stored in RAM. Dashes (---) indicate that the PROM programmer is expecting data input through the serial port. The dashes remain until an End of Transmission message appropriate to the transmission protocol is detected.
	Fc 2758	Data received without error; programmer is in standby mode.

3.3.2 OUTPUT

When you press the OUTput key, you intitiate a series of steps that leads to the sending of data from the PROM programmer over the built-in serial interface. This data may be sent to a remote computer, terminal, printer, or other device using any of a wide variety of transmission protocols.

After pressing OUTput, you must enter a 2-digit protocol code number; supported protocol codes are listed in Table 3-1 and described in Appendix B. Then you typically enter a Starting Address (i.e., the location in RAM where the first data byte to be transmitted is stored) and an Ending Address (for the last data byte). Some transmission protocols utilize an address offset in addition to the Ending Address.

OUT Example:
Transmit the contents of RAM locations *0000* through *07FF* to a printer so that data is listed 8 bytes per line (protocol code 47).

Press	Diplay is...	And it means
	Fc 2716	Programmer is in ready condition and equipped for 2716 EPROM (i.e., with personality module MOD 1 and socket adapter SA 1-1).
Output	OU 00	Output command accepted; transmission protocol code is set to 00 (not a valid protocol).
47	OU 47	Protocol code is now 47: List Mode, 8 bytes per line.
Enter	OU SA 0000 47	Default Starting Address is *0000.*
Enter	OU EA 07FF 47	Default Starting Address is accepted with ENTER (a different address could have been keyed in). The default Ending Address is displayed.
ENTER	OFFSE 0000	Ending Address default is also accepted (with ENTER). The offset is currently *0000.*
ENTER	-------------	Default offset value *(0000)* is accepted; data transfer in progress.
	Fc 2716	Programmer is in standby mode.

3.3.3 REMOTE KEY

When you press the REMote key, control of PROM programmer functions is transferred to the device connected at the serial communications port. The keyboard is effectively "locked" except for the REMote key itself. To return to local control of the programmer, press the REMote key again. Remote operation of the PROM programmers is described in Chapter 4.

The REMote key can also be used for "function escape" in local mode: to accomplish this, press the REMote key twice. The programmer responds by exiting from the current Operation and returning to local standby mode. There should not be a device connected to the communications port for this operation. (In fact, the programmer transmits characters -- a remote standby message -- over the serial port when the REMote key is initially pressed. If there is a device connected to the serial port and holding the ClearTo-Send (CTS) line low, the programmer hangs up until the device is disconnected.)

An alternative key sequence to reset the programmer at any time is the "CHecK" key immediately followed by the " - key. This sequence intitiates a CHECK operation and immediately cancels it, causing the programmer to return to local standby mode. This key sequence cannot cause the programmer to hang up, and ensures that RAM data will not be altered.

REM Example:

Reset the PROM programmer from a SET operation.

Press	Diplay is...	And it means
	SE SA 0015 9C	The current value stored at RAM address 15 is 9C.
REMote	----------------	Programmer is changing to remote operating mode.
	rE 2716	In remote mode.
REMote	Fc 2716	Programmer is in (local) standby mode.

3.4 ENGINEERING FUNCTIONS

Special functions associated with firmware engineering are conveniently executed with a single keystroke an the EPP-80 PROM programmer Special Function keypad. Some of there functions are also available an the MPP-80S, accessible via the data keypad when the programmer is in standby status. Special functions common to both programmers and those particular to the EPP-80 are described in following subsections.

Special Functions available an both the EPP-80 and MPP-80S PROM Programmers include:

Function	See Subsection	EPP-80 Key	MPP-80S Key
Fill a range of RAM locations with a value	3.4.1	FILL	F
Complement values in a range of RAM locations	3.4.2	INVert	E
Set or verify baud rate for serial interface	3.4.3	RS 232	B
Jump to a RAM address and execute Z80 instructions there	3.4.4	C	C

The following special functions are available an the EPP-80 (except with the IFL module as noted below), but not an the basic MPP-80S:

Function	See Subsection	EPP-80 Key
Perform logical operations (AND, OR, EXOR) an a range of RAM values	3.4.5	LOGic
Split 16-bit data into 8-bit fields, or "shuffle" 8-bit data into 16-bit words	3.4.6	16 BIT
Control operating conditions of the programmer itself	3.4.7	FC
Enable control of the EPP-80 via the optional IEEE-488 (GPIB) communications interface	3.4.8	IEEE
Search a range of RAM locations for a specified value string	3.4.9	SeaRCH
Begin or end compressed operating mode	3.4.10	COMPress
Display RAM contents in binary, ASCII, octal, or decimal notation	3.4.11	LIST
Set or inspect time for ultraviolet EPROM eraser	3.4.12	UV
Invoke special functions (for certain personality modules only)	3.4.13	*A

Many of these functions can also be performed an an MPP-80S PROM programmer equipped with the MDM intelligent personality module. See 6.3 for specific details.

Note that special function keys listed above are disabled when the IFL personality module is installed (because the functions do not apply). With this module only, the band rate for the EPP-80 serial port can be set using the "B" data key (as an the MPP-80S). Programmer Operation using the IFL module is described in 6.6.

3.4.1 THE FILL FUNCTION

When you press the FILL key an the EPP-80, or the F key an the data keypad when the MPP-80S is in standby mode, you initiate a sequence of steps to fill all or part of RAM with
a constant value. The programmer display immediately shows
a "set" value appropriate to the PROM device for which the PROM programmer is currently equipped; this is either hex *00* or hex FF. You can enter a different fill value at the data keypad, or accept the displayed default.

When you press ENTER to signify an acceptance of the fill value currently displayed, the default Starting Address
(0000) for the fill Operation is displayed. Again, you may
enter a different Starting Address. When you next press
ENTER, the Ending Address default is displayed; this address includes enough RAM to fill the chip for which the PROM programmer is currently equipped. You may accept the default Ending Address by pressing only the ENTER key, or you may enter a different ending address an the data keypad and then press ENTER.

The fill value you specify is placed into designated RAM memory locations, and the programmer returns to standby mode.

Note that the FILL function is not available if the IFL personality module (6.5) is installed.

FILL Example:

Place the hex value OC into RAM locations 0700 through 07FF.

Press	Diplay is...	And it means
	Fc 2716	Programmer is in standby mode, equipped for 2716 EPROM.
FILL	FI	*00* Fill Operation initiated; default fill value is *00.*
C	FI OC	Fill value is changed to OC.
ENTER	FI SA 0000 *OC*	Fill value accepted; default Starting Address (SA) is *0000.*
7 0 0	FI SA 0700 *OC*	New Starting Address is *0700.*
ENTER	FI EA OEFF OC	Starting Address accepted; default Ending Address (EA) is *OEFF,* which corresponds to SA + entire 2716 chip.
0 7 F F	FI EA 07FF *OC*	New Ending Address is 07FF.
ENTER	----------------	RAM contents are being modified.
	Fc 2716	Operation complete; programmer in standby mode.

3.4.2 THE INVERT FUNCTION

When you press the INVert key on the EPP-80, or the E key on the data keypad when the MPP-80S is in standby mode, you make it possible to logically invert (i.e., complement) data in all of RAM. Data is actually inverted when you press the " - " key after an initial display.

To abort an invert Operation after the INVert or E key has been pressed, press the ENTER key. Note that for the invert operation, " - " activates the function and ENTER cancels it.

INV Example:

Invert Data in programmer RAM.

Press	Diplay is...	And it means
	Fc 2708	Programmer is in standby mode, equipped for 2708 PROM.
INVert	ME Po	Ready for invert Operation.
	Fc 2708	Data in RAM is being changed. Operation complete, programmer in standby mode once again.

3.4.3 THE BAUD FUNCTION

When you press the RS232 key an the EPP-80, or the B key an the data keypad when the MPP-80S is in standby mode, the current baud rate setting for data transmission via the RS 232 C interface is displayed and you can change it. If the EPP-80 is equipped with two RS 232 transmission ports, you must first select port A or port B for consideration.

Programmer baud rate is initialized to a default value according to the setting of DIP switches; these switches are an the EPP-80 back panel or internal to the MPP-80 chassis. (See Chapter 5 for instructions an resetting the default baud rate). Switches are factory-set to 2400 baud. The Baud Function can be used to establish any of the following data transmission rates:

50	200	4800
75	300	9600
110	600	19200
134	1200	38400
150	2400	

After the current baud setting is displayed, you can press ENTER to display the next higher possible baud rate, or " - " to display the next lower rate. A change to the
displayed baud rate is effected when you press another function key or reset the processor to standby mode (by pressing the REMote key twice).

RS232 Example:
Examine the current baud rate setting, and change to *300* baud.

Press	Diplay is...	And it means
	FC 2716	Programmer is in standby mode, equipped for 2716 PROM.
B	bAUd 2400	Data transmission rate is currently set to 2400 baud.
ENTER	bAUd 4800	The next higher baud rate; we need a lower one!
-	bAUd 2400	The next lower rate
-	bAUd 1200	the next --
-	bAUd 0600	--
-	bAUd 0300	this is the desired rate.
REMote	rE 2716	Programmer reset to remote operating mode, with no connected remote device.
REMote	Fc 2716	Programmer reset to local standby mode.

3.4.4 THE JUMP FUNCTION

When you press the C key an the data keypad when the EPP-80 or MPP-80S is in standby mode, you can specify that the Z80 processor in the programmer takes its next instructions from locations in user RAM. Z80 machine-code instructions can be loaded into consecutive RAM locations from the data keypad,
or via the communications interfaces. The default RAM address to jump to is absolute location *8000,* which corresponds to user RAM address *0000;* you may specify another address after pressing the JUMP key.

The default instruction sequence stored in firmware at absolute address *0000* is a power-on initialization routine. You can store an assembled stream of Z80 instructions at any user RAM address, and "jump" to that address, causing the programmer to perform functions that are not built-in to system firmware; for example, the programmer can execute module calibration routines.

JUMP Example:
Execute the power-on initialization sequence from the keyboard.

Press	Diplay is...	And it means
	READY FOR 2716	Programmer is in standby mode; this display appears an the EPP-80.
C	JUMP 8000	"Jump to" address default is 8000 (i.e., user RAM location 0000). You can enter a new address at the data keypad followed by ENTER, or press ENTER now to accept the default.
0	JUMP 0000	New "jump to" address is 0000; this is where the power-on initialization routines reside.
ENTER	BUSY	Programmer processor is executing instructions that start at absolute location 0000; this is the power-on initialization sequence.
	READY FOR 2716	Programmer is again in standy mode; note that RAM values and programmer options are reset, just as if the programmer had been powered down and back up.

3.4.5 LOGICAL OPERATOR FUNCTIONS

You press the LOGic key on the EPP-80 to perform the logical operation AND, OR, or EXOR (EXclusive OR) on all or part of the data in RAM. First you select the operation with "A", *"0"* (zero), or "E"; then you enter a 2-byte data value (to be combined with current RAM data), followed by a Starting Address and Ending Address to indicate which RAM data is to be processed. The data value you enter is the first operand for the designated logical Operator; each RAM data value in the prescribed range becomes the second operand in turn. The result of the logical Operation is stored into the RAM address from which the second operand is taken.

For example, if you specify a logical AND of the data value 7F with all of RAM, each data value in RAM will have its most-significant bit reset to *0,* and all other data bits will remain unchanged.

LOG Example:

Complement the two least-significant bits of RAM data at addresses 0700 through 07FF.

Press	Diplay is...	And it means
	FC 2716	Programmer is in standby mode, equipped for the 2716 EPROM.
LOGic	AND OR EXOR	Select logic operation; A, 0, and E may be flashing in the display.
E	EX 00	EXclusive OR chosen; ready for data.
3	EX 03	Operand to be used is hex 03.
ENTER	EX SA 0000 03	Operand accepted; Starting Address default is 0000.
7 0 0	EX SA 0700 03	New Starting Address is 0700.
ENTER	EX EA OEFF 03	Starting Address accepted; default Ending Address is OEFF.
0 7 F F	EX EA 07FF 03	New Ending Address is 07FF.
ENTER	-----------------	Operation in progress.
	Fc 2716	Operation complete; standby mode.

3.4.6 SPLIT/SHUFFLE FUNCTION

You press the 16 BIT key an the EPP-80 Special Function keypad to utilize the Split/Shuffle Function. The first time the key is pressed, a data Split is enabled; after a Split function is executed, pressing the key enables the Shuffle function. The split or shuffle is actually performed when you press ENTER. The discussion immediately following applies to the EPP-80 only; for additional functionality associated with Split/Shuffle operations using the MDM personality module (6.3), see indented paragraphs at the end of this subsection.

The Split function may be used to program PROMs for a 16-bit CPU. A 16-bit memory can be implemented with two 8-bit wide parallel PROMS: one PROM contains the low-order byte of each word, and the other PROM contains the high-order byte at a corresponding address.

To facilitate programming of such 8-bit PROMs, the EPP-80 Split Function actually "splits" 16-bit data in RAM (one word of which occupies 2 consecutive RAM locations) into two memory segments, or blocks, each containing enough data to fill the 8-bit PROM for which the programmer is equipped. Contents of even addresses (i.e., low-order bytes) are transferred to the lower block of RAM, and contents of odd addresses (i.e., high-order bytes) to the higher block. The blocks can then be loaded into 8-bit PROMs separately, and the requisite parallel PROMs are programmed.

For example, if the Split Function is performed while the PROM programmer is equipped for a 2716 EPROM (8 bits wide X 2K, address range hex 000 through 7FF), the Split utilizes RAM addresses 000 through FFF. Data bytes from even addresses (0, 2, . . . , FFC, FFE) are assembled into a block at RAM addresses 000 through 7FF; data bytes from odd addresses (1, 3, . . . , FFD, FFF) are assembled into a block at RAM addresses 800 through FFF. To program a 2716 EPROM with low-order data, use the PROgram function with Starting Address 0000 and Ending Address 07FF; to program another 2716 EPROM with high-order data, use Starting Address 0800 and Ending Address OFFF.

You can also implement 16-bit wide memory with four 4-bit wide PROMs, if the programmer personality module accomodates them. After the Split Function is performed, you can program each pair of PROMs with low and high 4-bit "nibbles" from each memory block. The PROgram example in 3.1.3 indicate how this programming is done.

After RAM data has been split, a green indicator light next to the 16 BIT key is lit. Split data in RAM is indicated to a remote controlling device by the letter "S" appended to the remote prompt. (Remote control of the PROM programmer is described in Chapter 4.) The indicators are removed only when the 16 BIT key is pressed again, to effect a data Shuffle; thus the indicator light is a warning that the Shuffle operation is enabled. The Shuffle Function is actually performed when the ENTER key is pressed; it is the exact inverse of the Split Function, recombining low- and high-order blocks into 16-bit data in RAM.

To abort either a Split or Shuffle Function, press the " - " key instead of ENTER; the programmer returns to standby mode and RAM contents are unaffected.

The Split/Shuffle Function requires RAM capacity twice as large as the chip for which the programmer is equipped. Thus standard RAM (8K bytes) cannot accomodate a Split when the PROM programmer is equipped for a 64 K-bit PROM. Use of the optional RAM expansion unit (Chapter 5) makes these operations possible.

The following paragraphs indicate additional functionality associated with Split/Shuffle operations with the MDM personality module (use of the MDM personality module is described in 6.3).

Additional flexibility for Split/Shuffle operations is provided with the MDM module: the initial Split operation may be performed over an address range specified by the user, rather than a range encompassing the entire current device type.

When you press the 16 BIT key, the hexadecimal default block size (i.e., size of the currently selected device type) is displayed. To select a different block size, press the " " key to double the size of the block up to a maximum of *4000.* Additional strokes on the " - " key cause the displayed size to "wrap-around" to the minimum (20) and continue doubling from that point.

When the desired block size is displayed, press the ENTER key to cause the Split operation to be performed. The RAM address range used is twice as large as the selected block size, starting at user RAM address *0000.* Thus the selected block size becomes the first address of the block resulting from odd-numbered byte locations, and all lower addresses contain data resulting from even-numbered byte locations.

RAM locations with addresses greater than (2 x Split block size) are unaffected by this selective Split/Shuffle. When a Shuffle operation is performed with MDM, it automatically uses the address range specified for the last Split; the range for a Shuffle operation is not separately alterable. On the MPP-80S with MDM, the center horizontal segment of the sixth display element is lit to signify Split status (corresponding to the LED next to the 16 BIT key on the EPP80).

3.4.7 FUNCTION CONTROL

You can press the FC (Function Control) key to alter programmer operating conditions and perform hardware check routines. The actual function performed is denoted by one of the following codes:

Code	Function	Description
00	Recall software revision level	
02	Display Flash ON	These codes control flashing of the open (modifiable) field in the programmer display, or display an a remote device. The default setting is 02 (i.e., ON).
03	Display Flash OFF	
04	Inspect/Change EOF Character	Displays the current RS-232C serial communications EOF character (default is hex lA = Control Z). The EOF character is included in the output for all transmission protocols except 33, 34, 47, 48, and 60 (TransKon). The EOF character is not required for input.
05	Send a Character	Accepts a 2-digit hex coded character and writes it to serial Port A each time ENTER is pressed.
06	Software Handshake ON	Handshake ON (the default) permits serial output via either Port A or Port B to be interrupted by Control- S, and resumed by Control- Q.
07	Software Handshake OFF	
08	Byte Counter ON	Activates a byte counter which counts the number of bytes received over the RS-232C interface, and automatically displays the count when transmission is complete. Default is OFF.
09	Byte Counter OFF	
10		Display RAM size (hex)
11		Display PROM size (hex)

18	Intelligent check sum ON	Intelligent or additive check sum is performed alternatively. Default is 18.
19	Additive check sum ON	
77	Fill RAM with Test Data	Loads each RAM location with the 8 least-significant bits of its address.
A0	Exercise PROM Socket Address Lines	
E0	Enable Error Handling	Controls detection and display of errors during PROgram and CHecK operations. The default is E0; El is primarily for repair and calibration.
El	Disable Error Handling	

Additional functions are available if the MDM intelligent personality module is installed (see 6.3).

3.4.8 IEEE FUNCTION

The IEEE key performs no operation an the standard EPP-80 PROM programmer; it can be employed as an alternative function escape key. If the EPP-80 is equipped with the optional IEEE-488 (GPIB) interface, consisting of specialpurpose hardware and firmware, then pressing this key enables the PROM programmer to communicate and be controlled over the bus to which it is physically connected.

On units equipped with an IEEE interface, the programmer continues to operate normally in local mode after the IEEE key is pressed. The GPIB-enable status of the PROM programmer is indicated by the green LED next to the IEEE key an the PROM programmer front panel.

When the FROM programmer is addressed over to the bus, it automatically switches to GPIB remote control mode; this is noted in the programmer display, which shows "IEEE R xxxx" instead of "READY FOR xxxx". While the programmer is in GPIB remote control mode, the entire programmer keyboard is disabled except the IEEE key itself. If you press the IEEE key while the programmer is in GPIB remote control mode, the PROM programmer returns to local standby mode and the keyboard is once again operational.

The protocols used for GPIB remote control are similar to those used for serial remote control (i.e., those described in 4.2).

3.4.9 SEARCH FUNCTION

You can press the SRCH key to initiate a search of RAM for a particular string value. The string to be searched for may be up to eight hex digits Jong, and may include "wild card" (i.e., "don't care") digits signified with the " - " key. After you enter the string to be searched for, press ENTER to begin the search.

If a match to the specified string is found, the RAM address of the first byte in the matching string is displayed, as well as the data found at that location (i.e., the first byte of the "search string"). To continue the search with the same search string, press ENTER again. You may also press the SET key to alter RAM data at this time; the SET function uses the "found" address as its Starting Address. Press the " - " key to terminate the Search Function. If the search is unsuccessful, the message "NOT FOUND" is displayed.

The string you enter is saved; the next time the SeaRCH key is pressed, that string is part of the initial display. You may alter a previously entered search string or use it again.

SRCH Example:

Search for the string "02207B" in RAM, and change the second occurrence to "7D2003".

Press	Diplay is...	And it means
SeaRCH	SEARCH	Search Function initiated; enter desired search string.
0 2 2 0 7 B - -	SEARCH 02207B--	Six of eight possible hex digits are specified.
ENTER	---BUSY---	Search of RAM in progress.
	FOUND 0051 02	A matching string is found; it starts at address 0051.
ENTER	---BUSY---	Continue searching.
	FOUND 05FF 02	Another match is found; first byte is at address 05FF.

SRCH Example (continued)

Set	SE SA 05FF 02	A Set Operation is begun; Starting Address is 05FF.
7 D	SE SA 05FF 7D 02	New value for this location.
ENTER	SE SA *0600* 20	Next RAM address and value; note that this value is the second byte of the search string.
ENTER	SE SA 0601 7B	Value at location 0600 accepted; new address and value displayed.
0 3	SE SA 0601 03 7B	New value for this location.
ENTER	SE SA 0602 FF	Next location.
REMote	RE xxxx	Resetting from Set Operation.
REMote	Ready for xxxx	Programmer reset to standby mode.

3.4.10 COMPRESS FUNCTION

When you press the COMPress key an the EPP-80 Special Function keypad, the programmer begins Compressed Operating Mode. In this mode, starting and ending addresses for the PROgram, CHecK, LOAD, INput, and OUTput functions cannot be entered; instead, the addresses are assumed to be the same as the last set of starting and ending addresses you entered before invoking Compressed Operating Mode. If you have not specified any address ranges before pressing COMPress, default addresses for the entire PROM are assumed. Compressed Operating Mode can save you many keystrokes, especially when you need to repetitively program or check the same type of chip.

For example, the following keystrokes are required to Verify (with CHecK 1) an entire chip in normal (non-COMPressed) operating mode:

Press	Display is
CHK	CH 00
1	CH 01
ENTER	CH SA 0000
ENTER	CH EA 07FF
ENTER	CH P>
ENTER	---BUSY---

This Operation in Compressed Operating Mode requires the following keystrokes:

Press	Display is
CHK	CH 00
1	CH 01
ENTER	CH >
ENTER	---BUSY---

Completion messages and error displays associated with the PROgram function are not different in Compressed Operating Mode. However, CHecK functions performed while in Compressed Operating Mode result in only a PASS or FAIL message; the address and data that caused the failure are not shown.

Use of Compressed Operating Mode is not effective if you need to change Starting and Ending Addresses. A green indicator light next to the COMPress key is lit whenever the programmer is in Compressed Operating Mode. To return to normal operating mode, press the COMPress key again; the indicator light turns off.

The Compressed Operating mode can also be invoked while the programmer is controlled from a remote device; the Letter "P" is appended to the remote prompt to signify that the PROM programmer is in remote Compressed Operating Mode. See Chapter 4 for more about remote operation of the PROM programmer.

3.4.11 LIST FUNCTION

You may use the LIST function to view RAM contents on the PROM programmer display panel. Data is displayed in any of the formats: Binary, ASCII, Octal, or Decimal. Use of the LIST key provides data viewing capabilities similar to those associated with the SET function, but without the capability to change RAM contents.

After you press the LIST key, you select a data display mode by pressing the "B", "A", "0" (zero), or "D" key on the data keypad. You may then enter a Starting Address or accept the default address *0000.* Press the ENTER key to advance to the next RAM location, or the - " key to decrement the address.
You may enter the SET Function directly; the current LIST Function address becomes the Starting Address for SET.

The Binary and ASCII list modes present addresses in hexadecimal format; Octal list mode displays include octal addresses; Decimal list mode displays include decimal addresses. If you enter a Starting Address, it, must be siltered in hex format; the programmer converts it to appropriate listing format when ENTER is pressed.

LIST Example:
Inspect data values in RAM addresses *0300* through 0309, using Binary listing mode.

