INTERNATIONAL MICROSYSTEMS, INC.

OPERATING MANUAL

(916) - 885 - 7262

K+D Associates:
(301) - 636 - 1151

4/19/79

TABLE OF CONTENTS

TABLE OF CONTENTS, CONT'D.

INTERNATIONAL MICROSYSTEMS, INC.

IM 1010 OPERATING MANUAL

1.0 IM1010 SUMMARY

The IM1010 is a microprocessor-based universal PROM programming instrument. Easily changed personality modules allow the IM1010 to program a full range of programmable devices. Standard and optional interfaces allow for a wide range of programming system configurations.

1.1 STANDARD OPERATING FEATURES

IM1010 Mainframe

- Functions:

  INSERT, DELETE, MOVE, PROGRAM, BLANKCHECK, CHECKSUM, FILL, VERIFY, LIST, INVERT, NIBBLE SWAP, SEARCH, REMOTE LIST, REMOTE PROGRAM, Change Modules with Power On.

- RAM:

  32k bit standard; 128k bit RAM optional
  Selectable addressing for beginning and end of field

- Physical:

  14 digit alpha-numeric display
  Hexadecimal keypad

- Interface:

  Two (2) serial RS-232C ports, switch selectable to 20MA current loop
  Switch selectable baud rates 110 to 9600 2400

1.2 IM PERSONALITY MODULES

- High quality zero insertion/ extraction force sockets
- Adapters for generic families
- Backward insertion indicators on most modules
- Device select switches as required

1.3 CHANGING PERSONALITY MODULES

To change personality modules, the IM1010 should either be turned off or in the CHANGE MODULE mode. The module is changed by opening the top of the IM1010 and firmly grasping the personality module at the high end of the slope and applying pressure to slide the module back and out of the unit. See Figure 1 below.

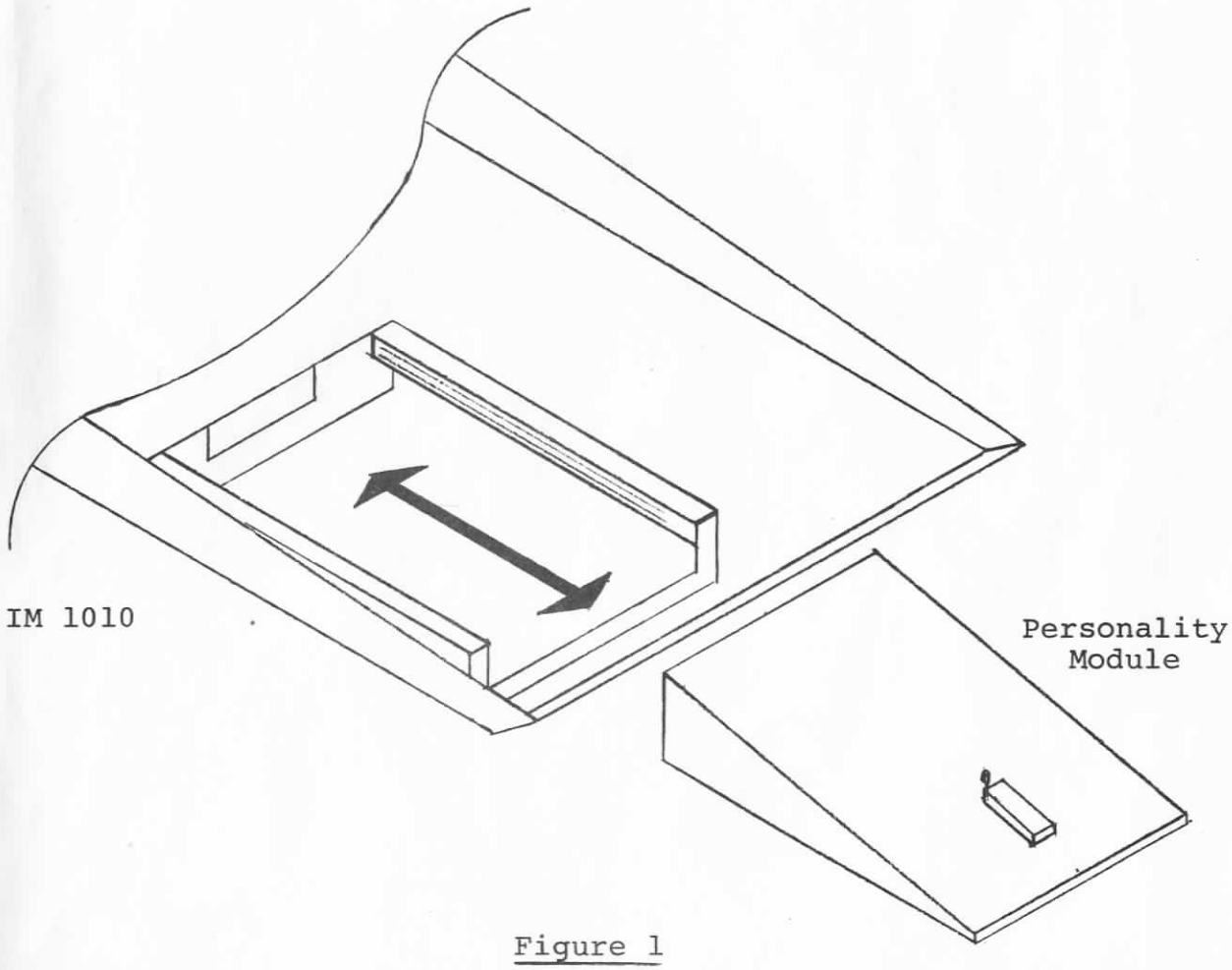


Figure 1

1.4 CHANGING ADAPTERS

Some IM personality modules require the use of a pinout Adapter. These adapters not only set the pinout for a particular size, but also tell the programmer the array size of the device to be programmed. Make sure the adapter is the correct one for the type of device in use. Check the IM Module Selection Guide if in doubt.