Press	Diplay is...	And it means
	FC 2732	Programmer is in standby mode, equipped for 2732 EPROM.
LIST	BIN ASC OCT DEC	Choose listing format;initial letters may flash.
B	B 0000	Binary mode selected; default Starting Address is 0000
3 0 0	B 0300	New Starting Address is 0300.
ENTER	B 0300 101101000	Current value at location 300.
ENTER	B 0301 101000110	Current value at location 301.
ENTER	B 0302 . . .	Current value at location 302.

LIST Example (continued)

Press	Diplay is...	And it means
ENTER	B 0309 00000101	Current value at location 309.
SET	SE SA 0309	SET Function begun;Starting Address is 309.

3.4.12 UV TIME CONTROL FUNCTION

When you press the UV key an the EPP-80 Special Function keypad, the time setting (in minutes) for the built-in Ultra-Violet EPROM eraser is displayed. If the eraser is not currently operating, this setting indicates the number of minutes the lamp will remain lit after the eraser is activated. If the eraser is currently on, the displayed number is the number of minutes remaining for the current erase operation.

Note that EPROMs to be erased should be inserted window side up in the built-in eraser drawer.

The default time setting for the UV eraser is 60 minutes. Consult the chip manufacturer's recommendations to verify that one to five EPROMs of the desired type can be adequately erased with the *.005* W/sq.cm lamp. You may reset the time that the lamp is an to any number of minutes in the range 45 to 99 by entering the appropriate decimal digits on
the data keypad after pressing the UV key. Pressing ENTER activates the eraser lamp, utilizing the displayed time limit.

To turn off the UV lamp, press the UV key and then the - key. The small LED next to the key is an whenever the lamp is on.

UV Example:
Set the UV timer to 75 minutes and activate the eraser lamp.

Press	Diplay is...	And it means
	FC 2716	Programmer is in standby mode, equipped for the 2716 EPROM
UV	UV TIME 60 MIN	Default time for eraser is 60 minutes.
7 5	UV TIME 75 MIN	Time is reset to 75 Minutes; place EPROMS in the eraser drawer before proceeding to the next step.
ENTER	FC 2716	Eraser has been started (signal light is on) and the programmer is now in standby mode.Additional PROM programmer functions can be performed.

3.4.13 MODULE-SPECIFIC FUNCTIONS

Certain intelligent personality modules (e.g., the MOS Device Module, AMD PAL module) make available special functions appropriate to the group of chips they support. You activate these functions with the #A key an the EPP-80 Special Function keypad. On an MPP-80S equipped with such personality modules, the functions are made available when you press the "A" key an the data keypad while the PROM programmer is in standby mode.

See Chapter 6 for information about the personality module you are using, and descriptions of the special functions provided via this key (if any). If the personality module currently mounted an the programmer does not incorporate special functions, the "#A" key (on the EPP-80), or "A" key (on the MPP-80S in standby mode) has no effect.

CHAPTER 4

COMMUNICATIONS

This chapter describes basic PROM programmer communications facilities. Included are descriptions of the communications interfaces, remote operation of the programmers, and use of the remote RAM editor.

4.1 COMMUNICATION INTERFACES

The MPP-80S and EPP-80 are each equipped with a serial Interface port (RS-232C and 20 mA current loop). These serial interface connections are implemented in one plug, which is located on the top panel of the MPP-80S and on the rear panel of the EPP-80 (Port A). The EPP-80 can be optionally equipped with another dual-purpose serial interface, with an IEEE 488-1978 (GPIB) interface, or with a parallel printer interface in Port B.

Internal connections for the 25-pin serial port connector are indicated in Table 4-1. Listed connections apply both to standard and optional serial interfaces; the optional serial interface for EPP-80 Port B is electrically equivalent to the standard serial interface (Port A).

The serial Interface uses an 8251A USART chip with level translation and buffering circuitry to achieve an electrically-isolated 20 mA current loop interface and a partial implementation of the RS-232C standard interface.

4.1.1 RS-232C CONNECTION

The recommended connection for RS-232C communications with the PROM programmer utilizes Pins 2, 3, and 7 only; pins 5 and 20 may optionally be used for hardware handshake, but all other pins should be left open. Connections to pins other than Pin 15 should not result in damage to the programmer, but may interfere with efficient communication. Pin 2 carries output data from the programmer, Pin 3 carries input data to the programmer, and Pin 7 is the Signal Ground connection.

This use of Pins 2, 3, and 7 applies for both data communications (i.e., INput and OUTput operations, 3.3.1 and 3.3.2) and serial remote control (described in 4.2). Data communications and remote control of the programmer are each possible for any supported baud rate; however, certain precautions (noted in following paragraphs) should be employed when a remote computer controls the PROM programmer.

Table 4-1. Serial Interface Connections

Pin		**RS-232C**	**Programmer**
Number	**Name**	**Line**	**Function**
1		GND	Protective Ground
2	BA	TD	Transmitted Data
			(output)
3	BB	RD	Received Data (input)
4	CA	RTS	Not implemented,always OFF (output set to high)
5	CB	CTS	Clear To Send,enables transmit (input); enabled if not connected
7	AB	SG	Signal Ground, Current Loop Out Return
8	-	-	1K Ohm to +12 V,Current Loop Input Current Source
14	-	-	20 mA Current Loop Data Output
15	-	-	+12 V, for MPP-80SAM modern (max. 50mA)
16	-	-	20 mA Current Loop Data Input (+)
18	-	-	20 mA Current Loop Data Output (-)
20	CD	DTR	Data Terminal Ready,indicates readiness to receive data

Note: Serial connector pins not included in this table are not internally connected.

Although data input (without handshaking) can proceed at any baud rate without overrunning the data receiving capability of the PROM programmer, remote control characters (i.e., commands) generally should not be sent at the full speed permitted by the baud rate, unless the baud rate is *300* or less. Thus overrun is possible when the programmer is remotely controlled by a computer (not a terminal).

Hardware handshake connections are available to prevent an overrun of remote control characters at rates above 300 baud. Typically, the user remotely controlling the PROM programmer from a computer needs to download rather large data files to the programmer and wishes to do so at a higher baud rate. In case hardware handshake is not available this can be accomplished as follows:

> Establish the serial connection at the desired (high) baud rate, and program the controlling computer to wait at least 250 ursec after sending each character in a remote command string. This procedure guarantees that the programmer will not be overrun while receiving a remote command. Data input can take place at full speed after the command and its parameters have been accepted.

4.1.2 20 mA CURRENT LOOP CONNECTION

The current loop connection to the programmer uses Pins 7 and 14 for data output to another device. Pin 14 is the +20 mA current source, and Pin 7 provides the current return path. Typically, these pins are connected to the electrically-isolated LED of an opto-isolator in the external circuit; Pin 14 connects to the external anode and Pin 7 to the external cathode.

Current loop data input to the programmer is via Pins 16 and 18, which are directly connected to the anode and cathode (respectively) of the LED in a TI 111 opto-isolator.

If data input originates with a device that actively sources current, then Pin 16 should be connected to the +20 mA source pin and Pin 18 should be connected to the current return pin.

If data input originates with a passive (isolated) switch that does not source the current, then Pin 8 of the programmer interface connector (1K Ohm to +12V) should be used as the current source to one siele of the external switch. Current return from the switch should be directed to Pin 16, and Pin 18 should be connected to Pin 7 (Ground) to complete the path for the internally-sourced current.

4.2 REMOTE CONTROL

Either the MPP-80S or EPP-80 PROM programmer (port A only) can be controlled by a remote device (terminal or computer) through the built-in serial interface. Necessary wiring of the connector is previously indicated in subsection 4.1.1. Programmer functions to be performed are indicated by ASCII characters sent from the remote device, as indicated in Table 4-2 and accompanying numbered notes (#n).

Following the Table are examples which illustrate the use of the Remote commands LOAD (R), CEecK (V), PROgram (P), INput (I), OUTput (0) , and Test (T).

Before the programmer is switched to remote operating mode, you should verify the baud rate, and change it as necessary so the PROM programmer is transmitting at the same rate as the remote device. This is accomplished with the RS232 key on the EPP-80 or the "B" key on the MPP-80S,,as described in subsection 3.4.3. After the baud rate of the programmer is properly set, press the REMote key once; the PROM programmer displays "rE xxxx" to indicate that remote mode is operative. While the programmer is in remote operating mode, all local keys except REMote are inoperative. The local operator can press REMote again to return the PROM programmer to local operating mode.

When you press the REMote key, the ASCII characters "R<CR><LF>" are sent over the RS-232C Interface. The character "R" serves as a prompt to the remote device/operator. The EPP-80 also transmits a remote command menu and the announcement "REMOTE xxxx" if the "Remote Comment" switch (DIP switch S8) on the back panel was ON during the last power-up initialization sequence.

If the PROM programmer display indicates that remote operation has begun but no message is received by the remote device, check the baud rates for the two devices and/or the physical connection at the serial port. If the PROM programmer seems to "lock up" when the REMote key is first pressed (e.g., no remote message is displayed on the programmer), an improper electrical state may exist at the Interface; for example, the remote device may not be powered up. You should disconnect the Interface cable; if the PROM programmer then responds with a valid display, the "lock up" problem is attributable to the remote device/connection.

The Q command terminates remote operation. After the PROM programmer receives "Q", it reverts to local standby mode -- unless the character is part of a test message (see Test command example at the end of this subsection). Further remote Operation is then enabled only by pressing the REMote key on the programmer once again.

Table 4-2. Remote Commands

Remote	Corresponding	Function	See Also	
Command	Local Key			
S	SET	Change RAM contents	3.2.1	
N	INSert	Insert data into RAM	3.2.2	
K	DELete	Delete data from RAM	3.2.3	
M	MOVe	Move data within RAM	3.2.4	
I	INput	Receive data at MPP/EPP	3.3.1	
O	OUTput	Send data from MPP/EPP	3.3.2	
P	PROgram	Copy RAM contents to chip	3.1.3	
V	CHecK	Verify RAM, chip, etc.	3.1.2	
R	LOAD	Read data from chip to RAM	3.1.1	
B	B	Set RS232 Baud Rate	3.4.3,	#1
C	C [Call]	Execute Z80 program in user RAM	3.4.4	
E	E [INV]	Invert RAM contents	3.4.2	
F	F [FILL]	Fill RAM with constant	3.4.1	
T		Display Test (send message)		#2
(SP)	ENTER	Execute or increment address	2.3	
-	"-"	Cancel or decrement address	2.3	
G	COMPress	Enter or exit compressed mode	3.4.10,	#3
Q	REM-REM	Reset to Local Standby	3.3.3	
Y	SeaRCH	Search RAM for a string	3.4.9,	#3
U	UV	Control UV eraser	3.4.12,	#4
X	16 BIT	Split/Shuffle RAM contents	3.4.6,	#3
L	LOGic	Do logical Operation	3.4.5,	#4
-> **		Enter remote RAM editor	4.3,	#3

**:(for cursor right use CNTRL F/L/P)

Notes for Table 4-2:

#1 The remote Baud Function is not available with the EPP-80. Unless the baud rates on the MPP-80S PROM programmer and the remote device are simultaneously reset, changing baud rates during a transmission breaks contact.

#2 The T command displays a 16-character message on the PROM programmer display panel. Display limitations associated with the MPP-80S dictate the following restrictions:

The next sixteen characters (upper-case ASCII only) entered on the remote device after "T" are displayed on the PROM programmer. The MPP-80S can only display the following: 0-9, - , A, C, E, F, H, I, L, P, S, and "b." for B, "d." for D, nrin for M, "n" for N, "o" for 0, "r" for R, urn for T, "c" for V, and (Blank) for Space and any other characters not listed above. To fully test the MPP seven-segment display, enter sixteen "8"s. After a T command is begun, no other remote command is accepted by the MPP-80S until 16 characters are transmitted.

The EPP-80 accepts a message of any length (terminated by the <ESC> character) and displays all printable ASCII characters.

#3 These functions are available only with the EPP, unless the MDM personality module (described in 6.3) is installed on the MPP-80S.

#4 Property of EPP-80 only.

In the following examples of remote commands, the column headed "Terminal" indicates ASCII characters entered at the remote device and transferred over the RS-232C interface to the PROM programmer. The column headed "Display" contains the information displayed on the programmer itseif and echoed to the remote device; for the MPP-80S, this echo occurs only if internal DIP switch S5 was open during initialization (see 5.1.1) and only after a space (<SP>) is received by the PROM programmer.

With the EPP-80 and/or MDM personality module, remote displays are updated simultaneously with PROM programmer displays if this function is enabled. Such remote update is enabled by setting MPP-80S internal DIP switch S5 to OPEN (when the MPP-80S is equipped with MDM), or by setting the EPP-80 "remote comment" DIP switch (on the rear panel) to ON. See Chapter 5 for more information about programmer DIP switches, and see 6.3 for details about the MDM personality module.

Remote LOAD Example:

Assume that the PROM from which data is to be loaded into RAM is properly mounted an the PROM programmer. The basic LOAD operation is described in 3.1.1.

Press	Diplay is...	And it means
R	rE 2716	Programmer is in remote standby mode; the R command initiates a LOAD operation.
<SP>	Ld SA *0000*	Default starting address in RAM is *0000;* different hex address can be transinitted from the remote device.
<SP>	Ld EA *07FF*	The <SP> key operates like ENTER, accepting the start address.Default ending address is displayed. A different hex address can be selected at the remote keyboard.
<SP>	Ld P -1	Insert device.
	----BUSY----	Loading is begun when <SP> is received. Display indicates that loading was successfurly completed; programmer is in standby mode.
<SP>	rE 2716	

Remote CHecK Example:

Perform Check 4 (RAM checksum). The basic CHecK operations are described in 3.1.2.

Press	Diplay is...	And it means
V	rE 2716	Programmer is in remote standby mode; the V command initiates a Check ("Verify") operation. Space <SP> must be entered next.
<SP>	Ch *00*	Check number *0* (blank check) is the default; enter 1, 2, 3, or 4 to select another type of CHecK.

Remote CHecK Example (continued):

Press	Diplay is...	And it means
4	Ch SA *0000* 04	Check 4 selected.Starting address default is *0000;* enter another if required.
<SP>	Ch EA 07FF	Start address accepted. Change End Address if required.
<SP>	ChESU xxxx	Checksum displayed upon completion of the check.
<SP>	rE 2716	Operation complete; programmer in remote standby mode.

If errors are discovered, they are displayed an the PROM programmer and the remote device. For example:

```
ECH1 0101 FF 62
```

where the fields indicate: error ("E"); CHecK Operation number ("CH1"), address ("0101"), PROM data ("FF"), and RAM data ("62"), just as in local operating mode.

Remote PROgram Example:

Program a mounted PROM with data from PROM programmer RAM. The basic PROgram Operation is described in 3.1.3.

Terminal	Diplay is...	And it means
P	rE 2716	Programmer is in remote standby mode; the P command intiates PROgram Operation; it must be followed by <SP>.
<SP>	Pr SA *0000*	Start address is displayed; it can be changed. Accept with <SP>.
<SP>	Pr EA *07FF*	End Address is displayed; it can be changed. Accept with <SP>.
<SP>	Pr P -1	Insert device.
	----BUSY----	
<SP>	rE 2716	Operation complete; programmer in remote standby mode.

The PROgram operation includes automatic checks as described in 3.1.3. If errors are detected, they are displayed an the PROM programmer and the remote device.

Remote INput Example:

Send data to RAM over the serial interface. The basic INput Operation is described in 3.3.1.

Terminal	Display	Meaning
I	rE 2716	Programmer is in remote mode; the I command initiates an Input Operation; it must be followed by <SP>.
[<SP>	IN A or B	On EPP-80 with 2 ports, select port]
<SP>	In *00*	Enter digits to select format code.
40	In 40	Intel Hex format selected.
<SP>	OFFSE *0000*	Select offset for addresses contained in transmitted data.

Remote INput Example (continued):

Terminal	Display	Meaning
<SP>	IN PO	Polarity is positive; enter " - " for data that is inverted.
<SP>	--------------	Transfer in progress.
	rE 2716	Operation complete;programmer in remote standby mode.

Remote OUTput Example:
Send data from the PROM programmer RAM over the serial interface. The basic OUTput Operation is described in 3.3.2.

Terminal	Display	Meaning
0	rE 2716	Programmer is in remote mode; the 0 command initiates an OUTput Operation; it must be followed by <SP>.
[<SP>	OU A or B	On EPP-80 with 2 ports,specify port]
<SP>	OU 00	Select output format code.
44	OU 44	Format 44 (Signetics Absolute) selected. Address selection is not required for this format.
<SP>	-------------	Data transfer in progress.
	rE 2716	Operation complete; programmer in remote standby mode.

Remote Test Example:
Send a message from the remote device for display an the PROM programmer display.

Terminal	Display	Meaning
T	rE 2716	Programmer is in remote standby mode; the T command causes the next 16 ASCII characters to be interpreted literally, not as commands.
1234<SP>KDS-CONN3Q9	1234 KDS-Conn3Q	Any 16 allowable characters are transmitted and displayed. The seventeenth character (MPP-80S) or <ESC> (EPP-80) terminates the test
<SP>	rE 2716	Programmer returns to remote standby mode.

Note that the character "**Q**" terminates remote operation, except when it is entered as one of the 16 characters in the test message.

4.3 REMOTE **RAM** EDITOR

The EPP-80 includes a screen-oriented editor that permits you to inspect and alter programmer RAM from a remote terminal. This editor is enabled only if DIP switch S8 for the serial communications port was CLOSED at the last initialization of the PROM programmer, and only if the programmer is in remote operating mode. (For details about setting external DIP switches, see 5.2.2).

While the editor is operating, current contents of sixteen consecutive RAM addresses are displayed across the terminal screen. The lowest address has zero as its least-significant digit (e.g., "03C0"). You can direct the screen cursor to any data value in the displayed line and modify that value by entering replacement hex digits. You can view contents of the next sixteen RAM addresses, or the previous sixteen, by entering appropriate editor commands; you can also specify the starting address of any block of sixteen RAM locations to be displayed.

You invoke the editor from a terminal controlling the programmer by pressing the " -> " (cursor right) key. (Control)-F, (Control)-L, or (Control)-P can substitute for the cursor right key in this context. Editor commands are described in Table 4-3.

Table 4-3. Remote RAM Editor Commands

Key	Name	Description
<CR>	INCREMENT ADDRESS BLOCK	Display the next 16 RAM locations
U	DECREMENT ADDRESS BLOCK	Display the previous 16 RAM locations
T	SET ADDRESS BLOCK	Display 16 RAM locations beginning with address "xxx0", where "xxx" is entered immediately after "T".
-->	CURSOR RIGHT	Move the screen cursor to the right in the display line.
<--	CURSOR LEFT	Move the screen cursor to the left in the display line.
N	NEXT DATA FIELD	Move cursor to the next 2digit hex value an the right.
Q	QUIT	Terminate Editor execution; return to remote standby mode.

The Editor screen display consists of a brief summary of Editor commands and two lines containing RAM addresses and data values, as shown in Figure 4-1. The top line of the data display includes the label "ADDR" at the left, followed by the last two hex digits of the sixteen RAM addresses. These digits are always 00 through *OF* because the base address has zero for its least-significant digit.

The second line of the data display includes the 4-digit base address for this block of RAM locations, followed by RAM data values. Data values are located beneath the corresponding 2-digit address offsets in the first display line. To the right of the displayed hex data values are ASCII representations of these sixteen data values, enclosed by asterisks (*). Unprintable or non-ASCII data is denoted by a period.

You use the cursor movement keys "->" and "<-" or the Editor command "N" to move the cursor along the line of displayed
data. To alter a value, simply enter a replacement digit when the cursor is over the value to be changed. The "N" command moves the cursor to the next data field (i.e., the next two-digit value) to the right. At the end of the display line, this command returns the cursor to the left end of the data line.

To access RAM locations not in the current display, enter <CR> for the next set of sixteen locations or "U" for the previous sixteen. You can also specify a new base address with the "T" command: enter three hex digits immediately after "T"; these are the three most-significant digits for the new base address. For example, to display RAM locations 05D0 through 05DF, enter "T05D".

When you enter "Q", the Editor terminates and you can enter any of the PROM programmer remote commands previously described in subsection 4.2.

```
ADDR 00 01 02 03 04 05 06 07 08 09 0A OB OC OD OE OF
0140 1B 02 20 4B 6F 6E 74 72 6F 6E 21 20 03 1B 1B FF *..Kontron!..*
```

Figure 4-1. Remote RAM Editor Data Display

CHAPTER 5

USER-MODIFIABLE HARDWARE OPTIONS

This chapter presents directions for modifying PROM programmer characteristics that are controlled by DIP switches and for installing the optional RAM expansion board. Physical differentes in the processors necessicate separate explanations for the MPP-80 (5.1) and the EPP-80 (5.2). Turn off either PROM programmer and disconnect the power cable before attempting any modification.

5.1 MPP-80 Modifications

	Disassembly of the processor is required to access DIP switches or RAM expansion board. To disassemble the unit, proceed as follows:
1	Unplug the AC power cord from the MPP-80 front panel.
2	Remove the four screws which secure the front panel to the case.
3	Lift front panel slowly, from the edge farthest from being careful not to tip the unit over.
4	Disconnect the two speaker wires, which are attached to in the bottom of the case by pressure fit connectors.
5	Rest the front panel upside down an the aluminum case; circuit side should be facing

To reassemble the unit, reverse the disassembly procedure.

5.1.1 DIP SWITCHES

DIP switches S1 through S8 an the MPP-80 are located near the center of the unit, as shown in Figure 5-1. Switches S1 through S4 determine the default baud rate for RS 232 communications; this is the assumed rate each time the processor is initialized. Table 5-1 indicates switch settings for supported default baud raten.

Note that the baud rate for data communications can be reset from the processor keyboard, as described in 3.4.3. Generally, it is advisable to change the default baud rate from 2400 baud to another rate only if that other rate is the only one to be used.

Figure 5-1 • MPP-80S Internal Layout

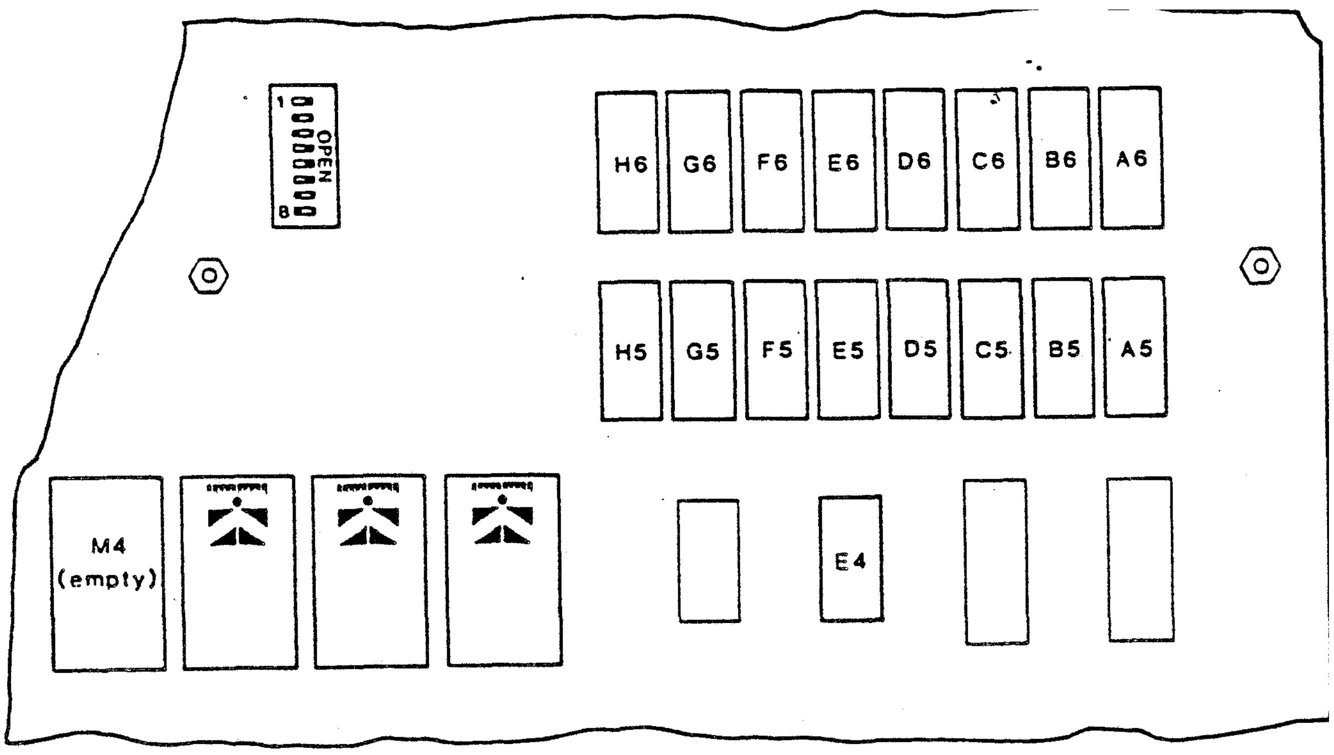


Table 5-1 • MPP-80 Baud Rates

Baud	S1	S2	S3	S4
50	CLOSED	CLOSED	CLOSED	CLOSED
75	OPEN	CLOSED	CLOSED	CLOSED
110	CLOSED	OPEN	CLOSED	CLOSED
134	OPEN	OPEN	CLOSED	CLOSED
150	CLOSED	CLOSED	OPEN	CLOSED
200	OPEN	CLOSED	OPEN	CLOSED
300	CLOSED	OPEN	OPEN	CLOSED
600	OPEN	OPEN	OPEN	CLOSED
1200	CLOSED	CLOSED	CLOSED	OPEN
2400	OPEN	CLOSED	CLOSED	OPEN
4800	CLOSED	OPEN	CLOSED	OPEN
9600	OPEN	OPEN	CLOSED	OPEN
19200	CLOSED	CLOSED	OPEN	OPEN
38400	OPEN	CLOSED	OPEN	OPEN

DIP switches S5 through S8 each control a separate feature of the PROM programmer, as indicated in following paragraphs. The factory default setting for each of these switches is CLOSED.

DIP switch S5 controls characters sent for remote display, as follows:

If S5 is CLOSED, the PROM programmer sends ASCII "R" followed by <CR> and <LF> upon completion of each operation. If an error occurs, the error message display on the processor front panel is also sent to the remote device.

If S5 is OPEN, upon receiving each ENTER or " — " command from the remote device the PROM programmer sends all information normally displayed on the processor front panel to the remote device. This can provide a confirmatory echo for a remote CRT.

DIP switch S6 controls the MPP-80 audio tone, as follows:

If switch S6 is CLOSED, the tone sounds only when processor initialization is complete and upon detection of an error.

If switch S6 is OPEN, the tone sounds for each key entry and upon completion of each operation.

DIP switch S7 can be used to limit processor operations, as follows:

If switch S7 is CLOSED, all operations are enabled.

If switch S7 is OPEN, only the programming functions PROgram, CHecK, and LOAD are enabled. This mode can be useful in a production environment to reduce the possibility of errors. The REMote key can be 'pressed twice to reset the programmer, but remote control of the processor is not possible.

DIP switch S8 controls processor seif—checks used during initialization and programming, as follows:

If switch S8 is CLOSED, internal checks are enabled; this is the intended operational setting.

If switch S8 is OPEN, internal checks are disabled; this setting is used by Kontron for device testing.

5.1.2 MPP RAM EXPANSION

The optional RAM expansion board can be installed by a user to increase processor RAM from 8K to 32K bytes. Programming of some large chips may not be possible without the increased memory.

RAM expansion is accomplished by installing a "piggy-back" printed circuit board. This board has three circuit connections to the main processor board: one built into the bottom of the piggy-back board, and a 16-pin and a 24-pin connector at the ends of ribbon cables. The 24-pin connector attaches to the (normally empty) socket M4. The 16-pin connector replaces the IC in socket E4.

Before installing the expansion board, remove the sixteen (16) RAM chips A5 through H5 and A6 through H6. Attach the ribbon cable connectors, then add the four threaded spacers (supplied with the expansion board) to main board retaining screws and rest the piggy-back board on the spacers. If the built-in connector on the expansion board does not seat properly, remove nuts below the spacers to position the new board closer to the main board. After the built-in connector is seated, afix nuts to secure the expansion board to the spacers.

5.2 EPP MODIFICATIONS

The EPP-80 PROM programmer includes three banks of DIP switches: there is a set of four switches inside the unit, and two banks of eight switches each on the back panel. Disassembly of the PROM programmer is not required to access the DIP switches on the back panel. To access the internal DIP switches or install the optional RAM expansion board, you must disassemble the unit as follows:

1. Unplug the AC power cord from the rear panel, remove personality module (if present), and carefully set the programmer face down on a flat surface.

2. Remove the six screws on the bottom panel.

3. Hold the top and bottom halves of the unit together and carefully turn the programmer upright (right side up).

4. Lift the top panel slowly, tilting it toward the rear of the unit. There will be three cables connecting the top panel assembly to the base.

5. Disconnect the three connectors (power line and I/O lines) from the main printed circuit board.

6. Set the top panel to the side of the rest of the unit, upside down with the printed circuit board facing up.

To reassemble the unit, reverse the disassembly procedure. Take care to firmly reseat the connectors.

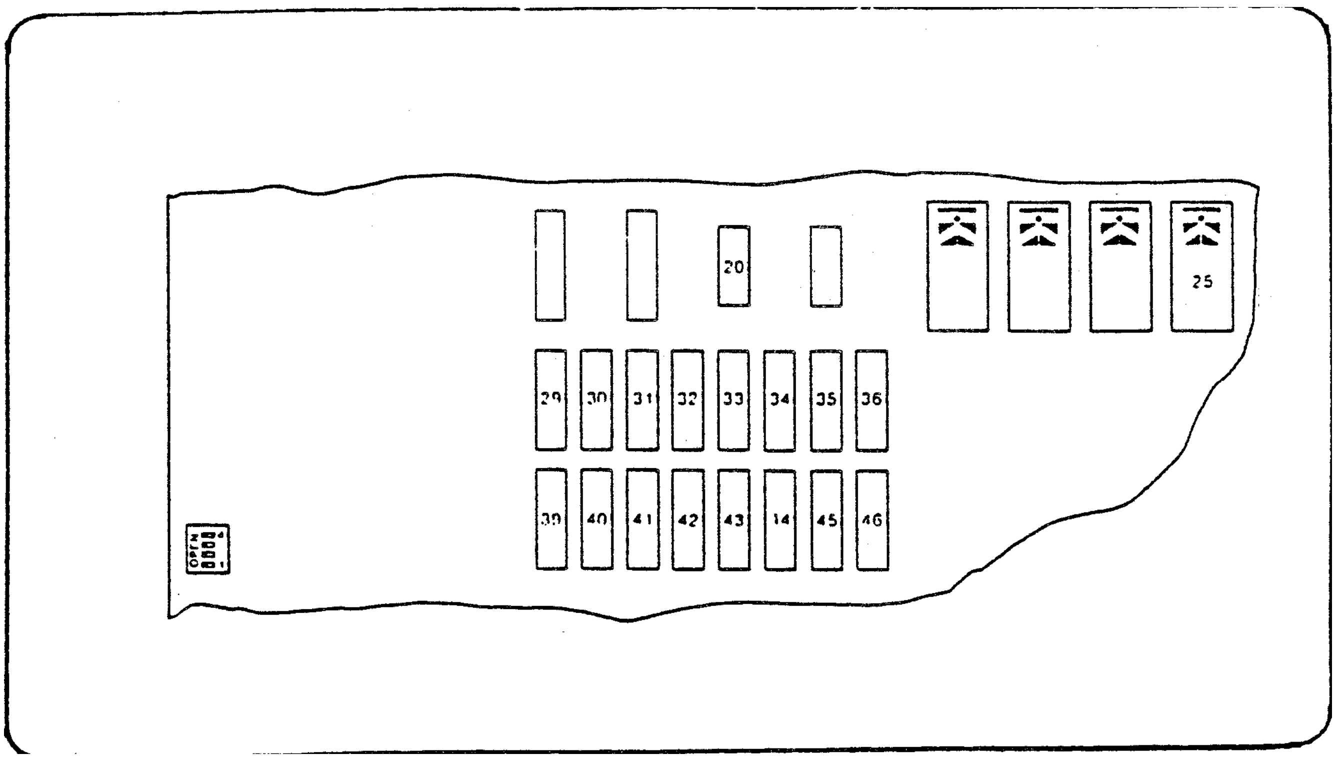
5.2.1 Internal Dip Switches

There is one set of four DIP switches on the main printed circuit board in the EPP-80. This bank of switches is locatcd near the Special function keypad,as shown in Figure 5-2. Switch Sl controls the ckecksum algorithm used for some CHKeck operation.Switches S2 and S3 together indicates the (optional) second communication interface; switch S4 contro1s the audio tone.
The factory default settings for these switches is closed.
Switch settings and corrusponding features are described in following paragraphes.

DIP switch S1 controls the algorithm for CHeck 3(PROM checksum) and CHeck 4 (RAM checksum), as follows:

> 1f switch S1 is CLOSED, a complex algorithm devised by Kontron is utilized for the checksums.These algorithm includes provisions to check for juxtaposed data, as well as data that is missing, or incorrect.
>
> If switch S1 is OPEN, a simple 16-bit binary byte addition algorithm is used for checksums; this algorithm is compatible with that used an DATA. 1/0 devices. Wien this algorithm is used, "+" is displayed to the right of the 4-digit checksum result.

Figure 5-2. EPP-80 Internal Layout



DIP switches S2 and S3 indicate to the PROM programmer which communications interface modules are installed; settings for these switches should be changed only if an optional interface is installed in the field. Switch settings are as follows:

If switch S2 is CLOSED and switch E3 is CLOSED, the processor is equipped with only a single RS232C serial port (Port A).

If switch S2 is CLOSED and switch S3 is OPEN, the processor is also equipped with a second RS232C serial port (Port B).

If switch S2 is OPEN and switch S3 is CLOSED, the processor is equipped with a parallel communications port for the Centronics printer.

If switch S2 is OPEN and switch S3 is OPEN, the processor is equipped with an IEEE communications interface.

DIP switch S4 controls the EPP-80 audio tone, as follows:

If switch S4 is CLOSED, the tone sounds only when processor initialization is complete and upon detection of an error.

If switch S6 is OPEN, the tone sounds for each key entry, when initialization is complete, and when an error is detected.

5.2.2 EXTERNAL DIP SWITCHES

There are two banks of eight DIP switches each on the back panel of the EPP-SO that control serial communications parameters; each bank of DIP switches is adjacent to and controls an RS232C communications port. If your PROM programmer has only one active port, it is Port A.

Settings of the external DIP switches are meaningful only to the processor intialization routines; after you change the settings of these switches, you must turn the power off and then on again to make the change effective. External DIP switches are clearly labeled as to function and setting.

DIP switches S1 through S3 determine the default baud rate for the associated RS232C communications port; this is the assumed rate each time the processor is initialized. Table 5-2 indicates switch settings for supported default baud rates. Note that the baud rate for either port can be reset from the processor keyboard, as described in 3.4.3.

DIP switch S4 controls the type of ASCII characters sent (and expected to be received) by the EPP-80, as follows:

If switch S4 is CLOSED, 7-bit ASCII characters are transmitted.

If switch S4 is OPEN, 8-bit ASCII characters are transmitted.

Table 5-2. Baud Rate Defaults for EPP-80

Baud	S1	S2	S3
110	OPEN	OPEN	OPEN
150	CLOSED	OPEN	OPEN
300	OPEN	CLOSED	OPEN
600	CLOSED	CLOSED	OPEN
1200	OPEN	OPEN	CLOSED
2400	CLOSED	OPEN	CLOSED
4800	OPEN	CLOSED	CLOSED
9600	CLOSED	CLOSED	CLOSED

DIP switch S5 controls the number of stop bits sent (or expected to be received), as follows:

If switch S5 is CLOSED, one (1) stop bit is transmitted.

If switch S5 is OPEN, two (2) stop bits are transmitted.

DIP switch S6 controls the sending of a parity bit with transmitted data, as follows:

If switch S6 is CLOSED, a parity bit is sent (or expected to be received) for each data byte. The type of parity is controlled by switch S7.

If switch S6 is OPEN, no parity bit is sent (i.e., "0" is sent in the parity bit position), and parity bits for incoming data are automatically set to 0. Note that switch S7 is ineffective if S6 is OPEN.

DIP switch S7 indicates the type of parity utilized for transmitted data, as follows:

If switch S7 is CLOSED, ODD parity is selected. If

switch S7 is OPEN, EVEN parity is selected.

DIP switch S8 controls characters sent for remote display, as follows:

> If switch S8 is CLOSED, the display on a remote terminal screen is updated several times per second, and the bottom line of that display always matches the display on the PROM programmer front panel. The sequence ASCII "R<CR><LF>" is sent when any operation is completed, and complete error messages are transmitted; either of these is followed by a complete menu of remote commands. Only when S8 is CLOSED is the Remote RAM Editor enabled; see 4.3 for information about how to use the editor.
>
> If switch S8 is OPEN, the EPP-80 sends ASCII "R<CR><LF>" upon completion of each operation (or when an operation is abnormally terminated). Complete error messages are also sent to the remote device, just as they appear on the processor display. The Remote RAM Editor is not available if S8 is OPEN.

5.2.3 EPP RAM EXPANSION

The optional RAM expansion board can be installed by a user to increase processor RAM from 8K to 32K bytes. Programming of some large chips may not be possible without the increased memory.

RAM expansion is accomplished by installing a "piggy-back" printed circuit board. This board has three circuit connections to the main processor board: one built into the bottom of the piggy-back board, and a 16-pin and a 24-pin connector at the ends of ribbon cables. The 24-pin connector replaces IC25 in the socket M4; IC25 is relocated to the expansion board. The 16-pin connector replaces IC20 in socket E4.

Before installing the expansion board, remove the sixteen (16) RAM chips IC29 through IC36 (locations A5-H5) and IC39 through IC46 (locations A6-H6). Attach the ribbon cable connectors, then add the four threaded spacers (supplied with the expansion board) to main board retaining screws and rest the piggy-back board on the spacers. If the built-in connector on the expansion board does not seat properly, remove nuts below the spacers to position the new board closer to the main board. After the built-in connector is seated, afix nuts to secure the expansion board to the spacers.

CHAPTER 6

PERSONALITY MODULES

This chapter describes use of personality modules in general, and special features of the "intelligent" personality modules UGM, MDM, PAL, and IFL.

6.1 INTRODUCTION

Personality modules connect to the parallel connectors on the top (front panel) of the MPP-80S or EPP-80 PROM programmers. Connectors are keyed; modules can be inserted only one way. You should not apply undo force when seating a personality module. • It is recommended that you routinely inspect modules for berat pins before installing them.

Socket adaptere are suited to individual personality modules; a socket adapter should be seated on the keyed connectors in the personality module. The appropriate socket adapter and personality module for a particular PROM device can be determined from the Kontron PROM Programmers Comparison Chart. The socket adapters provide proper pinout connections for selected chips.

Note that personality modules and socket adapters do not provide a check for devices inserted backwards. You need to verify that chips are installed in the socket adapter with pin 1 in the labeled location. Damage to the chip is possible if it is improperly installed.

Most personality modules and socket adapters can be replaced when the PROM programmer is on without loss of RKMdata. This can only be accomplished when the processor is in local standby mode; the MPP-80S display shows "Fcxxxx" and the EPP-80 display shows "READY FOR xxxx" in this state. To place the PROM programmer in standby mode following a local operation, press the REMote key twice (assuming that there is no connection at the RS 232C Interface). After a module is changed, press the ENTER key to access software associated with the new module; the subsequent display should include the model number of the new chip.

Some personality modules, notably the Universal Gang Module and the Integrated Fused Logic Module, require plug-in identifiers with firmware suited to the desired chip. Identifiers attach to connectors on top of the personality modules, and muss be installed for proper operation of the module.

6.2 UNIVERSAL GANG MODULE

The Universal Gang Module (UGM) includes sockets for programming as many as eight EPROM chips of the same type simultaneously. It attaches to the parallel connectors used for any personality module, and includes a separate Bocket for a master PROM. A suitable plug-in identifier configures the module for the desired chip type, which can be any of various single- or triple-power supply EPROMS having capacities as large as 64K.

All sockets on the UGM accept 28-pin Intel 2764 and Texas Instruments 2564 devices, as well as all 24-pin devices. Note that 24-pin devices should be inserted in the lower part of each socket, leaving the top four pin receptacles unoccupied; markings near the sockets serve to remind you of this.

The UGM supports all Check operations 0, 1, 2, 3, and 4, as described in 3.1.2. For each socket there is a pair of indicator lights; the red lamp lights if the corresponding PROM fails a test (or if the socket is unoccupied), and the green lamp lights if the PROM passes a test.

A master PROM can be read from the ninth socket which is labeled "MASTER". This socket cannot be used to program a device; the socket is "read only" to protect a master PROM.

Programming with the UGM is accomplished following the proaedures in 3.1.3; the actual programming operation is preceded by an automatic Check 02, and followed by a Check 01. If any programming socket remains empty during the programming operation, the appropriate red LED indicator is lit and error messages appear in the display.

6.3 MOS DEVICE MODULE

The MOS Device Module (MDM) is an intelligent, multi-device personality module that attaches to the module connectors on the EPP-80 or MPP-80S PROM programmers. This module permits you to program a wide variety of PROM, EPROM, and EEPROM devices without changing modules. It also provides additional operating features and functions, some of which are not otherwise available. In particular, the MDM module makes many advanced functions of the EPP-80 programmer available on the MPP-80S.

The standard MDM socket adapter (SA MDM) accepts all supported 24- and 28-pin chips. An LED indicator on the socket adapter is lit to indicate where pin 1 of the selected device should be inserted. Note that it is the operator's responsibility to ensure that chips are properly inserted, although the MDM module checks for insertion errors, short circuits, and incorrect device types.

Other socket adapters may also be used with the MDM personality module. Consult the Kontron PROM Programmers Comparison Chart to determine the appropriate socket adapter for the chips you require.

When you use the MDM module, PROM programmer software is bypassed; operations are controlled by the MDM software. However, most operations with the MDM module proceed as with basic programmer software that documented in this manual. The MDM module should not be removed while programmer power is on , loss of RAM data is probable, and the programmer will not function properly. To change to another personality module, you should turn off power to the programmer.

6.3.1 DEVICE SELECTION

When the MDM personality module is installed in the MPP-80S or EPP-EC PROM programmer during power-up, automatic initialization checks include the MDM firmware as well as that of the programmer. After processor initialization is complete, the message "DIAL TYPE xxxx" is displayed; "xxxx" is the default PROM device model number. You can change the default number by following the procedure described in 6.3.2.

If the displayed model number corresponds to the chip you intend to work with, press ENTER to initialize MDM. Otherwise, press " - " to cause another device model number to appear. You can successively press this key to "step through" recognized model numbers; press ENTER when the proper number is displayed. Alternatively, you can enter the model number on the data keypad and then press ENTER. If the device model number you key-in is not supported by MDM, it is ignored when you press ENTER.

When MDM initialization is complete, the message "READY FOR xxxx" is displayed. If you have the SA-MDM socket adapter mounted on the personality module, an LED is lit next to the proper receptacle for Pin 1 (this is necessary because the adapter physically accepts both 24- and 28-pin chips, but the software supports on only one at a time).

The Device-Select procedure described above can be repeated during normal processor operation. You need only press the "#A" key (on the EPP-80) or the hex "A" key (on the MPP-80S); the message "DIAL TYPE xxxx" appears, and you can proceed to reselect the device model number. The Device-Select procedure has no effect on processor RAM; therefore you can load the RAM from one model chip and use that data to program another model, all without changing the personality module.

Although the MDM personality module should not be removed while the PROM programmer is receiving power, you may change the socket adapter at any time that the programmer is idle (i.e., in standby mode). A change of socket adapters will probably necessitate a new Device-Select operation.

The MDM software verifies that the installed socket adapter matches the selected PROM device model number during each PROgram, LOAD, and CHecK operation. If the socket adapter is not the correct one for the currently selected chip, the message "ERROR SAD" is displayed and the operation is aborted.

The MDM personality module and general-purpose socket adapter SA-MDM together support the following programmable devices:

2704	2516	2532	6653	68764
2508	2716	2732	6654	27128
2708	TMS 2716	2732A	6657	3704
2758	2816	2564	6658	3708
2758A	48016	2764		

Each supported device is identified to the processor by the model number above except the TMS 2716 device, which is selected as "2716-3".

The following devices can also be programmed with the MDM personality module, using the indicated socket adapter:

Device	Adapter
2920	SA33
8741	SA8
8748	SA8
8755A	SA14

Device	Adapter
8749	SA8-2
8751	SA36
8755	SA14

Support for newly available MOS devices is acquired by installation of upgraded MDM and/or PROM programmer firmware. Check with your Kontron representative for availability.

6.3.2 DIP SWITCHES

A bank of four DIP switches are located on the MDM circuit board as shown in Figure 6-1. The setting of switches S1 through S3 determine the default device selection for the module; Table 6-1 indicates possible switch settings and corresponding devices. You can change the settings of switches Si, S2, and S3 to correspond to a chip model that you frequently use; switches are easily accessed through the socket adapter opening.

MDM DIP switch S4 is currently unused; it is reserved for future options.

DIP switches located on the MPP-80S and EPP-80 PROM programmers retain their definitions (given in Chapter 5) with one exception: MPP-80S internal DIP switch S7 is ignored when the MDM personality module is installed, because MDM is not intended for use in a production environment.

Figure 6.1 MDM Dip Switches

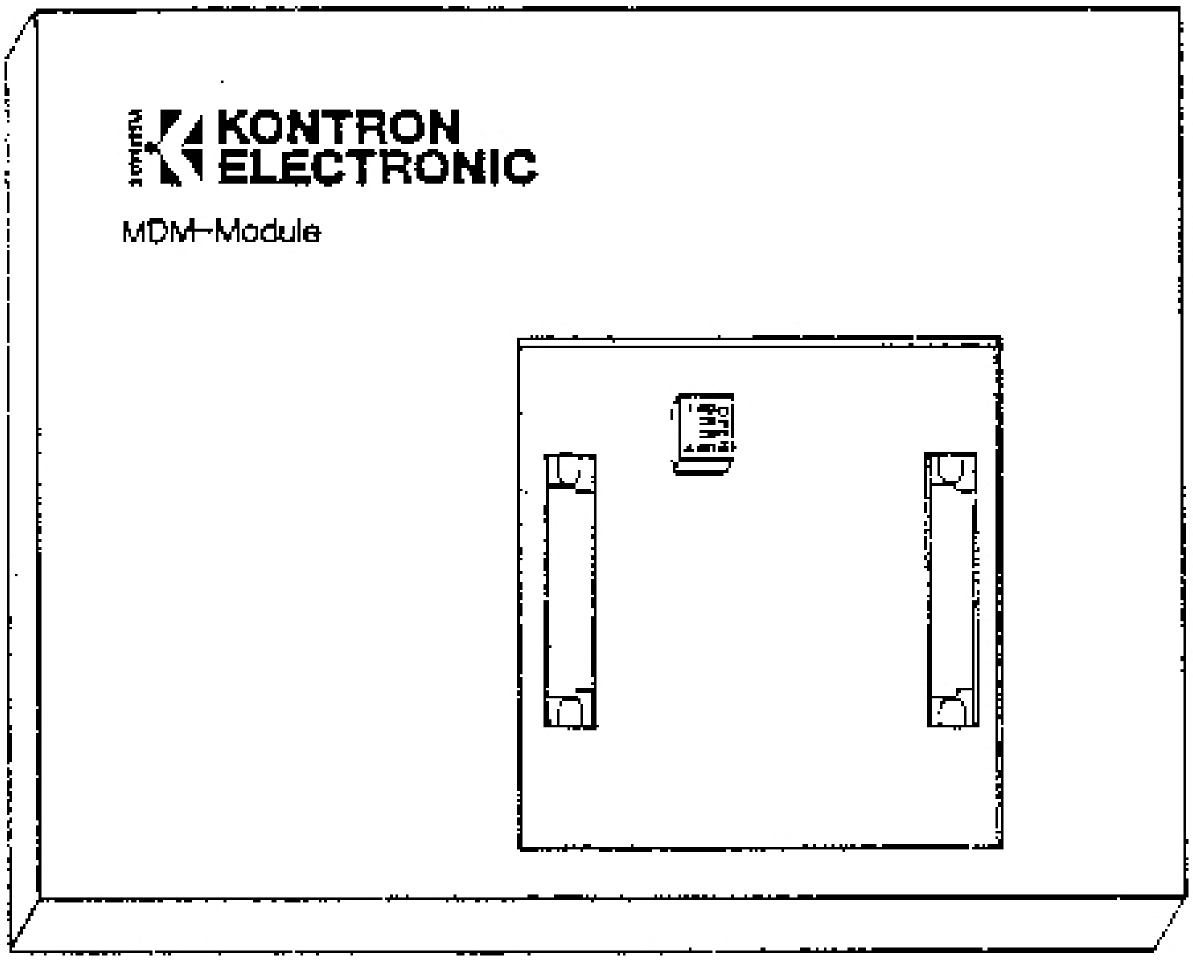


Table 6-1. MDM Dip Switch Settings

Devi ce		S1	S2	S3
2706		CLOSED	CLOSED	CLOSED
2716		CLOSED	CLOSED	OPEN
2532		CLOSED	OPEN	CLOSED
2564		CLOSED	OPEN	OPEN
2516		OPEN	CLOSED	CLOSED
2716-3	(TMS 2716)	OPEN	CLOSED	OPEN
2732		OPEN	OPEN	CLOSED
2764		OPEN	OPEN	OPEN

6.3.3 ADDITIONAL FUNCTIONS

Many functions of the advanced EPP-80 PROM programmer are made available to users of the MPP-80 programmers with the addition of the MDM personality module. Hex data keys on the MPP-80S (when the processor is in local standby mode) replace some of the special function keys of the EPP-80. Special functions and differences between MDM operation and EPP-80 operation are described in following paragraphs, which each include a reference to the subsection in which the Special Functions are thoroughly described.

SeaRCH

Described in 3.4.9; invoked with the hex "D" key on the MPP-80S when the programmer is in local standby mode.

COMPress

Described in 3.4.10; invoked with the hex "7" key on the MPP-80S when the programmer is in local standby mode.

16 BIT Split/Shuffle

Described in 3.4.6; invoked with the hex "9" key on the MPP-80S when the programmer is in local standby mode. Added functionality is the PROM length field "yyyy" in the display "Sp 8-->16   yyyy". The value initially displayed corresponds to the entire length of the selected chip (in bytes). You change the displayed value (and range for the Split/Shuffle function) by pressing " - " to step through the allowable values: 20,40,80,100,200,400,800,1000,2000,4000.

Function Control (FC)

Described in 3.4.7; invoked with the hex "8" key on the MPP-80S when the programmer is in local standby mode. Added functionality provided by two new codes: 00 displays the MDM firmware revision level (without a decimal point); F1 initiates the MDM test sequence described in 6.3.4.

CHecK

Described in 3.1.2; added functionality includes: choice of checksum algorithms on the MPP-80S (like that an the EPP-80); new CHecK operations 5 and 6.

CHecK 5 is an access-time test of a programmed device. Device data outputs are strobed at 50 nsec intervals after device chip-enable input is activated to determine when the device output data becomes valid. The resultant display shows the lowest value (750 nsec maximum) for which all tested locations present valid data. You select the supply voltage ("VCC") in the range 4.0 V to 5.9 V by entering two digits on the hex keypad, then press "ENTER"; the decimal point in the voltage value need not be entered. The voltage you set is used for all CHecK 5 and 6 operations until you reset it. You do not need to load a valid data pattern into RAM before executing CHecK 5.

CHecK 6 is a real-time verify of chip contents; it does require valid data pattern in user RAM against which to compare the PROM data. The output-strobing circuitry associated with CHecK 5 is used; the same voltage selection process is required. You are also requested to set an upper limit for the access time; default limit is 750 nsec. To change the access time limit, press the " - " key to decrement the time by 50 nsec; the limit can be as small as 50 nsec. The time limit you set is used for subsequent CHecK 6 operations until you reset it. CHecK 6 provides pass and fail indications like those for CHecK 1.

Note that CHecK 5 and 6 operations are valid only for PROM and EPROM devices, not with single-chip microcomputers. The message "NOT POS" is displayed if these checks are inappropriately attempted.

6.3.4 TEST AND CALIBRATION

You should test the MDM module every six months or whenever programming yield falls below chip manufacturer's specifications. Test software is included in the MDM module. You will need a digital voltmeter (Kontron 4021 or equivalent) and a dual trace oscilloscope (Hitachi V 1050 or equivalent) for the teste. Relevant schematic diagrams are included at the end of this document.

Complete testing of the unit includes the following:

Measurement of performance (voltage) at SA-MDM eins

DC calibration of MDM as required

Tests of the access-time logic

Verification of the programming sequence

The test and calibration steps are described in following paragraphs. These procedures must be performed in the order they are presented here to ensure that the MDM operates to specification.

Power Supply - Performance Check

Install the MDM module and SA-MDM socket adapter on the EPP-80 or MPP-80S PROM programmer, and apply power to the processor. To begin the, performance check operation, press the FC key (on the EPP-80) or -the-hex **Key on the MPP-80S.** When "FC 00" appears on the programmer display, select the Pin Test Function by pressing "F1" and "ENTER".

The display shows "PINTEST 00". Initiate the first test for Pin 20 on the socket adapter by pressing hex data keys "10" (the Pin Test Code) and press "ENTER". The display now shows "VCC 00"; you enter the VCC code "FF" and measure voltage at Pin 20. The voltage should fall within the range in Table 6-2 (first line). Proceed by repeating the above sequence, using all the Pin Test Codes and VCC Codes in the table, in the order listed.

Table 6-2. Power Supply Tests for MDM

Pin Test	VCC Cod	Pin Numbe	Voltage
10	FF	20	25.6 +- 0.2
10	7F	20	12.8 +- 0.2
20	FF	20	25.6 +- 0.2
20	7F	20	12.8 +- 0.2
30	FA	20	40.0 +- 0.5
30	7F	20	20.5 +- 0.5
45	FF	27	12.8 +- 0.1
55	FF	27	10.0 +- 0.2

If the module passes each of the tests summarized in the table, it is properly calibrated. If one or more of the voltages are not in the listed range, the unit must be recalibrated. To recalibrate the MDM, dismantle the unit to access the printed circuit board, and make adjustments as described in the following paragraphs.

Calibration of MDM:

To calibrate the MDM personality module, you press the same sequences of keys as for the Performance Check operation, but measure voltages of some additional test points an the printed circuit board and modify certain alignment points (P1 through P 7). Table 6-3 reveals the testpoints, voltages, and alignment points; these must be processed in the listed order. Calibration should be verified by measuring voltages denoted "(Check)".

Table 6-3. Voltage Alignment

Pin Code	VC Code	Testpoint	Voltage	Alignment
10	FF	T54 case	30.0 +- 0.5	P 4
10	FF	T68 collector	3.4 +- 0.025	P 6
10	FF	Pin 20 SA-MDM	25.6 +- 0.2	P 1
10	7F	Pin 20	12.8 +- 0.2	(Check)
20	FF	Pin 20	25.6 +- 0.2	P 2
20	7F	Pin 20	12.8 +-0.2	(Check)
30	FA	C 13 minus	46.5 +- 1.5	(Check)
30	FA	Pin 20	40.0 +- 0.5	P 5
30	7F	Pin 20	20.5 +- 0.3	(Check)
45	FF	Pin 26	12.8 +- 0.1	P 3
55	80	Pin 26	10.0 +- 0.2	P 7

After the calibration sequence summarized in Table 6-3 is performed, voltage alignment for the MDM personality module is complete. If module performance during PROgram or CHecK operations has been suspect, you should also check all internal pin switches. Checking of internal pin switches requires use of the Pin Codes and testpoints presented in Table 6-4. By altering the VCC Code to "7F", you can recheck the testpoints, expecting one-half the voltage listed; note that voltage tolerance is also one-half of that listed, except that Pin Code 46 should result in 6.40 V +- 0.1.

Table 6-4. Internal Pin Switch Tests

Pin Test Code	VCC Code	Testpoint Pin Number	Voltage
11,21	FF	21	25.6 +- 0.2
12,22	FF	22	25.6 +- 0.2
13,23	FF	23	25.6 +- 0.2
14,24	FF	1	25.6 +- 0.2
17,27	FF	2	25.6 +- 0.2
46	FF	27	12.8 +- 0.1

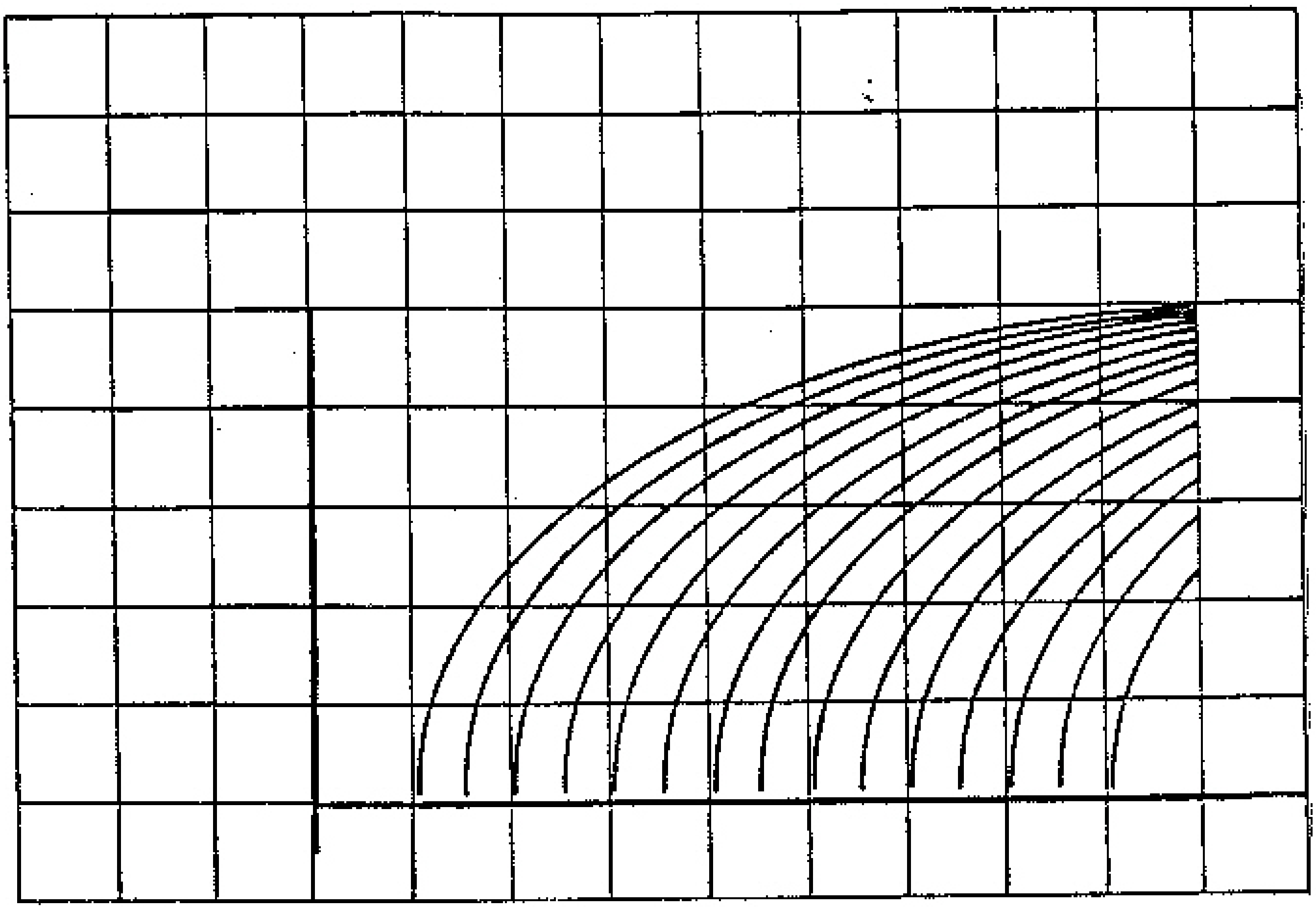
The MDM personality module includes an Overcurrent Protection Circuit that removes supply voltage whenever excessive supply current is drawn by s device in the programming socket; this results in the display "E CUR". Such a condition may indicate improper insertion of the device, or a device-internal VCC short circuit. To test the Overcurrent Protection Circuit, connect the collector of T65 (see schematics for location) to ground while any CHecK operation is in progress. To verify internal power sources individually, apply a 270 Ohm/0.5 W resistor between pins 14 and 20 as well as a 33 Ohm/0.5 W resistor between pins 14 and 28 of the socket adapter. Execute the following Test Codes to intentionally create the "E CUR" error message:

Test Code	VCC Code
10,20	FF
45	D5

Access-Time Logic Tests

The built-in test of access-time logic is run when you choose Special Function code "E6" (i.e. "FC" or "8" followed by "E6"). Measure power at pins 20, 21, 22, 23, 27, 1, and 2; compare the resulting wave forms with those depicted in Figure 6-2.

Figure 6-2. Access-Time Logic Wave Forms



Testing the Programming Sequence

This test provides the means to check all programming parameters during real-time execution of the PROgram Operation without halting for an error condition. lt is recommended that this test should be conducted only by Kontron personnel.

Select the Special Function code "Ei" to inhibit all error reporting, then observe PROM programmer Operation via an oscilloscope. Note that you must select Special Function code "EO" to reset the error-intercept capabilities (or power the processor down and back up) before resuming

6.4 PROGRAMNABLE ARRAY LOGIC MODULE

Programmable array logic (PAL) chips having 20 or 24 pins can be programmed an the MPP-80S or EPP-80 PROM programmers using personality module MOD 21 and associated socket adapters. The module includes many blank fuse maps in its firmware, and special functions suited to these chips. Consult your Kuntron Programmers Comparison Chart for supported chips and corresponding socket adapter models.

The PAL module provides the following special features:

- one socket adapter serves all 20-pin PAL typen
- 24-pin PAL devices supported with another socket adapter
- blank chips can be simulated in programmer RAM
- PAL Assembler and "fuse pattern" formats supported over serial port
- inntelligent checksum for trustworthy verification
- built-in module test routines, including selectable voltage levels
- blank check and illegal bit tests
- read-protect ("last fuse") capability
- programs PAL devices from MMI, National, and TI

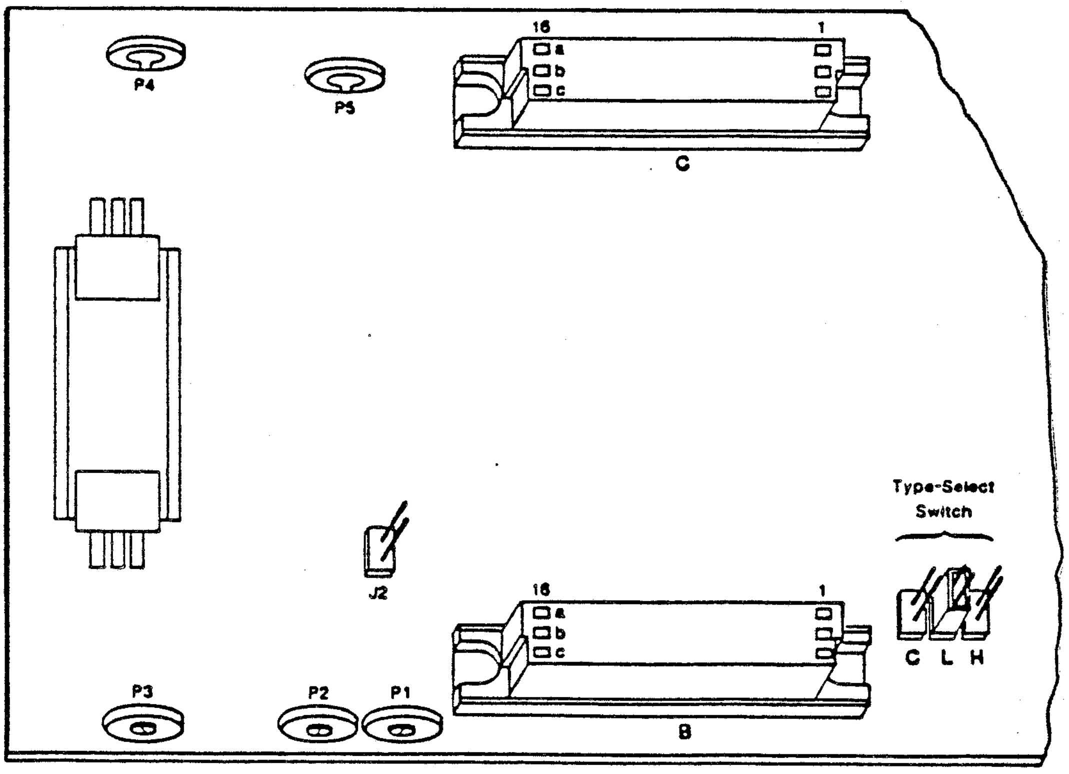
Basic PROM programmer operations (e.g., LOAD, PROgram) described in Chapter 3 apply also to PAL programming. However, actual data must follow the formats prescribed by chip manufacturers.

6.4.1 Switch Settings

Before a socket adapter is mounted an the PAL personlity module, the "type Select" slide switch must be set to indicate the kind of programmable logic to be used. The Type Sclect switch is a 3—position switch whose location is shown in Figure 6-3. PAL logic classifications and corresponding Select switch settings are as follows:

Type of Programmable Logic	Switch Setting
VAR (wriable)	C -- for chip 16 C1 only
VOL (voltage open low)	L -- for all VOL chips
VOH (voltage open high)	H -- for all VOH chips

Figure 6-3. PAL Board Layout



If you wish to be able to read—protect devices after programming them, the jumper J2 must be shorted out. For more information about this special function, see 6.4.3.

6.4.2 BLANK CHECK

PAL devices can only be programmed once; a blank check should be routinely performed on each chip. Because PAL devices contain circuits to be modified during programming, blank check operations utilize a blank fuse map of the device. You specify the fuse pattern for your PAL device to the PROM programmer via one of the following codes:

Code	Fuse Pattern
10	10 H 8, 10 L8
11	12 H 6
12	14 H 4
13	16 H 2
14	16 C 1
15	12 L 6
16	14 L 4 .
17	16 L 2, 16 L 8, 16 R 8, 16 R 6, 16 R4

For PAL devices corresponding to code 17 only, Check 00 (blank check) can be performed directly. For other devices, you must load a blank map into RAM and compare the RAM with the device. To load the blank fuse map for your device into RAM, you press the "#A" key (on the EPP-80) or the hex "A" key (on the MPP-80s), followed by the 2-digit code number. Then you can perform Check 01 to compare RAM with the device, as described in 3.1.2.

BLANK FUSE MAP LOADING Example: Load the fuse map for a code 10 PAL device (10 H 8 or 10 L 8) into programmer RAM.

Press	The display is	and it means
	Fc PAL	Processor is in standby mode and equipped with Mod 21.
A	Fc PAL 00	Special function invoked.
1 0	Fc PAL 10	Code 10 specified.
ENTER	--------------	RAM loading in progress.
	Fc PAL	Loading complete; Check 01 operation can now proceed.

6.4.3 PROGRAM VERIFY PROTECT

The PAL devices include two fuses which may be "blown" to prevent further verification. The option of blowing these fuses is automatically presented at the end of each programming operation, if the jumper J2 has been shorted out. (See Figure 6-2 for the location of jumper J2).

You perform the steps outlined in 3.2.3 to program a PAL device. If the Verify Protect option is active, the PROM programmer display shows "FUSS" when all programming is complete. You can then press the PROgram key to read-protect the chip, or any other key (e.g., "   ") to complete the operation without blowing the last fuse.

VERIFY PROTECT Example: Read-protect a PAL chip after programming.

Press	The display is	and it means
	Fc PAL	Programmer ready.
PROgram	Pr SA 0000	Select Starting Address.
ENTER	Pr EA 01FF	Select Ending Address.
ENTER	Pr Lo 01FF	Data in Low Nibble?.
ENTER	Pr P -I	Yes, Low Nibble (press " - " for High Nibble); seat device in socket.
ENTER	-------------	Programming in process.
	FUSE	Programming complete; ready to blow last fuse.
PROgram	Pr P 1-	Fuse blown, device can be removed; press any other key than PRO to omit blowing the last fuse.
PROgram	Pr SA 000	To program the next device, repeat the sequence; or press ENTER to return to standby (ready) mode.

6.4.4 SPECIAL INPUT/OUTPUT MODES

Special I/O functions for the MPP-80S and EPP-80 PROM programmers are provided by the PAL personality module. To utilize these functions, you press the "fiA" key (on the EPP-80) or the hex "A" or "D" key (on the MPP-80S) when the programmer is in standby mode (i.e., when "Fc PAL" or "READY FOR PAL" is displayed); you may then press digits corresponding to one of the following function codes:

Code	I/O Function
00	Output RAM data via the serial interface in PAL Assembler formet (hex dump).
01	Input data via the serial interface in PAL Assembler hex formet.
02	Output RAM data via serial interface in "fuse pattern" formet.

When the desired code is shown in the programmer display, press ENTER to begin data transmission or reception. For more information about these I/O formats, see the manufacturer's PAL device documentation.

6.4.5 MODULE TEST FUNCTION

You can run many tests on the PAL personality module to verify that it is functioning properly. There should be no chip in the socket when tests are run; if a chip is in the socket during a test, it. will be programmed (destroyed). Tests are accessed via the special function key ("PA" an the EPP-80 or hex "A" on the MPP-80S) with special function code 20. Six different tests can be run, as follows:

Test Code	Test
1	Vcc at 11.5 V -- probe an C14a,c -- adjust P2
2	Vcc at 6.0 V -- probe an C14a,c -- adjust P3
3	Vcc at 4.5 V -- probe on C14a,c -- adjust P1 21.0 V supply -- probe an C13a,c -- adjust P4 11.75 V supply -- probe on C15a,c -- adjust P5
40	Addressing circuits exercised
60	Read multiplexer test
80	Programming routine runs continuously

Test codes 1, 2, and 3 are referenced to ground at C16a,c. If adjustments are made, each of these tests should be rerun in case other settings are affected. You can interrupt the programming check (code 80) by pressing any key. Appropriate messages are displayed if the module fails any of the tests40, 60, or 80.

MODULE VOLTAGE TEST Example: Execute test at 11.5 V (Test Code 1).

Press	The display is	and it means
	Fc PAL	Programmer ready.
A	Fc PAL 00	Special function invoked.
2 0	Fc PAL 20	Code 20 (Module test) specified.
ENTER	Pr 00	Test program code can be entered.
1	Pr 01	Test code 1 selected.
ENTER	Pr 01	Test is executed.

6.5 INTEGRATED FUSED LOGIC MODULE

You can use the Integrated Fused Logic (IFL) personality module with associated identifiers to program all supported Signetics 20- and 28-pin integrated fused logic devices. Check the Kontron Programmers Comparison Chart to determine the appropriate identifier for your device type.

There are four DIP switches in the IFL module which are accessible through the opening for the identifier at the back right of the unit. These control the following module functions:

Switch	Module Function When On
1	Power-on CRC check, software check, and RAM check enable
2	Ring bell an all keys enable
3	Ring bell only for valid data enable
4	Display entered data for 200-300 ms enable

The IFL identifier can be changed whenever the programmer is in standby mode. When you subsequently press the "ENTER" key, the new device type is included in° the standby display and RAM memory is automatically erased to a virgin state appropriate to the new device. If such automatic erasure is not possible, all PROM programmer operations are suspended until memory is manually erased (with the "F" key, see 6.5.1).

Supported IFL device types include the following:

FPLA: Field Programmable Logic Array -- this family of devices includes a programmable .AND array (corresponding to product terms) and a prograt:nable OR array (corresponding to sum terms). Each OR term controls an output function which can be programmed for "true active high" or for "true active low^n. The true state of the output function is activated by logical combinations of the input variables (and/or their complements) which are expressed as sums of products.

FPLS: Field Programmable Logic Sequencer this family of devices consists of a programmable NAND array to control output functions that can be programmed to "true active high" or "true active low".

FPGA: Field Programmable Gate Array -- this family of devices includes both a programmable AND array (product terms) and a programmable feedback OR array (sum terms). The on-chip State and Output registers consist of Qp and Qf edge-triggered S/R flip-flops with asynchronous preset option. The internal input variable "C" is used to generate true and complemented transition terms.

FPRP : Field Programmable ROM Patch

Programmable Gate Arrays.The device consists of 48 8-bit word which are addressed by a 16-bit programnable address comparator.

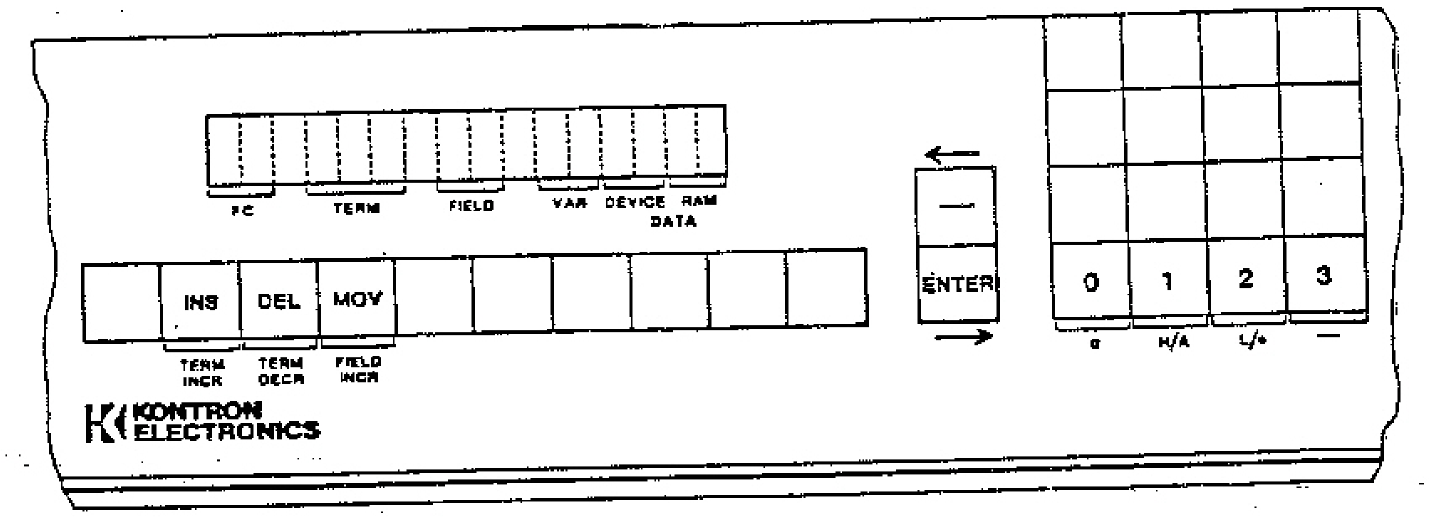
The IFL module attaches to sockets on the front panel of the MPP-80S personality modules. An identifier suited to the IFL device type must be inserted in the module before power is applied to the programmer.the IFL module performs an automatic self-check during the initialisation seuence. An LED is lit on the module, next to the proper socket for the selected device.

When initialization is complete, the programmer displays the device type to indicate that it is in standby mode; for example, the display might be:

FPLS 82 S 104

Data for IFL devices is organized into terms, each of which may contain several fields; each field contains one or more data variables. Data formats, fields and terms for IFL devices are described in manufacturer's manuals, such as the Signetics Bipolar and MOS Memory Data Manual. The set of pressure sensitive labels supplied with the IFL module should be affixed to the MPP-80S front panel to indicate IFL-specifid variable values as well as funetions associated with programmer keys and display fields; cut apart the labels and place them as shown in Figure 6-4.

Figure 6-4. IFL Label Locations



6.5.1 LOCAL IFL OPERATION

When you use the PROM programmer IFL module, characteristics of IFL devices and data necessitate some modifications in programmer operations as they are described elsewhere in this manual. IFL RAM data is organized into numbered terms, each cGnsisting of a number of predefined fields; fields contain numbered variables, where the numbers proceed from high (at the left) to low (at the right). The number and size of the fields is device-dependent. Newly affixed labels beneath the hex data keypad denote the permitted data values: 0, H, L, - (not the " - " key above ENTER), A (not the hex key "A"), and ".".

RAM data is identified by term number (not by a hex address), field (e.g. Input), and variable number within the field. You can use the SET function to enter, inspect, and/or modify field values. When you first press the SET key, the programmer display indicates the first available term, field, and default value; for example, "SET SE TOO C 0". Here term TOO, field C and value 0 are indicated; the field has only one variable, so no variable number is needed. You can enter any 2-digit term number to access that term; if the term number you specify is beyond the range for the selected device, the last valid term is assumed. You can also enter hex A to inspect the P/E term (FPLS devices) or AL term (FPLA or FPRP devices).

In this SET mode, the following specially labeled keys enable you to move within and between terms:

Key	Function
TERN INCR	Proceed to first variable (e.g., first Input value) in the next sequential term; for example -- if term T12 is currently displayed, the term T13 is accessed with this key.
TERN DECR	Proceed to first variable in the last sequential term; for example -- if term T21 is currently displayed, the term T20 is accessed with this key.
FIELD INCR	Proceed to first variable of the next field in the term currently displayed. If you are currently valuing the last field of a term, this instruction "wraps around" to the first field of that term.

You can change tue displayed data value by pressing any of the aforementioned specially labeled data keys (0, H, L, A, .). Not all of these are valid for a particular field and a particular device; only valid data values are accepted by the programmer. New values momentarily show in the display if DIP swtich 4 was set; the next variable in the field is then displayed. After you have completely valued a data field, the programmer automatically performs the FIELD INCR function.

When you are in the SET mode, press "ENTER" to advance to the next variable without changing the current one. Similarly, you can press " " (i.e., minus -- the key above ENTER) to return to the previous variable without changing the current one. Press the hex "E" key at any time to exit the SET mode.

When you use the IFL module, most other PROM programmer keys perform expected functions, as documented in Chapter 3. The following list summarizes relevant functions and refers you to detailed information:

Key	Function	See	Description
F	Fill (Erase)	3.4.1	Erase RAM memory to a virgin state for the selected device. The display shows "ErASE"; the Operation begins when you press "ENTER".
IN	Input	3.3.1	Input data via RS 232 port in Signetics paper tape format.
OUT	Output	3.3.2	Output RAM data via RS 232 port in Signetics paper tape formt; carriage returns/linefeeds are included for display and printing.
PRO	Program	3.1.3	Program IFL device.
CHK	Checks 0,1,2	3.1.2	Check 0 is blank check; Check 1 compares RAM with device; Check 2 is the illegal¯ bit check. Select Check number, then press "ENTER".
LOAD	Load	3.1.1	Load RAM with data from device currently in the IFL module socket.
REM	Remote operation	6.5.2	Disable all keys except "REN"; control PROM programmer from RS 232

PROM programmer error messages associated with any operation are presented in IFL data format, as indicated by the labels you have affixed below the display window. For example, "E2 T13 I 11 H 0" indicates an error during Check 2, in term 13, the Input field, variable 11: the device value is H, which cannot be programmed to the RAM value 0. When such an error is displayed, you can press "ENTER" to advance to the next error (if any), or to the standby display (if there are no more errors). Press the hex key "E" to return to standby mode unconditionally when a Check error message is displayed.

Operational examples for each of the IFL device types follow.

FPLS Example: Enter values for the 82 S 104 device into programmer RAM.

Press	The display is	and it means . . .
ON	FPLS 82 S 104	Power-on initialization complete; the audio tone sounds until you press " ENTER[11], RAM is automatically set for the virgin state of the indicated device,
SET	SE P-E H	Current (default) value for P/E term is displayed. This can be set to H or L.
L	SE P-E L	Change value to L; newly entered value shows for a brief time if DIP switch 4 is CLOSED.
	SE T00 C 0	Programmer automatically increments to first field (C) of the next term (T00); current value is zero.
-	SE T00 C	The value - (IFL-labeled key) is entered; 0, A, , are also possible.
	SE T00 I 15 0	Programmer auto-increments to first variable (15) of the next field.
H	SE T00 I 15 H	The value H is entered for variable 15 in the Input field,
	SE T00 I 14 0	and the next variable is displayed.
	….	
	….	
	….	
L	SE T000 I 0 L	Last Input variable is valued,
	SE T00 P 5 0	and programmer auto-increments to next field of TOO (Present state).
	….	
	….	
L	SE T00 F 0 0	Last variable of the last field in term 00 is valued,
	SE T01 C 0	and programmer auto-increments to first variable of the next term,
H	SE T47 F 0 H	Last variable of last term is valued.

The FPLS devices include special "test terms" 48 and 49. The following notes indicate how there are used during PROM programmer IFL operations:

CHecK operations	Check 1 causes all of terms 0-47 to be checked, but only C (complement array) and NF variables are checked in term 48 and only the NF variables are checked in term 49. Checks 0 and 2 cause only terms 0-47 to be checked.
LOAD operations	All of terms 0-47 are copied from the FPLS to RAM.
INput operation	First erases RAM to the device—dependent virgin state, then reads in terms 0-47 and fills terms 48 and 49 with the DELETE test pattern.
OUTput Operation	Writes only terms 0-47.
ERASE operation	Erases RAM to the device—dependent virgin state and fills terms 48 and 49 with the VIRGIN test• pattern.
PROgram operation	When the FPLS device is first programmed, terms 48 and 49 are filled with the DELETE test pattern. The programming sequence is

1. Check to see if there is anything to program; if not, exit.
2. Program the test array in terms 48 and 49 first.
3. Perform CHecK 1 for the test array.
4. Perform CHecK 2 (programmability) for terms 0-47.
5. Program the device from RAM.
6. Perform CHecK 1 an all terms 0-49.

FPLA Example: Program the 82 S 100 device.

Press	The display is	and it means
ON	FPLA 82 S 100	Power-on initialization complete.
ENTER	FPLA 82 S 100	Silences automatic audio tone.
SET	SE AL 7 H	Access AL (Active Level) term; choose H or L value,
H	SE AL 6 H	Programmer auto-increments through AL values,
	…	
	…	
	SE P00 I 15 0	First Input field variable of tern P00 is accessed; allowed values include 0, H, L, - . Following each entered value, the programmer advances to the next variable.
	…	
	…	
L	SE P00 10 0	Last Input variable for P00 is valued.
H	SE P00 F 7 A	and programmer auto-increments to first variable of output Function field; allowed values are A and "."
	…	
	…	
A	SE P01 I 15 0	When the last variable of P00 is valued, programmer auto-increments to first variable of P01.
	…	
	…	
	SE P47 F 0 A	After the last variable of the last term is valued, programmer auto-increments to P00 (not AL).
E	FPLA 82 S 100	Pressing the hex "E" key ends the SET mode; to program the device, press "PRO".

FPGA Example: Enter values for the 82 S 102 device.

Press	The display is	and it means
ON	FPGA 82 S 102	Power-on initialization complete.
ENTER	FPGA 82 S 102	Silences automatic audio tone,
SET	SE F00 AL L	Access AL (Active Level) field of the first term; AL is not a separate term.
H	SE F00 I 15 0	Programmer auto-increments to next field (Input variables); allowed values are 0, H, L, or -. Press "ENTER" to skip to the next variable, or "TERM INCR", "TERM DECR", or "FIELD INCR" to display another variable without entering a new value.
	…	
	…	
	SE F00 I 0 0	Input variables are valued; the programmer steps through them,
L	SE F01 AL L	and increments to the next F-term.
	…	
	…	
	SE F08 I 0 0	After all variables in all terms are valued, the programmer steps back to the AL term of FOO.

FPRP Example: Enter values for 82 S 106 device.

Press	The display is	and it means
ON	FPRP 82 S 106	Power-on initialization complete.
ENTER	FPRP 82 S 106	Silences automatic audio tone.
SET	SE AL H	Access AL (Active Level) term; values can be viewed (with ENTER and " - ") but not changed in this device. In this situation only, pressing any yellow key advances to P00; pressing "TERM INCR" advances to P01.
	SE P00 I 15 L	First Input variable is displayed; allowable values are 0, H, L, -.
H	SE P00 I 14 L	Programmer auto-increments to next variable as new value is entered.
	…	
	…	
	SE P00 I 0 L	After the last variable in Input field is valued,
	SE P00 F 7 A	programmer advances to Output Function field (F); allowed values are A and "."
	…	
	…	
	SE P47 F 0 A	After term P47 is valued, programmer jumps to the start of term P00.

6.5.2 REMOTE IFL OPERATION

You can operate the PROM programmer from a remote terminal or computer via the RS 232 interface. Before you press the REMote key, you should verify that the communication baud rate for the programmer is that used by the remote device; see 3.4.3 for procedures to change the PROM programmer baud rate. Wien you press the REMote key on the programmer, commands are expected at the communications port; only the REMote key itself is active on the programmer
keyboard you can press that key again to return the PROM programmer to local control.

In remote operating mode the carriage return character (<CR>) serves as a positive acknowledgement character, much like the "ENTER" key for local operations. Commands entered at the remote device are not actually executed unless they are immediately followed by <CR>. A command line can therefore be modified with screen-device editing facilities (e.g., cursor movement keys) before the command is executed by the programmer.

As soon as you press the REMote key to put the programmer into remote operating mode, the programmer sends a device identification line and a complete remote command menu to the remote device. The prompt character "." appears at the end of the menu. The terminal display might be as follows:

```
                    FPLA 82S100/101



        B         BLANK CHECK
        V         VERIFY CHECK
        I         ILLEGAL BIT CHECK
        P         PROGRAM
        F         ERASE RAM
        T         TAPE INPUT
        O         TAPE OUTPUT
        D         DISPLAY TERM(S)
        L         LOAD
        C         COMPRESSED DISPLAY
        Q         QUIT
        ->        EDIT
```

"D", "C", "Q", and "-->" represent functions unique to remote operation. Except for the remote editor (6.5.3), each of these functions is described in detail in an indented paragraph later in this subsection. The other remote commands are similar to those used for local operation of the IFL module; the following list refers you to detailed descriptions of these similar commands and indicates special considerations for remote use.

Command	See	Notes for remote IFL operation
B IV	3.1.2	These correspond to Check 0, 1, and 2, respectively. The remote display shows "P<" to indicate that the IFL device should be loaded, and "P>" to indicate that it should be removed. Press <CR> to proceed to the next step. Error displays are identical to those on the programmer itself, and are accompanied by a BEL character (‹Control>-C). To view the next error, press <CR>; to return to remote command menu, press "0".
L	3.1.1	The remote display shows "P<" to indicate that the IFL device should be loaded, and "P>" to indicate that it should be removed. Press <CR> to proceed to each next step.
P	3.1.3	The remote display shows "P<" to indicate that the IFL device should be loaded, and "P>" to indicate that it should be removed. Press <CR> to proceed to each next step. Automatic checks may result in errors: "E2" indicates an error during illegal bit check before programming; "EP" indicates an error during programming; "El" indicates a verification error (device-RAM mismatch) after programming
F	3.4.1	The reminder "ERASE" is displayed on the screen when you press "F". Press <CR> to begin; the message "COMPLETE" appears when RAM is reset to the virgin state for the selected device, then the-remote-command menu reappears.
T	3. 3.1	The PROM programmer expects input over the RS 232 communication line in Signetics paper tape format. When that input is terminated, the remote command menu reappears. If an input error is detected, the message "E In" is displayed.
0	3.3.2	The PROM prograammer sends complete contents of the IFL data in RAM via the RS 232 interface in Signetics paper tape format. When the transmission is complete the remote command menu is displayed.

DISPLAY TERMS - D

You can use the remote Display function to examine individual terms of IFL data as they exist in PROM programmer RAM. When you enter "D" followed by <CR>, the prompt "TERM:" is displayed. You enter a range of term numbers in response; the range consists of a beginning term number and an ending term number, separated by the slash character "/". For example, you can enter "12/24" to view terms twelve through twenty-four. The Symbol "A" denotes the Active Level term, which is assumed to precede term 0. If you enter an ending term number that is less than or equal to the beginning term number, only the beginning term is displayed. In the resulting display, each term occupies one line. Fields within each term are separated by a space and each field is preceded by "*x", where "x" is the identifying letter (e.g., "I" for the Input variables field).

COMPRESSED DISPLAY - C

You can use the Compressed Display function to list (view) an individual term of IFL data on the screen display of your remote terminal. When you enter "C" followed by <CR>, the prompt "SELECT:" appears on the screen. You should enter a term number after this prompt; if the number consists of only one digit it must be followed by a comma ",". Then you enter <CR> again, and the current values for all variables in the term are displayed. If you specify a term number that is outside the predefined range for the selected device, the last term is shown.

QUIT Q

The remote Quit command is similar to the hex "E" key in local IFL operation: you can press "Q" to return to the remote command menu after viewing a Check operation error message, and you can also press "Q" when the menu is displayed to exit from remote operating mode. The PROM programmer returns to local standby mode when you terminate remote operations.

See the next subsection for a description of the IFL Remote Editor.

6.5.3 IFL REMOTE EDITOR

When the PROM programmer equipped with the IFL personality module is in remote control (as described in 6.5.2), you can press the "-->" (i.e., cursor right) key to invoke the remote editor. If your terminal does not have such a key, press any of the double-key entries <Control>-F, <Controp-L, or <Control>-P to invoke the editor. If a control sequence invokes the editor, that special character is echoed by the programmer instead of the cursor movement key during the editing session.

Displays and IFL data in the following discussion assume that an FPLA device is to be programmed. Operational differences for other IFL devices are described at the end of this subsection.

An IFL editor command menu is displayed an your terminal screen when the editor begins operation. Commands are similar to Chose described in 4.3, but are customized for IFL devices. The commands are as follows:

Menu		Command Description
U	DEC (UP)	Display previous term; from 00, show AL
T	SET TERM #	Let me specify next term number
N	NEXT FIELD	Advance cursor to next field in this term
C	CLEAR TERM	Change variables in term to virgin state
D	DELETE TERM	Set variable values so term is ineffective
Q	QUIT EDIT	Terminate editor, return to remote mode
-->	CURSOR RIGHT	Move cursor to next variable
<--	CURSOR LEFT	Move cursor to previous variable
(CR)	NEXT TERM	Display next term

The menu is followed by a display of the Active Level (AL) term, as follows:

```
     76543210
   *A HHHHHHHH
```

The screen cursor overlays the first AL variable (number 7). You can press the "L" key to change the value of this variable, or cursor movement keys "-->" or "<--" to move the cursor within the term. The cursor cannot be moved out of the term. Only the data values "H" and "L" and valid editor commands are accepted; pressing other keys produces no response. Data values entered are immediately reflected in the screen display and transferred to the PROM programmer RAM.

After you inspect and/or modify the AL term, you will typically enter <CR> to view the next term of IFL data, or "T" to specify a term number. The AL values remain on the screen, and the display might be as follows:

```
                                     76543210

                                     *A HHLLHHHH
                 111111
                 5432109876543210   76543210
        *P 00 *I 0000000000000000 *F AAAAAAAA
```

Note that field identifiers are preceded by the asterisk character (e.g., "*F"), and current values of the variables are displayed. The cursor is positioned over the first variable in the first field of the term (hexe, 115) and you can change values or move the cursor within the term, just as you did for the AL term. When you press <CR>, the next sequential term (P01 for the display shown) replaces the current term. Similarly, the "U" command replaces the current term in the display with the previous term.

When you press "T", the first part of a term display is presented as a prompt ("*P 00) and the cursor is placed over the leftmost zero. You can now enter any valid two-digits to specify the next term for display, or another editor command. If you specify a term number, that term is displayed and ready for editing.

The "N" (Next field) command advances the cursor to the beginning of the next field. For example, if the cursor is positioned over 112 in the preceding display, it advances to F7 when you press "N". If you again press "N" (when the cursor is in the F field) the cursor jumps "around" to the I field again; that is, the "N" command only moves the cursor within a term.

The "C" command changes all variables in the current term to their virgin state, which is device-dependent. This command is effective from any cursor position in any term, even the AL term.

The "D" command changes all variables in the current term to values that cause the term to have no net effect on the output states of the device. This command is useful to eliminate a misprogrammed term, as when it must be replaced by another term. For the FPLA and FPRP devices, the "D" command changes all F field variables to "."; for the FPLS devices, the command changes all variables in the C, N, and F fields to "-"; the command does nothing for FPGA devices.

To terminate IFL editor execution and return to IFL remote operating mode, enter "Q" at any time. The remote command menu (see 6.5.2) and prompt then appear on the terminal screen.

Notes for editing other IFL devices:

- For the FPLS, terms T48 and T49 are available only for viewing, not for editing.
- For FPRP devices, the AL term is displayed when the editor begins execution; but these variables cannot be changed, so the editor automatically skips to term 00.
- For FPGA devices, AL variables are integrated into the terms; each *G term includes one *A variable.

6.6 IFL BOOLEAN OPTION

The IFL personality module may be equipped with a Boolean option. This option consists of a special firmware set (included in the module itself) that permits you to

I. create and edit a source file that contains equations describing output pins of the selected IFL device in terms of input and control variables, and

2. assemble the equations into fuse map format in PROM programmer RAM, ready for programming the device.

Use of the IFL module with Boolean option requires a "dumb" terminal connected to your EPP-80 or MPP-SOS PROM programmer via the RS 232 C port; on the EPP-80 this must be Port A. Boolean firmware assumes that communications proceed at 9600 baud with no parity and one stop bit, irrespective of the settings of PROM programmer DIP switches.

To use the IFL Boolean module, you should initialize the IFL module according to the procedures in 6.5. Then set the PROM programmer baud rate to match that for your terminal; 9600 is the recommended baud rate. Now you can press the "A" or "D" key on the programmer data keypad to invoke the Boolean option.

For detailed information about the use of the Boolean module with various IFL devices and specific programming suggestions, see 6.6.6; any IFL Boolean user should examine that information before preparing a source file and programming a device.

- The message "Initializing" is displayed on the terminal screen while the IFL Boolean intialiization sequence executes. Upon completion of the intialization sequence, the following Boolean command menu appears:

```
A Assemble
I   Initialize
P   Print
T   TTY Editor
--> LINE Editor ESC
Quit

Selection:
```

You can now enter any of the listed commands after the prompt "Selection:". Typically, you will first use one of the editors; press "-->" (i.e., cursor-right) to use the LINE Editor on a terminal with a screen, or press "T" to use the TTY Editor if yours is a hard-copy TTY terminal. Use of the editors is described in 6.6.3. Later you will want to assemble your source file; use of the Boolean assembler is described in 6.6.4. For I/O operations (e.g., to save your source file on paper tape) you can use the "P" (Print) command described in 6.6.5.

You can enter "I" after the prompt "Selection:" to reinitialize the Boolean module whenever the Boolean command menu is displayed. Note that reinitialization changes all pin labels back to their defaults, and clears the symbol table. Before reinitialization actually begins, you are prompted:

"Dc You Wish To Reinitialize? (Y/N)"

Enter "Y" to reinitialize or "N" to receive the "Selection:" prompt again.

You can enter <ESC> whenever the Boolean command menu is displayed to terminate module execution. The message "PRESS ENTER" appears on the PROM programmer display panel. After you press the "ENTER" key, any PROM programmer function (e.g., PROgram, Input, etc.) can be performed from the programmer front panel. Basic PROM programmer functions are described in Chapter 3. The Boolean source file or assembled data in programmer RAM are not disturbed unless you explicitly modify RAM; press "ENTER" twice to return control to the IFL module.

6.6.1 BOOLEAN SOURCE FILES

A Boolean source file is organized into four blocks, which together can include as many as 98 lines. The first five lines constitute the Heading block. You supply the textual information for these lines to identify the source file as follows:

Line	Contents	Maximum Size
01	company name	40 characters
02	your name	40 characters
03	date	8 characters
04	revision	3 characters
05	I.D. code	15 characters

The second block of a Boolean source file is the Pin List block. This is generated automatically by the IF1 Boolean module; it lists pin numbers, function codes, and default labels appropriate to the selected IFL device. Control variables are listed as "virtual pins" with the character "x" preceding the pin number. Information in the Pin List block is used to produce a symbol table for the assembly. You may change (edit) only the labels in this block; new labels can include any symbols that are not predefined (see 6.6.2).

The third block of a Boolean source file is the Equation block. Here you enter equations, each of which defines an output pin (i.e., its label -- complemented or not) to be the result of logical operations on other pins (labels). Boolean equations are described in detail in 6.6.2.

The fourth block of a Boolean source file is the Description block, in which you may enter any descriptive text and comments. The Description block is not processed by the Boolean assembler.

6.6.2 IPI BOOLEAN EQUATIONS

Equations you enter in the Equation block of a Boolean source file may include only the following symbols:

any pin Label as defined in the Pin List block	
+	OR operator
*	AND operator
/	complement operator
:	flip-flop designator
•	equation separator
,	output label separator

Equations must take one of the following forms:

```
1.  symbol "=" expression
2.  symbol ":" "S" "=" expression  (for S/R flip-flop)
    symbol ":" "R" "=" expression
3.  symbol ":" "J" "=" expression  (for J/K flip-flop)
    symbol ":" "K" "=" expression
4.  symbol ":" "J" "=" expression  (for D flip-flop)
    symbol ":" "K" "="  "/" "J"
```

<u>symbol</u> is any pin label, complemented or not
<u>expression</u> is any logical sum (+) of logical products (*)
characters enclosed by quotes (" ") are literal symbols which must appear as shown.

The equation separator "." serves to separate equations that are on the same source line. You can minimize the necessary number of source lines by judicious use of this symbol, making more of the 98 lines in the source file available for the Description block.

Another way to lessen the number of lines in the Equation block (and thereby simplify editing) is to use the output label separator ",". This symbol may be placed between output labels on the left side of an equation to make each of them equal to the expression on the right side of the equation.

The symbol ":" should be used only after a flip-flop output label. An assembly error results if the symbol appears anywhere else in the Equation block.

Equations in the source file may be continued for more than one line, but a , pin label cannot be interrupted by the end-of-line (<CR>).

For example, suppose a device has the following pin list:

PIN*	FUNCTION	LABEL	
1	I	11	
2	I	12	
3	I	13	
4	I	14	
10	0	Fl	
11	0	F2	
27	Q	Q0	assume these are S/R flip-flops
28	Q	Ql	

Then the following are examples of valid expressions for the IFL Boolean source file:

I1	- label of an input pin
/I1	- complemented input label
I1 * /I2	- product
I1 + /I2	- sum
I1 * I2 + I3	- sum of products

And the following are examples of valid equations, as they might appear in the Equation block of the source file:

```
Fl = I1
F2 = I1 * /I2 + I3
Ql : S = I1                      these two are flip-flop equations
Ql : R = I2 * / I3
Ql S = I1                        another complete flip-flop equation
Fl = I1 * /I2 . F2 = I2 * /I3    two equations on the same line
Fl , Ql : S , Q0 : R = I4 * I3   three output labels equal same expression
```

6.6.3 USE OF THE BOOLEAN EDITORS

Use of the two Boolean editors is essentially the same. Descriptions in this subsection apply to both the LINE and TTY editors, unless otherwise noted.

When the editor begins execution, the block prompt "HEADING BLOCK" is displayed. This indicates that you are automatically placed into the Heading block of a new source file. Your response to this or any block prompt must be one of the Block commands summarized in Table 6-5 and described in following paragraphs. Note that each Block command must be terminated with carriage return (<CR>), but <CR> entered alone an a line puts you into editing mode with another set of available commands. It is with the editing mode commands that you can erster and change characters in the source

Table 6-5. Boolean Editor Block Commands

Name	Format	Purpose
Block change	Bn	Jump to block number "n" (1-4)
Delete lines	mDn	Delete "n" lines, beginning with line number "m"
Insert lines	mIn	Insert "n" null lines, ahead of existing line number "m"
List lines	mLn	List (display) "n" lines, beginning with line number "m"
Next block	N	Advance to the next sequential block
Print "x"	P	List the internal labels ("virtual pins") for this device
display Space	S	List the number of bytes remaining in the Symbol table
exit the editor	<ESC>	Return to Boolean command menu
editing mode	<CR>	Enable modification of lines

Note: parameters "m" and "n" are (1-98, except in the "B" if "n" is oMitted, it is assumed to be 1; if "m" is omitted, it is assumed to be the current line number.

Editor block commands are each described in detail in following paragraphs. Note that the <CR> command makes available a new set of commands for entering and modifying contents of the source file one line at a time.

B Command (Block change)

When you enter a B command, the editor responds with a new block prompt to indicate which of the source file blocks you are now in. The blocks are: 1 - HEADING BLOCK; 2 - PIN LIST BLOCK; 3 - EQUATION BLOCK; 4 - DESCRIPTION BLOCK. Note that a B command entered without parameters places you immediately into the HEADING BLOCK. Example: B3 places you in the EQUATION BLOCK.

D Command (Delete lines)

When you enter a D command, the editor deletes "n" lines beginning with line number "m". Lines in the HEADING BLOCK cannot actually be deleted; instead, they are replaced by null lines that consist of <CR> only. The D command has no effect in the PIN LIST BLOCK. In the EQUATION BLOCK and DESCRIPTION BLOCK, the specified lines are removed and succeeding lines are renumbered. Example: in the EQUATION BLOCK, 2D3 deletes lines 2, 3, and 4; former line 5 becomes line 3, and so on.

I Command (Insert lines)

When you enter an I command, the editor inserts "n" null lines after line number Im". Following lines are renumbered accordingly. You can insert meaningful data into the null lines after putting the editor into editing mode (i.e., with a <CR> command). The I command is effective only in the EQUATION BLOCK and the DESCRIPTION BLOCK; lines cannot be added in the other blocks. Example: 21 inserts one null line after line number 2; the new line becomes line number 3, and former line 3 becomes line 4, and so on.

L Command (List lines)

When you enter an L command, the editor displays "n" lines beginning with line number "m". Line numbers are included at the left of each line; only lines in the current block can be listed. In the PIN LIST BLOCK, all physical pins are listed regardless of the parameters in the command. The last listed line becomes the current line for the next editor command. Example: in the EQUATION BLOCK, L displays the current line number and contents of the line.

N Cammand (Next block)

When you enter the N command, the editor places you in the next sequential block of your source file. A new block prompt is presented to remind you which block you are in. The N command has no parameters; if any are included, they are ignored. Example: if you are in the PIN LIST BLOCK, N moves you to the EQUATION BLOCK.

P command (Print "x" labels)

When you enter the P command, the editor lists labels and functions of any fuses which must be programmed, but are not directly connected to physical pins. This command has no parameters; it is only active in the PIN LIST BLOCK, and only if the device has internal labels. The displayed list is similar in form to the PIN LIST, but with "x" appearing immediately before each pin number to indicate a "virtual pin".

S command (display Space)

When you enter the S command, the editor displays the number of free bytes remaining in the symbol table. You may enter the S command in any block.

<ESC) command (exit the editor)

When you enter <ESC>, the editor immediately returns you to the Boolean command menu. Your source file is just as you left it; you may proceed to assemble the source file, copy it, or return to the editor to modify it. Example: <ESC> <CR> results in the appearance of the Boolean command menu, as shown at the beginning of 6.6.

<ce> command (editing mode)

When you enter a <CR> alone an a command line, you can make modifications in the current line. Operation of the two editors is different in editing mode.

If you are using the TTY editor, the line is printed and you are prompted with ">" for a replacement line. Type a replacement line terminated by carriage return (<CR>) to effect the replacement. You will then be prompted to replace the next line in the block. Enter <ESC> at any time to return to the editor block command menu. These are the only editing mode operations with this editor.

To use the LINE editor in editing mode, you place the cursor at the desired location in the displayed line (with --> and BACK SPACE) and then enter any of the following single—character commands:

Command	Name	Function
A	Again	retrieve the original line from the symbol table
C	Change	change characters in the line, beginning at the current cursor position
D	Delete	delete character at the current cursor position
E	End	moves cursor to last character in the line
I	Insert	inserts characters at the current cursor position
R	Replace	deletes characters from current cursor position to the end of line, and expects new characters (like Insert)
S	Start	moves cursor to the start of the current line
X	eXtend	moves cursor to end of the current line and expects new characters
-->	cursor right	moves cursor one character to the right in the current line
BACK SPACE		moves cursor one character to the left in the current line
<CR>	carriage return	stores the current line and presents the next line of the block for editing
<ESC>	escape	after Insert or Change, ends the entering of new characters and permits another editing mode command to be entered; as an editing mode command by itself, exits editing mode and returns to the editor Block command menu
<RUB>	rubout	after Insert, erases all characters entered for this Insert operation and permits another editing mode command to be entered

The LINE editor display of the current line includes the prompt character "." after the line number when any command can be entered, or the prompt character ">" when inserted text is expected, or ":" when changes to existing text are expected. Note that <CR> after inserted or changed text accepts the changes, and <ESC> ignores them.

In the LINE editor if the current line does not exist or is a null line, the editor automatically prompts you for new text with ">". If you use editing mode in the PIN LIST BLOCK and attempt to store a null pin label, you are automatically returned to the editor Block command menu because all pin labels must remain defined.

6.6.4 USING THE BOOLEAN ASSEMBLER

When you have completed the editing of your source file, you are ready to assemble the source file. From the Boolean command menu, you enter "A" to , invoke the assembler. The prompt "Starring PTERM ?" is displayed at your terminal when the assembler begins execution.

You can enter a decimal term number followed by <CR>, or just <CR> in response to the prompt. If you enter a term number, the fuse information that results from the assembly affects only that term and succeeding terms, not terms with lower numbers. If you enter <CR> only, assembly begins with term 0 and the fuse map is reset to the virgin state. If you enter the number 0 followed by <CR>, assembly begins with term 0 but the fuse map is not reset. The fuse map is not reset if any other number is entered.

All equations in the EQUATION BLOCK of the source file are assembled after you have responded to the "Starting PTERM ?" prompt. The IFL Boolean assembler makes two passest all tokens are displayed on the terminal while assembly *is* in process. If an error is detected, a display indicates which token is in error and the type of error. The assembly halts when an error is detected.

The second pass of the assembler executes only if there were no errors during the first pass. During the second pass any outputs not defined in the equation set are reset to the virgin state for each used term. This permits greater flexibility during future use of these outputs. Note that for FPLA device equations, unused outputs are reset to the active state (the virgin state for that device), whereas for FPLS device equations unused outputs are reset to the inactive state (virgin state for that device). Thus in a programmed FPLA there will be a signal comming out of unused outputs.

The assembler inserts an end—of—file marker after the EQUATION BLOCK before assembly actually begins. The marker is removed when assembly is complete and you are presented with the Boolean command menu once again. To inspect the fuse map after a successful assembly, you should exit the Boolean module (with <ESC> at the Boolean command level) and press the programmer REMote key to utilize the Display command of the Remote IFL Editor (6.5.3).

Previously assembled terms are held in PROM programmer RAM when a new assembly is begun. To guarantee that only terms from the current assembly appear there, you should exit the IFL Boolean module with <ESC> at the Boolean command level and use the F (Erase) command described in 6.5.2.

. To actually program an IFL device after, a successful assembly, exit the Boolean module with <ESC> at the Boolean command level and follow programming procedures in 6.5.1 or 6.5.2.

6.6.5 BOOLEAN MODULE I/O OPERATIONS

You can enter "p" in response to the Boolean command menu to utilize I/O functions especially for Boolean source code. The following menu is then displayed:

```
1  Print Hardcopy

2  Output On Paper Tape
3  Input From Paper Tape, Generated From Option 2

4 Output To Another MPP80 Using TRANSKON
5  Input From Another MPP80 Using TRANSKON

Selection:
```

Enter a number from one to fixe after the colon (:) to select the I/O operation to be performed. If you press any other key, this function is terminated and the Boolean command menu is displayed.

These input and output operations are for source code only. Consequently, the formet used for operations 2 and 3 is not Signetics paper tape format. Instead, a variation of the Kontron TRANSKON format is employed; *see* the description of formet 61 in Appendix B for details.

To output assembled data in Signetics paper tape format, exit the Boolean IFL module software after a successful assembly and use the PROM programmer OUTput function described in 3.3.2.

After you specify the I/O operation to perform, the following display appears:

```
SELECT BAUD
1  110
2 300
3 600
4 1200
5 2400
6 4800
7 9600

Selection:
```

You enter a number (1-7) after the colon (:) to indicates the baud rate for the I/O operation you have selected. If you press any other key, the operation terminates and the Boolean command menu reappears. Note that only baud rates of 600 or less are permitted for I/O operations 2 through 5.

After you have selected the baud rate, the following message is displayed:

```
PRESS ENTER ON THE MPP80 FRONT PANEL TO BEGIN
DATA LENGTH — XXXX
BEGIN
```

The number "XXXX" indicates the hexadecimal number of bytes in the symbol table. You can change serial cable connections as necessary, then press "ENTER" to start the data transfer. If you press any other key on the PROM programmer front panel, the I/O operation is aborted. If you press "ENTER" twice, execution of the IFL Boolean module is terminated and no 1/0 is performed.

Following data transfer, you may safely press "ENTER" twice to return to IFL module execution. Then to reuse the Boolean firmware, press the "A" or "D" key on the data keypad. Previously entered data has not been changed.

6.6.6 NOTES CONCERNING IFL BOOLEAN OPERATIONS

This subsection presents specific infoimation and recommendations about using the Boolean IFL module, including:

- Pin Functions and Equations
- Control Terms
- FPLS Complement Array
- FPLS Enable Fuse
- Device—specific Considerations

Each listed topic is presented in following indented paragraphs.

Pin Functions and Equations

The sole purpose of the PIN LIST BLOCK in a source file is the assignment of labels to pins and special—purpose fuses. Listed pin functions are not subject to change by you. In particular, you cannot specify a complemented output in the PIN LIST BLOCK; this is accomplished via equations in the EQUATION BLOCK. Likewise, a bidirectional pin cannot be restricted to input only or output only in the PIN LIST BLOCK; its use in the equations of the EQUATION block how the pin is actually to be used.

You may use the character "%" in a pin label to designate a complemented input or output, but this character is ignored by assembler and the pin name must still be complemented (with "/" in the associated equation set. Any character except the predefined operators may appear in a label.

Equations consist of three main parts, as follows: output

expression " = " input expression

in which "output expression" specifies the output pin/variable to be programmed and the active level (high or low) for that variable. If the output pin is complemented, the active level will be LOW; if the output pin is not complemented, the active level will be HIGH. For this purpose it is important to note that pin labels appearing on the left of an equal sign are used as outputs, and pin labels appearing on the right of an equal sign are used as inputs.

For example, given the following pin list for an imaginary device:

PIN	FCN	LABEL	Comment
1	I	Il	input pin
2	I	12	input pin
3	I	13	input pin
4	I	14	input pin
10	0	F1	output pin
11	0	F2	output pin
17	B	131	bidirectional I/O pin
18	B	B2	bidirectional I/O pin
19	P%OE	P%OE	Preset or /Output enable, used with ENB
x31	Q	P1	assumed to be S/R flip-flop
x32	Q	P2	assumed to be S/R flip-flop
x33	D	DO	control term
x34	D	D1	control term
x35	CPL	CPL	complement array
x36	ENB	ENB	enable fuse

suppose that you want to use pin 10 as a complemented output. You can relabel Pin 10 to "%OUT" (instead of the default "Fl"). However, in order for the active level of Pin 10 to be a complemented output (i.e., programmed low or TRUE ACTIVE LOW) it must be so used in an equation. For example, it may appear as follows:

```
01../ %OUT = I1 * 12
```

If any output pin is used on the left side of more than one equation, the net result is an OR of the "Input expressions" from the equations
1:
with the active level of the output 1-11 determined by its use in the last "output expression". Thus if the following equations both appear in the EQUATION BLOCK:

```
01../ %OUT = I1 * 12 + 13 02..%0UT =
/12 * 13
```

the net result is equivalent to:

```
01..%0UT = I1 * 12 + 13 + /12 * 13
```

Note that the ative level is programmed HIGH by equation 02.

Control Textas

Control terms for 20—pin IFL devices are treated the same as an output pin DO = II * 12 ") except that control terms cannot be defined by an expression that includes the OR operator (+). If this is attempted, an assembly error results.

Control term outputs may be included among other output label types; for example:

```
    01.. DO , Fl , Dl , P1 : S = IO * /II
```

To blow all control term fuses, enter the equation

```
    01.. DO = 1
```

To leabe all control term fuses intact, the equation is

```
    01.. DO = 0
```

Complement Array

The complement array (CPL) for FPLS devices is treated like a bidirectional pin. To program the complement array, the equation might be (for example):

```
    01..CPL = I1 + 12 * 13
```

To use the complement array as an input, the equation might be (for example):

```
    02..P1   S = IO * CPL
```

Because the complement array has no active level, complementing it makes no sense. The assembler ignores the complement operator if it applies to the complement array label. Thus

```
    03../CPL = I1 + 12 * 13 is
```

equivalent to

```
    03.. CPL =      12 * 13
```

and

```
    04..P1 : S = I1 * /CPL is
```

equivalent to

```
    04..P1   S = I1 * CPL
```

Enable Fuse

The ENABLE FUSE (default label "ENB") in certain FPLS devices is **used** in conjunction with Pin 19 (default label "P%0E"), as indicated in the Device—specific Considerations that follow. It may be appear in the "output expression" of an equation only as follows:

```
ENB = P%OE          or          ENB = / P%0E
```

An assembly error results if you attempt to use the ENABLE FUSE label with any other output labels. However, more than one ENABLE FUSE may be included in an "output expression". Similarly, an ENABLE FUSE label must not appear in an "input_expression" with any other type of label.
Also, ENABLE FUSE labels cannot be used in "input expression" with the AND operator (*). For those devices with more than one ENABLE FUSE label, it is permissible to use these labels in the same

"input expression" provided they are joined by the OR operator (+).

Device—specific Considerations

Characteristics of the IFL devices dictate the following conditions governing use of the IFL,Boolean module with them:

FPGA 82S102/103 --

1. The device has no OR matrix, so the OR operator (+) must not appear in the equation set.
2. The active level logic of the device itself takes into account the fact that the AND matrix is replaced with a NAND matrix. Therefore you should consider that the device has AND gates when using the IFL Boolean module.

FPLS 82S104/105 --

1. Outputs of the Next State Field may be used as inputs.
2. The output array has no active level; thus it cannot be complemented in the equation set.
3. The fuse that determines the funtionality of Pin 19 (either PRESET or /OUTPUT ENABLE) has the default label "ENB". It is programmed as follows:

```
if 01..ENB = P%OE   then Pin 19 acts as PRESET
if 01..ENB = / p%OE then Pin 19 acts as COMPLEMENTED
                                        OUTPUT ENABLE
```

where "P%OE" is the label of Pin 19.

FPRP 82S106/107 --

This device has not active level, so the outputs must not be complemented in the equation set.

Device—specific Considerations(Continued)

FPLA 82S152/153 --

1. All outputs are bidirectional and have active levels (i.e., they may be complemented).
2. This device contains ten control terms, designated DO through D9.

APPENDIX B

DATA TRANSMISSION FORMATS

This appendix includes (in subsection B.1) descriptions of data transmission formats that may be used for serial communications between DIPP-808/EPP 80 PRO programmers and remote devices such as terminals, microcomputers, and printers. The formats are identified to the PROM programmers by a 2-digit code number during Input and Output operations; these operations are described in 3.3.1 and 3.3.2, respectively.

Also in this appendix (B.2) is the formal specification for Trans1on data transmission format (Format 60). This specification defines the actions required of sending and receiving process'ors (computers and/or programmers) which use this error-tolerant format.

B.1 DATA FORMATS

There is an entry for each defined format in this appendix. Entries are ordered by format number, and presented as follows:

```
    nn        name      description

         sample

              Input: (programmer action as receiver)

              Output: (programmer action as sender)
```

where:

nn	Denotes the format number;
name	Is the identifying name of the format;
description	Is a brief textual description of the format;
sample	Is a sample transmission stream in this format.

PROM programmer actions may be the same for one format as for another, similar format; thus some format entries include the statement "Same as Format xx" instead of a restatement of programmer actions for Input and Output.

10 **BPNF** ASCII—coded binary "P"ositives and "N"egatives, each data byte enclosed by "B" and "F".

```
<STX>BPPNNFPNPF BNN----PPF BPNPNPNPNF <ETX>
```

Input: Start of Text (<STX>) character is ignored. The programmer starts with first "B" received, sequentially stores each data byte as "F" is received. Input operation ceases when the End of Text (<ETX>) character or 10 nulls are received.

Output: The prorammer sends 50 null characters, Start of Text, 6 data bytes separated by spaces, carriage return, additional data bytes, etc. Transmission ends with End of Text character, followed by 50 nulls.

11 **BHLF** ASCII—coded binary "H"ighs and "L"ows, each data byte enclosed by "B" and "F".

```
<STX>BHHLHLLHLF BHL----HHF BHHLLLLLHF <ETX>
```

Input: Same as Format 10.
Output: Same as Format 10.

12 **B1OF** ASCII—coded binary "1"s and "0"s, each data byte enclosed by "B" and "F".

```
<STX>B11101001F B10----00F B10001011F <ETX>
```

Input: Same as Format 10. Output: Same as

Format 10.

20 ASCII HEX SPACE ASCII—coded hex digits, each data byte (i.e., each character pair) delimited by a space.

```
<STX>$A0000,<CR><LF>
CD 12 08 21 F4 CD 78 ... <ETX>
```

Input: After Start of Text (<STX>), the programmer recognizes the symbols "$A" as preceding a 4—digit hex starting address; comma, carriage return, and line feed mark the beginning of data bytes, which are stored sequentially from the starting address, unless another address is encoded. Input operation is ended when the End of Text (<ETX>) is received.

Output: Programmer sends 50 null characters, followed by Start of Text and the address for the first data byte. End of transmission is signalled by the End of Text character, which is followed by 150 nulls.

21 ASCII HEX PERCENT ASCII—coded hexadecimal digits, each data byte (i.e., each character pair) delimited by the percent character.

```
<STX>SA0000,<CRXLF>
 CD7:127,08521%F4%CM787...%<ETX>
```

Input: Same as Format 20.

Output: Same as Format 20.

22 ASCII HEX APOSTROPHE ASCII—coded hexadecimal digits, each data byte (i.e., each character pair) delimited by an apostrophe.

```
<STX>$A0000,<CR><LF>
 CD'12'08'21'FA'CD'78'...'<ETX>
```

Input: Same as Format 20.

Output: Same as Format 20.

23 ASCII HEX COMMA ASCII—coded hex digits, each data byte (i.e., each character pair) delimited by commas.

```
<STX>$A0000.<CR><LF >
CD,12,08,21,F4,CD,78,...,<ETX>
```

Input: Same as Format 20, except that a period follows the starting address instead of a comma.

Output: Same as Format 20, with period after starting address.

30 **BINARY** A binary stream of data.

<NUL> ... <NUL><DEL*>110100

Input: The programmer looks for any number of nulls, followed by a rubout character (<DEL*>) in which the most-significant bit is 1 (i.e., hex 0FF). Binary data is then stored in RAM according to specified Starting Address-to-Ending Address range. No end-of-file character is recognized in the input data stream.

Output: The programmer sends 50 nulls, a rubout character, and the data. It appends 50 nulls after the data stream.

31 **DEC BINARY** A binary stream of data.

<DEL*> <DEL*><NUL>110100

Input: The programmer looks for any number of rubout characters in which the most-significant bit is 1(<DEL*> = hex 0FF), followed by a null. Binary data is then stored in RAM according to specified Starting Address-to-Ending Address range. No end-of-file character is recognized in the input data stream.

Output: The programmer sends 50 rubout characters, a null, and the data. It appends 50 rubouts after the data stream.

32 **INTEL NON INTELLEC** ASCII-coded hex digits, with optional imbedded addresses and delete characters. This format is valid only for MPP-80S firmware version 2.0.

:3D520F28 ;

Input: The programmer looks for the data stream immediately after a colon. Data is stored sequentially from the specified Starting Address, but a new address field (hex digits enclosed in parentheses) is recognized as applying to data bytes immediately following. The delete characters "X" and *'9"* can be included between data bytes: X deletes the previous character; 1/ deletes the previous byte. The programmer recognizes a semicolon as an end-of-record character.

Output: The programmer sends 50 null characters before the colon which signifies the start of the data stream® After the semicolon signifying the end of the data stream, it also sends 50 nulls.

33 **7-BIT ASCII W/ ECHO** [Insufficient source information] Available only on the EPP-80 and/or with the mpn personality module.

Input: characters received over the RS 232 C interface are transmitted back to the sender for verification/display (?)

Output: "any input character writes the next output to RS 232" (?)

34 **7-BIT ASCII W/OUT ECHO** [Insufficient source information] Available only on the EPP-80 and/or with the MDM personality module.

Input: characters received over the RS 232 C interface are not transmitted back to the sender for verification/display (?)

Output:

40 **INTEL HEX** Record-oriented hex data with checksums.

```
:10000000CD12082F ... AB :10
...
:00001001
```

Input: The programmer looks for a colon as the start-of-record character. This is followed by a 2-digit (hex) byte count that defines the record length. Next is a 4-digit RAM load address, followed by two characters that represent the record type: "00" denotes a data record and "Cl" denotes an end-of-file record. Data which follows (2 characters per byte) is stored sequentially, beginning in the starting address contained in the transmitted record. Following the data is a record checksum, which is the two's complement of the binary sum of all the data bytes in the record. The programmer verifies this checksum and reports an error if a mismatch occurs. The input operation continues until an end-of-file record (byte count 00 and record type 01) is received.

Output: The programmer sends the start code (colon), a count of the data bytes in the record, the address of the first byte, record type, data, and a checksum. The end-of-file record described above is sent to terminate the transmission.

41 **MOTOROLA EXORCISOR** Record-oriented hex data with checksums.

```
S1130000CD120821 ... 08 S113
...
S9030000FC
```

Input: The programmer recognizes "Si" as start characters for a data record. This is followed by a 2-digit byte count, which includes two bytes for the address and one for the checksum. Next is the 4-digit address for the first data byte. This is followed by the data bytes and the checksum, which is the one's complement of the byte count. The end-of-file record uses start characters "S9", byte count 03, any 4-digit address, and the checksum FC.

Output: The programmer sends the start characters ("Sl"), the byte count, address of the first data byte, data, and the checksum. To facilitate display, the programmer includes non-printable carriage return/line feed characters at the end of each record. For the end-of-file record the programmer includes the address 0000, and other fields as previously described.

42 RCA COSMAC

Record—oriented sequential hex data with optionally specified address change.

```
!M0000<SP>
CD120821F4    ,<CR>
322C ... <CR>
```

Input: The programmer expects the start code "!M" to be followed by a start address and a space. The character pair ",<CR>" (i.e., comma and carriage return) terminates a data record. Data is stored sequentially from the start address, unless the previous record is terminated with ";<CR>" (i.e., semicolon and carriage return), in which case a new start address and a space precede additional data bytes. The last data record is terminated with <CR> only.

Output: The programmer precedes and follows the transmission with 50 null characters. The transmission stream for output is the same as for input, including use of commas and carriage returns as record delimiters.

43 F8 FORMULATOR

Eight—byte fixed length data records with single—digit checksums.

```
S0000
XCD120821F4    E<CR><LF>
```

Input: The programmer looks for the start character "S", which is followed by the starting address. The character "X" precedes each data record. Eight data bytes are contained in a record; the 1—digit checksum follOws data bytes, and precedes the carriage—return and line—feed characters that terminate the record. The asterisk is interpreted as an end—of—file character.

Output: The programmer sends data in output format identical to the input format.

43 SIGNETICS ABSOLUTE

Record-oriented hex data with address and data checks.

```
:00001020CD120821F4   BD
:00100020
```

Input: The programmer expects a colon start character for each record, followed by a 4-digit address and a 2-digit count of data bytes for the record. An address check is received next; this 2-digit quantity is the EXclusive OR of the previous byte with its predecessor byte, rotated left one bit, then that result is EXUed with it predecessor byte. Following the data bytes in a record is a data check, which is calculated in the same way. The end-of-file record includes an address and a byte count of 00, followed by the address check.

Output: The programmer sends 50 null characters before and after the transmission. Following each transmitted record the programmer sends a carriage return and line feed.

45 TEKTRONICS REX

Record-oriented hex data with address and data checks.

```
/00001010CD120821F4 ... 45
/00000010
```

Input: The programmer looks for the slash character "1" that signals the beginning of a record. This is followed by a 4-digit address, a 2-digit count of data bytes, and a checksum of the address and byte count. After the data bytes is a checksum of those data bytes. Each checksum is the 8-bit sum of the relevant bytes. The end-of-file record includes the byte count 00 and the address-count checksum.

Output: The programmer transmits this formet exactly as it expects to receive it.

46 MOS TECHNOLOGY — Record–oriented hex data with checksums.

```
;100000CD120821 ... 0555
;0000100010
```

Input: The start character is a semicolon; this is followed by a count of the data bytes in the record, and a 4–digit start address for the data. Data bytes come next, followed by a 16–bit binar• checksum of all bytes in the record (including byte count and address). The end–of–file record includes the byte count 00, a 4–digit total of data bytes in the transmission, and a 16–bit checksum of the record.

Output: The programmer transmits this format exactly as it expects to receive it.

47 LIST, 8 BYTES/LINE — Line–oriented hex data FOR OUTPUT ONLY.

```
CD 45 20 81 62 4F 53 37 <CR><LF>
```

Output: The programmer sends bytes of ROM data with spaces between the bytes; carriage return and line feed are sent after every 8 data bytes. Starting and ending addresses are as specified on the keyboard. This format is especially for display/printing of data.

48 LIST, 16 BYTES/LINE — Line–oriented hex data FOR OUTPUT ONLY.

```
CD 45 20 81 62 4F 53 37 63 5C 04 FF FF 56 62 3A <CR><LF>
```

Output: The programmer sends bytes of ROM data with spaces between the bytes; carriage return and line feed are sent after every 16 data bytes. Starting and ending addresses are as specified on the keyboard. This format is especially for display/printing of data.

60 **TRANSKON W/ACK** Record—oriented hex format with address and whole—record checksums, and record—levc.:1 ACK/NAK communication between sender and receiver.

```
;10000001120821   0555<CR>
           [ACK or NAK from receiver]
;00001001<0R>
```

Input: After you press appropriate keys to establish Input format and offset, the FROM programmer displays "rEAdy" while waiting for transmission to begin. During transmission, only the ENTER key is functional an the FROM programmer keyboard; press this key t-'ice fo discontinue communications and return the programmer to local standby mode. If the programmer is under remote control, press "GG" of t.L'z. remote keyboard between transmitted records to return the programmer to local standby mode.

The start character is a semicolon; this is followed by a 2—digit count of the data bytes in the record, a 4—digit start address for the data, and a 2—digit checksum of the previous six 4—bit nibbles. Up to 32 data bytes come next, followed by a 16—bit hex checksum of all bytes in the record (including byte count and address).

While a record is being received, the FROM programmer display shows a message like "rcd 0000 trns 01"; this indicates the transmitted starting address without offset (e.g., 0000) and the transmission number (e.g., 01). If a record is retransmitted due to an error, the transmission number is incremented. A carriage return (<CR>) is the record terminator.

When a record has been successfully received, the FROM programmer display indicates this with a message like "rcd 0000 rcvd" and returns the ACK (Acknowledge) character to the sender. The next record is then transmitted from the sender, and the display shows (for example) "rcd 0020 trns 01".

If a record is incorrectly received, the PROM programmer displays an error message such as "rcd 0000 E nAc J", noting the records start address (0000 in this case) and the NAK (Negative Acknowledge) character sent back to the sender ("J" in this case). If a record is incomplete, the receiving FROM programmer times out after 15 seconds and sends the NAK character "N". The record is retransmitted; if a second timeout occurs for a record, the PROM programmer displays "E tiME" and beeps for Operator attention.

The end—of—file record includes the byte count 00, a 4—digit total of data bytes in the transmission, and a 16—bit checksum of the record. After successful receipt of an entire transmission, including the end-of-file record, the PROM progammer displays "dAtA corrEct". If the number of data bytes field of the end-of-file record does not check with the receiver's count, the message "E nr oF rcds" is displayed; under these circumstances the integrity of the entire file is in question and it should be retransmitted.

The PROM programmer transmits an asterisk (*) to the sender after each record checksum is verified; this is the ACK character. One of the following NAK (Negative Acknowledge) characters is sent if an error occurs during transmission:

Character	Meaning
H	Character overrun Character framing Undefined character
K	Data count error
L	Address checksum error
M	Data checksum error
N	Timeout error (after 15 sec.)
o	Number of records error (end-of-file record)
U	Any error

Output: The programmer transmits this format exactly as it expects to receive it. A message such as "rcd 0000 trns 01" is displayed while the PROM programmer is sending a record; the address (e.g., 0000) includes any offset that was keyed in. While it is awaiting the ACK or NAK, the programmer displays "rcd 0000 rcvd ?". If no response is received within 5 seconds, the PROM programmer displays "rcd 0000 E tiME", and retransmits the record. If the programmer times out a second time for the record, transmission ceases and the PROM programmer beeps for the operator, displaying the message "E tiME".

When the positive acknowledgement character "*" is received, the sending PROM programmer briefly displays "rcd 0000 rcvd" and begins to transmit the next record. During record transmission, a message such as "rcd 0020 trns 01" is displayed. If a negative acknowledgement character (as previously listed) is received, the display indicates it: "rcd 0020 E nAc J", where "J" is the NAK character (in this case). That record is then retransmitted, while the programmer display shows "rcd 0020 trns 02" (i.e., the second transmission of the record with Start address 0020).

After the end-of-file record has been correctly received and acknowledged, the sending PROM programmer displays "dAtA corrEct". If the message "E nr oF rcds" appears after transmission is complete, data integrity of the entire file is suspect; the entire transmission should be repeated.

61 **TRANSKON WOUT ACK** Record-oriented hex format with address and whole-record checksums.

```
;10000001120821 ... 0555<CR>
;00001001<CR>
```

Input: PROM programmer actions are similar to Format 60, except that ACK/NAK characters are not sent. FROM programmer displays indicate errors with the NAK character codes. The display "dAtA corrEct" indicates a successful transmission.

Output: PROM programmer actions are similar to those for Format 60, except that ACK/NAK characters are not expected after each record; records are transmitted in immediate succession. After the end-of-file record has been transmitted, the PROM programmer display shows "trnS".

63 **TRANSKON MESSAGE** ASCII-characters for display on the PROM programmer; FOR INPUT ONLY.

```
PrESS EntEr......
```

Input: The programmer accepts exactly 16 characters and displays them as best it can; on the MPP-SOS some non-alphabetic characters may be unintelligible symbols in the display. The character <Control>-G (BEL) is accepted and causes the programmer audio tone to beep; this is not counted as one of the 16 characters. The displayed message remains until it is cleared by pressing ENTER twice, or with "G" from a remote controlling device. Note that the character "G" is considered patt of the display until 16 characters have been received.

70 **12-BIT BINARY** Stream-oriented binary data (octal output), with 2 RAM locations to hold each 12-bit value. Available only on the EPP-80 and/or with the MDM personality module.

```
10000000
10000000
11000000
00000000
00010010
00100100
   00.
```

Input (BIN load): This format. is especially for loading 12-bit binary data from paper tape. The programmer looks for any number of leader punches (10000000) to begin, and ignores them; it also ignores field settings (11000000). The least-significant 6 bits of eacl. 8-bit "row" is data or an address; 2 consecutive 8-bit rows represent one 12-bit "word". A punch in the 7th bit (Olxxxxxx) indicates that an address is specified in this and the next 8-bit row. Stored data occupies the stated starting address in RAM, and the address one PROM-size greater. For a 512-byte PROM, data specified to start at address 0000 (hex) actually is stored there and at 0200. The programmer recognizes any number of trailer punches (10000000) as signifying the end of transmission.

Output: An octal dump of the 12-bit data (stored in RAM as described above) is sent for display to a terminal or serial printer. The display includes a labeled start address for each line, followed by as many as 8 data values.

71 **12-BIT BINARY DUMP** Stream-oriented binary data (OUTPUT ONLY) from 2 RAM locations that hold each 12-bit value. Available only on the EPP-80 and/or with the MDM personality module.

Output: This format is similar to Format 70, except that displayed data is 12 binary bits in 3-bit fields. Two 12-bit "words" of data are displayed per line, with each octal address specified.

80 **INTEL 8086 HEX** Record-oriented ASCII coded hex data for object modules associated with 16-bit processor chips (INPUT ONLY). Available only an the EPP-80 and/or with the MDM personality module.

```
●● ●
020000020020DC
●● ●
```

Input: Data records and end-of-file records for this format are the same as for Format 40. Start Address records (denoted record type "03" in the 7th and 8th hex digits) are ignored. Instead, Extended Address records (record type 02) are used to specify bits 4-19 of the Segment Base Address, which defines a 64K byte region nf memory. Bits 4-19 are called the Upper Segment Base Address (USA); bits 0-3 of the Segment Base Address are zeros. The 3-digit USEA field immediately foliows the record type field in the record; it is followed by the 2-digit checksum.

The actual RAM address for a data byte is the sum of the Segment Base Address and an offset, where the offset is the sum (modulo 64K) of the Load Address Field of the data record and the index of this data byte in the'record. Extended address records may appear anywhere in the transmission; the established Segment Base Address remains in effect until a new Extended Address Record is received. The positive or negative offset keyed-in at the programmer (if any) is added to the calculated address. Programmer actions are otherwise identical to those for Format 40.

B.2 TRANSKON FORMAT SPECIFICATION

TransKon (Format 60) is a serial communication format that uses eight bits per character, with one stop bit; data is transmitted as ASCII characters. Only the seven least-significant characters are effective: the parity bit is set to 0 for transmission and ignored by the receiver. No handshake lines are required in the RS 232C interface; only pins 2, 3, and 7 are used. Transmission speed is limited to 600 baud to accomodate device calculations and displays.

B.2.1 SCOPE AND PURPOSE

This specification defines a format and protocol for communication between two independent systems: typically two PROM programmers or a PROM programmer and a microcomputer. The Kontron MPP-80SAM and EPP-80 PROM Programmers can use TransKon.

The purpose of the TransKon specification is to provide a communications protocal that includes error recovery, minimal and straight-forward operator Intervention, and assurance of data integrity for both the sender and the receiver. TransKon is useful over very long or somewhat "noisy" transmission lines because it permits transmission delays of up to three seconds.in either direction.

B.2.2 DATA LINK REQUIREMENTS

The transmission medium should be a full-duplex asynchronous communication line, such as an EIA RS 232C line. A part of the link may be provided by a common-carrier voice circuit with a modern (or modern and acoustical coupler) at each end of the link. A typical transmission rate is 300 baud, but the TransKon format does not require a particular data rate. Round-trip transmission delay time cannot be less than 6 seconds.

B.2.3 DATA FORMAT

Binary data is transmitted as 8-bit bytes; each byte is associated with a 16-bit address. For most transmissions, data addresses are sequential and the address of the first data byte is 0000 (hexadecimal). One collection of data occupying sequential addresses (f.e., one "file") may be appended to another file by making the first address of the second file one greater than the last address of the first file. However, if some or all of the addresses for the second file are the same as addresses for the first file, data in the second file will overlay (i.e., supersede) data in the first file. Receivers have limited address space; some addresses may not be valid for a particular receiver. For example, the MPP-80SAM and EPP-80 standard complement of memory is 64k bits, which is 8K bytes; this corresponds to memory addresses 0000 through 1FFF. Memory for these PROM programmers may be expanded to 32K bytes, corresponding to addresses 0000 through 7FFF.

Data is transmitted as ASCII characters, coded as in Appendix A. Each character consists of eight bits; the seven least—significant bits correspond to the ASCII code. The sender may include anything in the eighth bit, but the receiver must ignore the eighth bit. One start bit and one stop bit per character are appended by hardware upon transmission and removed upon reception.

Each byte of binary data is actually transmitted as a pair of hexadecimal digits, and each digit is a coded ASCII character. For example, the binary value 00101101 is transmitted as the codes for "2D"; that is, the transmitted bit patterns are "00110010" and "01000100".

Up to 32 bytes of data may be transmitted in a data record. This means that data records are necessarily less than 80 characters long, facilitating implementation an systems that include 80—character line buffers. A complete transmission tvpically consists of more than one data record; a transmission must be ended with an end—of—file record.

Each data record includes a start address, which is the address in the receiver for the first data byte. Additional data bytes in the record are stored in consecutive addresses from the start address. Usually, contiguous data is transmitted; therefore the start address of each successive record is typically the next higher address from that occupied by the last data byte of the previous record.

8.2.4 RECORD FORMAT

A TransKon data record must have the format:

    ;JJKKKKLLDD ... =Cell« CR>

where:

	is the "start of record" character
JJ	is the 2—digit hexadecimal count of the number of data bytes in this record
KKKK	is the 4—digit hex start address for the data in the record
LL	is the 2—digit hex checksum for the previous 6 hex digits
DD DD	is up to 64 hex digits, where each pair represents a byte of data
MMMM	is the 4—digit hex checksum of all previous hex digits in the record
<CR>	is a coded ASCII carriage return (i.e., hex OD)

An ASCII line—feed character (hex OA) may optionally be included after the carrriage return.

Data from a data record cannot be stored in the receiver memory unless the address checksum "LL" is verified.

The end-of-file record in a TransKon transmission contains a zero byte count ("JJ"), and a hex count of total distinct records in the field "KKKK" (retries do not add to this count). For a complete and accurate transmission, the receiver must verify this record count.

B.2.5 ACKNOWLEDGEMENT CHARACTERS

After a record is completely transmitted the sender expects an acknowledgement from the receiver. This may be a positive acknowledgement (ACK) or a negative acknowledgement (NAK). The ACK character is an asterisk ("*", ASCII 2A hex); this indicates to the sender that the record was correctly received. A NAK character indicates that the record was incorrectly received.

Any character other than an asterisk is considered to be a NAK character; however, some ASCII characters are especially designated to identify particular error conditions, as indicated in the discussion of Format 6C in B.1. If a NAK is returned to the TransKon sender, the last record is retransmitted; retransmission continues until an ACK is returned.

B.2.6 COMPLETE TRANSMISSION

A complete transmission in TransKon format consists of any number of data records each followed by an acknowledgement from the receiver, and an end record also followed by an acknowledgement. Retries (retransmissions) of .records are not limited in number; retransmission of a record is required if a negative acknowledgement is received by the transmitter.

Each data record and the end-of-file record must be fully packed; that is, valid ASCII characters must be sent consecutively, as fast as the transmission baud rate permits.

Prior to the first data record, anything may be transmitted except a semicolon (;). If characters other than data records or an end-of-file record are transmitted, these must be followed by an ASCII carriage return and line-feed (<CR><LF>). The carriage return and line feed characters immediately precede TransKon records to clear the line buffer of a device that uses buffering.

When a receiver notes an error in a transmission, it can send the universal NAK character "U", or a type-specific NAK to indicate the type of error. The sole exception to the foregoing rule is that the NAK character "o" must be sent to indicate a number-of-records error in the end-of-file record. All NAK characters except "0" result in the same automatic response (retransmission of the record). The advantage to using type-specific NAKs is that the operator of the transmitting device can be informed of the type of error. en the MPP-80SAM and EPP-80 PROM programmers, for instance, the NAK code is displayed by the transmitter and the receiver.

B.2.7 TIMING CONSIDERATIONS

The TransKon receiver must send an acknowledgement character within one second after receiving the end of a record; an acknowledgement character must not be sent before receiving the end of the record being acknouledged. TLerefore, even if an error is detected soon after receipt of a record begins, the receiver must wait for the end of the record before sending the NA} character. End of a record is defined as the terminating carriage return character.

To ensure that the receiver sends an acknowledgement even if the carriage return is improperly received, the receiver should set a time—out period. The time—out period is determined by the receiver based upon the data count contained in the record. The receiver sends a NAK character within one second after it should have received the carriage return to terminate the record.

The receiver should wait at least 15 seconds after sending an acknowledgement character for the semicolon that signifies the beginning of the next record. If no new record is begun in that time period, the receiver should send An unsolicited timeout NAK. After two consecutive timeouts of this type, it is assumed that the transmission line is dead; the receiver should cease reception, display an error message, and signal for the local operator.

The transmitter must clear its Input buffer immediately before it sends the carriage return that terminates a record transmission. This ensures that a previously received ACK is not interpreted as applying to the record just transmitted; the procedure also guards against interpretation of line "noise" as responses from the receiver.

The transmitter must timeout if an acknowledgement character is not received within seven seconds after sending a record terminator (carriage return). Upon the second such timeout, it is.assumed that the transmission line is dead; the transmitter should cease transmitting, display an error message, and signal for the local operator.

B.2.8 IMPLEMENTATION REQUIREMENTS

The TransKon standard is defined by the data and timing requirements stated in previous subsections. Each device which is to meet the standard must perform as stipulated.

A system with TransKon implemented should specify data transmission rates for which TransKon may be used; this is important because of TransKon timing requirements. For example, systems which perform real—time decomposition of records may thereby be limited in the maximum data rate they can accept. Furthermore, the timeout function for end of a record is dependent upon a transmission rate; it could overflow too slowly if a carriage return is missed, resulting in loss of more than one data record. The MPP-80SAM and EPP-80 PROM programmers may be used with TransKon at all baud rates between 75 and 600 baud. The MIT-80SAM built—in modern can be used only at 300 baud.

MDM / EPP- 80 FORMAT 81

TM990 ABSOLUTE OBJECT CODE FORMAT

Description of the I/O format 81 in MDM and EPP-80 for data transfers from and to Texas Instruments development systems using the "TM 990 Object Code Format".

09.09.84 by D.H.
translated by W.S.

CONTENTS

1.1 Usage Notes

The new format 81 refers to the Texas Instrument mrM 990 Object Code Format", also called "SDSMAC", running under DX 10 operating systems. For users of other operating systems, TI offers conversion utilities to allow a data transmission with format 81.
The format is implemented in the EPP-80 operating software starting with Rev. 4.0, in the MDM it is also Rev. 4.0 or, with the old Rev. counting system, Rev. 2.6.

In input mode, the programmer waits for tag before it starts to evaluate the subsequent tags and data. The transmission must be invoked with the "PF" (Print File) command. Should you use a "CC" (Copy) command, a 'CRLF' (Carriage Return Line Feed) or 'FF' (Form Feed) has to be inserted at the beginning of the file.

Data not transmitted with "PF" or the modified "CC" in the TM 990 Object Code Format cannot be received by the PROM programmer with format 81.

1 Format Description:

Tags Evaluated By The PROM Programmer:

Tag	Hex Field (4 Characters)	Second Data Field (4, 6 or 8 Chars)	Meaning
0	length of relo-catable code	program name (8 chars)	start of program
1	start address	not existing	absolute start address
2	start address	not existing	relocatable start address
3	address of last occurence of a symbol	symbol (6 chars)	last occurence of external reference in relocatable code
4	dito	dito	as obove in absolute code
5	address	symbol (6 chars)	relocatable external definition
6	dito	dito	absolute external definition
7	checksum of present data block	not existing	checksum for detection of errors
8	ignore checksum	not existing	checksum not used to detect errors
9	load address	not existing	absolute load address
A	dito	not existing	relocatable load address
B	data	not existing	absolute data

Tag	Hex Field (4 Characters)	Second Data Field (4, 6 or 8 Chars)	Meaning
C	data	not existing	relocatable data
D	absolute load address, not generated by assemble]	not existing r	defines start address to load to
F	not existing	not existing	end of data block
G	position	symbol	relocatable symbol definition
		(6 chars)	
H	dito	dito	absolute symbol definition
Tag	Hex Field (4 Characters)	Second Data Field (4, 6 or 8 Chars)	Third Data Field (4 Chars)
I	PSEG(R) address	program ID (8 chars)	not existing
J	DSEG(R)/CSEG(R) address	symbol (6 chars)	common no.
M	DSEG length	$data	0000
M	CSEG length	$blank	0001
M	blank common length	common name (6 chars)	common no.
N	CSEG(R) address	common no.	not existing
P	CSEG(R) address	dito	not existing
R	value	repeat count	not existing
S	DSEG(R) address	not existing	not existing
T	DSEG(R) address	not existing	not existing
U	0000	symbol (6 chars)	not existing
V	PSEG(R) address of chain entry	dito	not existing
W	DSEG(R)/CSEG(R) address	dito	common no.
X	DSEG(R)/CSEG(R) address of chain	dito	dito

2 Example

Below, you see a program and the SDSMAC- record derived from it:

Sample	SDSMAC 945278 **			
0001				IDT 'SAMPLE'
0002	0000	0006		DATA WSPACE
0003	0002	008A		DATA START
0004	0004	0000		DAT 0
0005	0006		WSPACE	BSS 32
0006	0026		TABLE	BSS 100
0007	008A		START	
0008	008A	04CC		CLR 12
0009	008C	04C0		CLR 0
0010	008E	0202		LI 2,TABLE
	0090	0026'		
0011	0092	C800		MOV 0, TABLE+2
	0094	0028'		
0012	0096	1001		JMP $+4
0013	0098		LOOP	
0014	0098	0204		LI 4,>1234
	009A	1234		
0015	009C	0244		ANDI 4,>FEED
	009E	FEED		
0016	00A0	DC84		MOVB 4,*2+
0017	00A2	0205		LI 5,>5555
	00A4	5555		
0018	00A6	C805		MOV 5, TABLE
	00A8	0026		
0019				END
Source	position	Machine		
Statement	Counter *	Code		

* address relative to first object byte

The corresponding object code format file looks as follows:

```
000AASAMPLE A0000C0006C008AB0000A008AB04CCB04C0B0202C0026BC8007F200F 000
C0028B1001B020481234B0244BFEEDBDC84B0205B5555BC805C00267F3C1F        000
      SAMPLE   00/00/00 08:14:23            SDSMAC 945278 **
```

EPP- 80 / MDM FORMAT 82

HP64000 ABSOLUTE OBJECT CODE FORMAT

Description of the I/O format 82 in MDM and EPP-80 for data transfers from and to Hewlett- Packard 64000 development systems using the "HP64000 Absolute File Format".

15.09.83 by D.H.
translated by W.S.

CONTENTS

1 Usage Notes

The new format 82 refers to the Hewlett- Packard "HP64000 Absolute File Format", used in HP's 64000 series development systems for file transfer via RS-232C serial interface.

The PROM programmer uses the same records as all HP64000 Logic Development Systems do to transmit or receive programs through serial interface:

```
"Copy <FILE> :absolute to rs232"

             or

"Copy rs232 to <FILE> :absolute"
```

In the PROM programmer, input and output functions are invoked as usual.

If an odd number of data bytes is transmitted, one fill byte (0) is appended after the last data byte.

2 Format Description

In general, all records (information and data) are transmitted binary.

2.1 Record 1

This record contains information an data bus width, data base width, transfer address (LS word) and transfer address (MS word). Additionally, at the beginning of each record, the number of transmitted words and at the end, a checksum is transmitted.

Structure Of Record 1:

```
Word Nr.          Format                         RS-232C
                                                 number of words!
1          15       8   7      0                 15             8
                                                 7              0
2          15       8   7      0                 15             8
                                                 7              0
3          15       8   7      0                 15             8
                                                 7              0
4          31       24  23     16                31             24
                                                 23             16
                                                 Checksum
```

Number of words:	In record 1 always 4
Word 1:	Defines data bus width
Word 2:	Defines data base width
Word 3:	Transfer address, LS word
Word 4:	Transfer address, MS word
Checksum:	Modulo 256 sum of word 1 through 4

2.2 Record 2 Through n

These records contain information an the amount of transmitted data, load address (LS word), load address (MS word) and data itself. They consist of m words, whereby m S 128. To these m words, at the beginning of the record the number of transmitted words and at the end a checksum is added.

<u>Structure Of Record 2 Through n:</u>

Word Nr.	Format				RS-232C number of words!	
1	15	8	7	0	15	8
					7	0
2	15	8	7	0	15	8
					7	0
3	31	24	23	16	31	24
					23	16
4	15	8	7	0	15	8
• • •					7	0
• • •						
• • •						
m	15	8	7	0	15	8
					7	0
					checksum	

Number of words:	Number of words in this record
Word 1:	Number of data bytes
Word 2:	Load address, LS word
Word 3:	Load address, MS word
Word 4:	First data word
Word m:	Last data word
Checksum:	Modulo 256 sum of word 1 through m

3 Example:

```
Record 1
Number of words     4
Word width         8
Address base       8
Transfer addresse   0000H

Record 2
Number of words     14
Number of data      21
Load address       0000H
2104 01F9 003C F507 0732 0501 CD00 02F1
FEM C205 0000

Record 3
Number of words     16
Number of data     26
Load address       0150H
4845 574C 4554 5420 2D20 5041 434B 4152
4420 2036 3430 3030 2020

Record 4
Number of words     30
Number of data      54
Load address       0015H
04C5 CD00 02C1 3E55 B8C2 0400 0CC5 C13E
44B9 CA15 0013 D5D1 3E12 BAC2 2100 3E34
BBC2 2100 2BE5 E13E 56BC C22A 003E 78BD
C22A 00C3 0400

Record 5
Number of words     7
Number of data      8
Load address       0200H
3A05 01E6 411F 1FC9
```

Data transmitted via RS-232

```
a  b   c    d   e      f  a g    h    i     k
040008000800000000100E001500000000210401F9003CF50707320501CD00
                       l  f   a g     h  i      k
02F1FEAAC2050000DA10001A015000004845574C455454202D205041434B
                                   f  a  g    h    i     k
41524420203634303030202020651E003600150000 04C5CD0002C13E55B8C2

04000CC5C13E44B9CA150013D5D13E12BAC221003E34BBC221002BE5E13E
                                            f   a g    h     i    k
56BCC22A003E78BDC22A00C304006A07000802000003A0501E6411F1FC9
f  m
7800
```

The small letters directly above each new word or byte transmitted have the following meanings:

```
a:  length of record
b:  word width
c:  addressing base
d:  transfer address, LS word
e:  transfer address, MS word
f:  checksum
g:  number of valid data bytes to be transmitted
h:  load address, LS word
    load address, MS word
k: data (up till "f" = checksum), eventually with fill byte
1: f ill byte
m: end character
```

APPENDIX C
IC HANDLER

The MPP-80S can be connected to an IC handler such as the EXATRON Model 800. Special communications formats • 49 and 50 are used for programming and verifying chips in the handler. For this application, data is transmitted through the appropriate personality module and socket adapter. The serial port of the MPP-80S is a controller interface; this port is not avialable for other communications while the IC handler is connected to the programmer.

C.1 INTERFACE DESCRIPTION

Control information is passed between the MPP-80S and the IC handler via a cable from the handler that plugs into the serial port on the front panel of the PROM programmer. The RS 232C handshake line DSR, STR, and RTS are used to signal Start, Pass, and Fail, respectively.

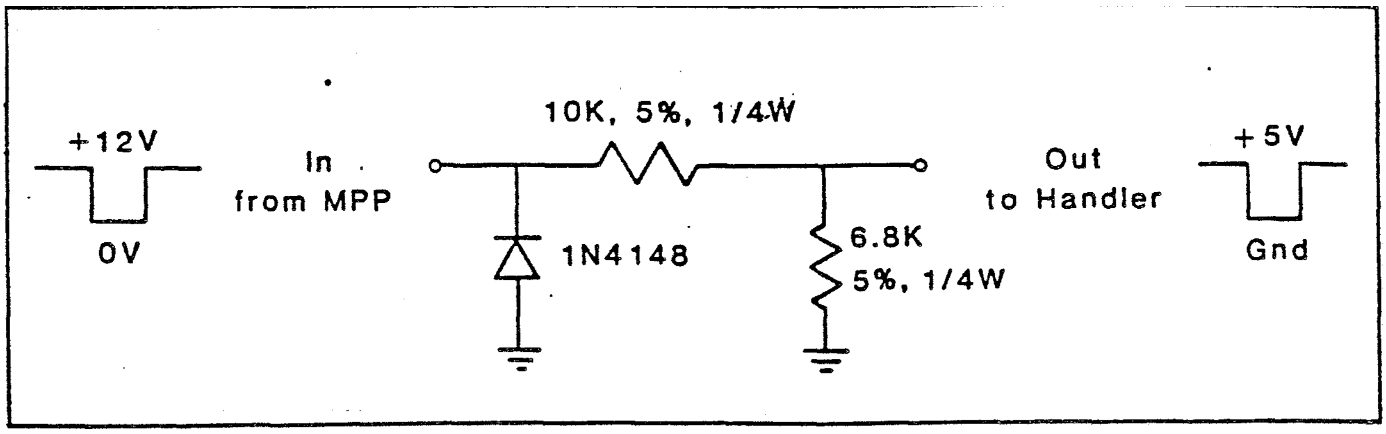
The Start signal is sent by the handler to indicate that a device to be programmed and/or checked is in position (and that pin contact has been effected). The Pass or Fail signal is sent from the MPP-80S after the Program or Check Operation is complete. The FROM programmer then waits for the next Start signal.

When the handler receives a Pass or Fail signal, it routes the PROM device to an appropriate bin, positions the next device, and sends a Start pulse to the programmer.

A device is programmed and/or checked via a cable that attaches to the socket adapter; the cable connector is seated much like a PROM device in the socket adapter and on the personality module appropriate to the device. The "Contact Head" in the IC handler contacts all pins of the device for power, addresses, data, chip selects, and programming.

The Start pulse sent by the IC handler to the MPP-80S is active—low, ground—referenced, in the range 3 to 25 volts, lasting at least 100 microseconds. The Pass/Fail signal sent from the programmer to the handler is idle at +12 volts, with pulse amplitude —12 volts lasting 1.4 milliseconds. The external network shown in Figure C-1 can be used to translate Pass/Fail signals to 5 volt TTL level if necessary.

Figure C-1. Level Translator Network



The cable connection between the MPP-80S serial port an the IC handler control port arc as follows:

MPP-80S Pin	Function	Exatron Handler Pins
20	Pass signal(MPP to Handler)	1
4	Fail signal (MPP to handler)	2
6	Start Signal (Handler to MPP)	3
7	Ground	7

The control port at tue back of the IC handler accepts a 14-pin rihbD7-. connector (Amphenol 57-30140). The connecting cable is available fru:, EXATRON.

A Kontron Interface Board which is available with the EXATRON Model nü IC handler incorporates the translation network illustrated in Figure C-1. This board includes D1P switches or jumpers that control the active levels of signals from the programmer. DIP switches 2, 4, and 6 should be CLOSED (ON) for active low signals.

C.2 PROGRAMMING WITH THE IC HANDLER

The OUTput function of the MPP-80S is utilized for programming devices on the IC handler, and for verifying devices that have already been programmed. For **a** complete description of the OUTput function, see 3.3.2.

To program a PROM in the handler as if it was plugged directly into the socket adapter, press the OUT key on the processor. Then select format 49, enter a Starting and Ending Address (or accept displayed de'aults), and an address offset.

WLen you press ENTER after entering (or accepting) an offset, the programmer display shows "IP" until a Start pulse is received from the IC handler. Upon receipt of this pulse, the PROM programmer begins the programming sequence: CHecK 2 (for illegal bits) is performed, followed by actual programming, followed by CHecK 1 (Verify).

If the PROM passes Check 2, programs without error, and verifies then the MPP-80S sends a Pass signal to the IC handler. The handler routes that PROM to the "pass" bin and positions the next PROM device.

If the PROM fails Check 2, does not program properly, or fails the verify check (Check 1) then the MPP-80S sends a Fail signal to the IC handler. The handler routes that PROM to the "fail" bin and positions the next PROM device.

During Check 2 in the programming cycle a row of dashes is displayed an the PROM programmer. Uhen programming actually begins, "Pr xxxx" is displayed where xxxx is the PROM device number.

Handler Example: Program a 2716 PROM in the EXATRON IC handler.

	Fc 2716	Processor in standby mode.
OUTput	OU 00	Output command accepted; transmission protocol is 00 (not a valid protocol).
4 9	OU 49	Protocol code is now 49: programming via the IC handler.
ENTER	OU SA 0000 49	Default Starting Address is 0000.
ENTER	OU EA 07FF 49	Accept Starting Address; default Ending Address is displayed.
ENTER	OFFSE 0000	Accept Ending Address default; offset is currently 0000.
ENTER	----------	Accept default offset value (0000); CHecK 2 in progress.
	Pr 2716	Programming in process.
	Fc 2716	Programming and Verify (CHecK 1) complete.

To verify (CHecK 1) a PROM in the handler as if it was plugged directly into the socket adapter, press the OUT key an the processor. Then select format 50, enter a Starting and Ending Address (or accept displayed defaults), and an address offset.
The programmer display shows "IC" until the Start signal is received.from the IC handler. A Pass or Fail signal is sent from the PROM programmer uhen the verify operation is complete, as in the programming cycle.

APPENDIX D

ERROR MESSAGE SUMMARY

This appendix presents error messages that may appear in the PROM programmer display during operation, an Interpretation of their meaning, and recommend-,:d user actions.

The following messages may appear on either the EPP-80 or the MPP-80S with nny personality module; display format shown is that of the MPP-80S:

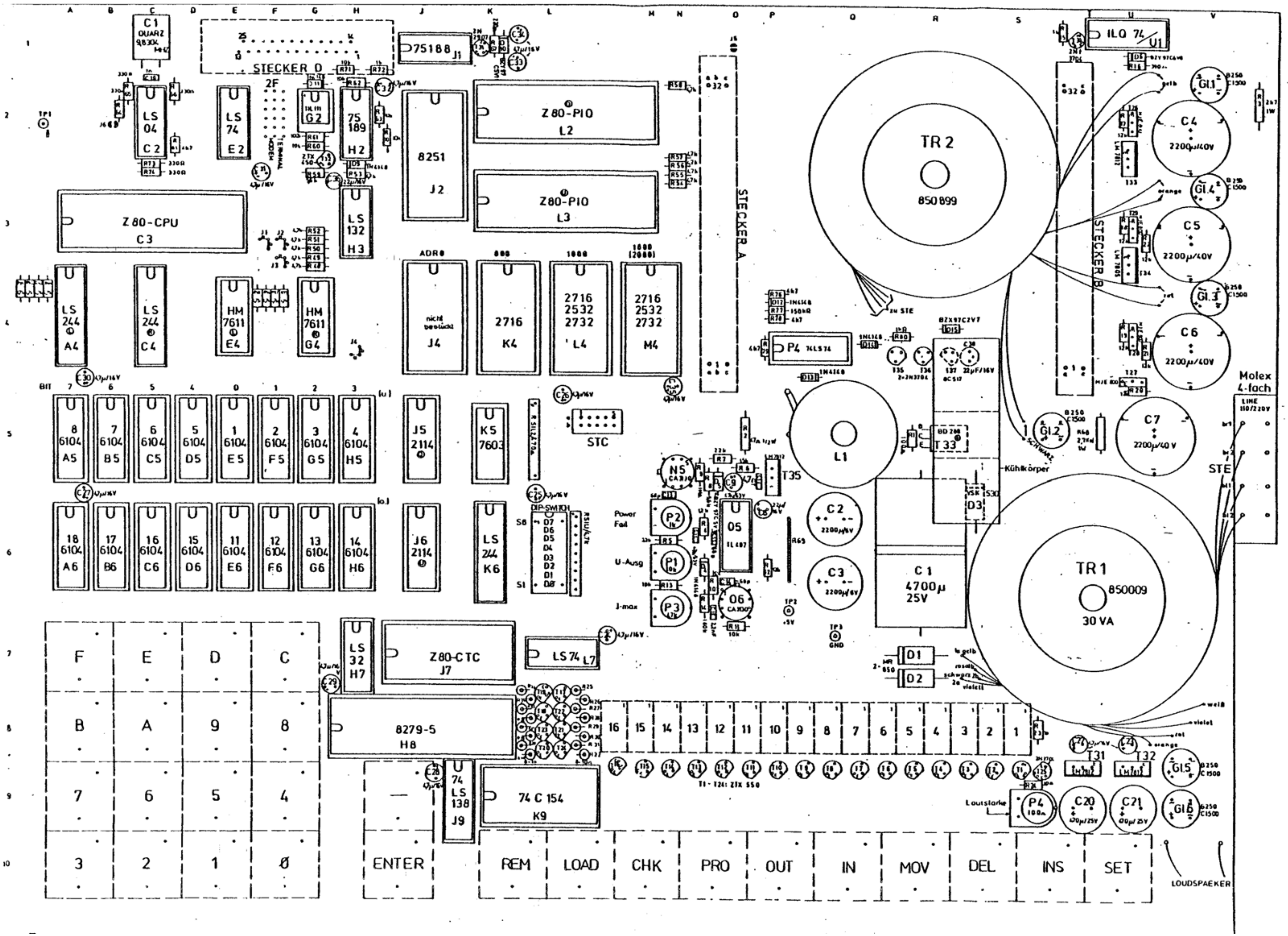
Display	Meaning	Suggested User Action
E CrC	RAM data integrity is suspect	If the error occurs during device initialization, the unit is not usable If it occurs during CHecK 1, CHecK 2, OUTput, or PROgram operations, you can reinitialize (power off, then on) and retry the operation.
E PE	Incoming power fluctuations (>15%) caused internal shutdown	Reinitialize the device (power off, then on).
E CH0	PROM is not blank	Use another PROM, and/or erase this one (if possible).
E CH1	PROM contents do not match RAM	Retry PROgram operation, perhaps with a different chip.
E CH2	This PROM cannot be programmed to match RAM	Use another chip. Verify that RAM data polarity is appropriate to the chip; use memory INVert function if appropriate.
E Pr	Error during PROgram operation	Retry operation, perhaps with another chip of the same type.
E So	Improper socket (some modules)	Check that installed socket adapter is meant to be used with this personality module (see Comparison Chart).

Display	Meaning	Suggested User Action
E RAM	Invalid RAM address specified	Repeat last operation, specifying only addresses valid for your RAM (minimum RAM includes 0000 - 01FF); check offsets.
E In	Invalid character or checksum during INput	Retransmit record or file; see Appendix E.
E RAM	Invalid RAM address specified	Repeat last operation, specifying only addresses valid for your RAM (minimum RAM includes 0000 - 01FF); check offsets.
E In	Invalid character or checksum during INput	Retransmit record or file; see Appendix B.

The following messages may be output by the MDM personality module when it is used an either the EPP-80 or MPP-80S PROM programmer; display formst is that of the EPP-80:

Display	Meaning	Suggested User Action
ERROR CUR	Excess current is being drawn by chip in the socket; supply voltage is cut off.	Verify that device is properly inserted, Use tests in 6.3.4 to check for internal shorts.
ERROR DAT	Short circuit was detected in device data lines prior to program function	The device is probably bad; it should be replaced.
ERROR ADD	Short circuit was detected in device address lines prior to program function	The device is probably bad; it should be replaced.
ERROR SAD	Mismatch between installed socket adapter and selected device type	Program function is terminated; replace socket adapter or change selected device type, as described in 6.3.1
ERROR SOC	Expected Address Latch Enable (ALE) signal not present	Single-chip microcomputer improperly installed or missing; correct this, and/or reselect device type (see 6.3.1).

INDEX



STECKER D
STECKER A
STECKER B
C1 QUARZ
LS 04 C2
LS 74 E2
G2
75 189 H2
75188 J1
8251 J2
Z80-PIO L2
Z80-PIO L3
Z80-CPU C3
LS 132 H3
TR 2 850899
GL1
GL2
GL3
GL4
GL5
GL6
C4 2200µ/40V
C5 2200µ/40V
C6 2200µ/40V
C7 2200µ/40V
ILQ 74 U1
LS 244 A4
LS 244 C4
HM 7611 E4
HM 7611 G4
ADR 0
nicht bestückt J4
2716 K4
2716 2532 2732 L4
2716 2532 2732 M4
P4 74LS74
8 6104 A5
7 6104 B5
6 6104 C5
5 6104 D5
1 6104 E5
2 6104 F5
3 6104 G5
4 6104 H5
J5 2114
K5 7603
STC
L1
T33
Kühlkörper
Molex 4-fach
LINE 110/220V
STE
18 6104 A6
17 6104 B6
16 6104 C6
15 6104 D6
11 6104 E6
12 6104 F6
13 6104 G6
14 6104 H6
J6 2114
LS 244 K6
DIP-SWITCH
Power Fail
U-Ausg
J-max
P1
P2
P3
N5
O5
O6
C2 2200µ/6V
C3 2200µ/6V
C1 4700µ 25V
D3
TR 1 850009 30 VA
TP1
TP2
TP3 GND
F
E
D
C
B
A
9
8
7
6
5
4
3
2
1
Ø
LS 32 H7
Z80-CTC J7
LS 74 L7
8279-5 H8
74 LS 138 J9
74 C 154 K9
D1
D2
16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
T1 - T21: ZTX 550
T31
T32
Loutstärke
P4
C20
C21
ENTER
REM
LOAD
CHK
PRO
OUT
IN
MOV
DEL
INS
SET
LOUDSPAEKER