To insert an adapter, follow the procedure below:

1. Make sure pin one is in the upper left.
2. Insert the adapter in the zero insertion/ extraction force sockets on top of the personality module.
3. Make sure the adapter is seated correctly.
4. Close the contacts to hold the adapter in the socket securely.
5. Depress RESET on the IM1010.

See Figure 2, below.

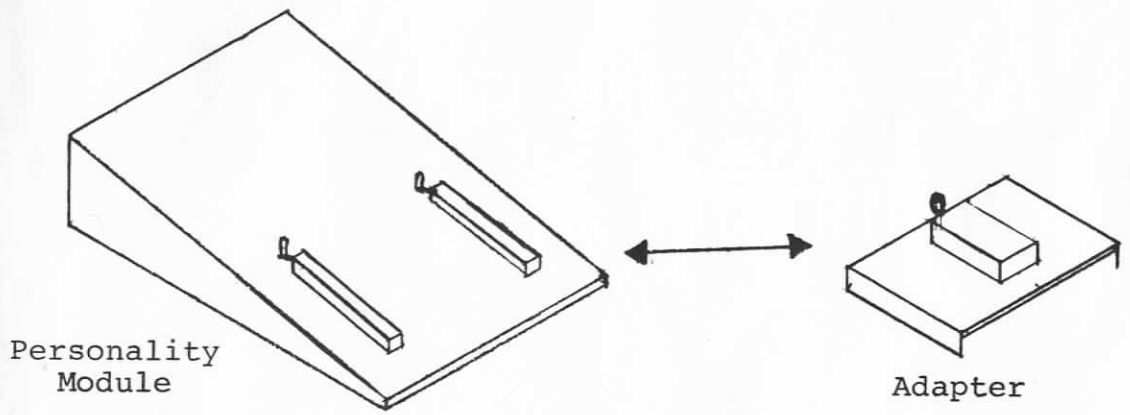


Figure 2

2.0 UNPACKING INSTRUCTIONS

Remove unit from shipping carton and place on bench. Loosen thumb screws on the front bottom of unit. Open and remove internal packing material. Inspect unit for shipping damage.

Look for:

1. Boards loose from supports
2. Cables loose from pins
3. IC packages loose from sockets
4. Stray packing material
5. Make sure power switch on back of unit is in the OFF position.

2.1 INSERTION OF MODULES

1. Open top of IM1010 unit.
2. Firmly slide module into unit making sure card edge goes all the way into the connector.
3. Turn unit on.
4. Depress RESET, display should read "-" which means that the IM1010 is ready to receive an instruction.
5. If the IM1010 display reads (E PLUG IN XXX), it means that the IM1010 does not recognize the module in the socket. Check for correct insertion of the module and/or adapter.

2.2 INITIAL PROGRAMMER CHECKOUT

Keys to be Depressed	Resulting Display	Function
1. RESET	-	Initialize module parameters
2. FILL, ENTER	- F DONE	Fill RAM to the size of the PROM with the ERASED condition
3. BLCK ENTER	- B ErASED Note: Lower case R is correct	Check the RAM to the size of the PROM for the ERASED condition
4. PROGRAM, ENTER	P 000 XX	Program the RAM starting at 000. Note: XX is the ERASED condition of the PROM.

Keys to be Depressed	Resulting Display	Function
5. 5A, ENTER	P 000 XX 5A	Puts 5A in Location 000
6. A5, ENTER	P 001 XX A5	Puts A5 in Location 001
7. RESET	-	Terminates previous function
8. VER, ENTER	V 000 5A XX	Compares RAM to PROM socket and flags discrepancy at address 000
9. ENTER	V 001 A5 XX	Increments address to the next location where an error occurs
10. ENTER	- V DONE	Incremented and found no more discrepancies and indicates function is done

If the above procedure checks out, proceed with manual. If problems occur, please call factory.

3.0 CARE OF PROMS AND PROM PROGRAMMERS

The use of PROMs and PROM programmers deserves extra care from the user than is normally warranted with many types of integrated circuits. This is due to the essential feature of PROMs, namely the destructive process which programs various locations in the PROM. Although PROMs program in many different ways, all share certain programming characteristics such as high current, high voltage and fast high energy pulses. The list below details a few tips on PROM care which may aid the user of PROMs from mistakenly blowing PROMs in the wrong way.

1. Clearly number all PROMs making sure pin one is clearly discernable.

2. Test all EPROMs after purchase for erasure and programmability. This is especially true of PROMs bought from hobby type suppliers.

3. The greatest PROM failure is caused by upside-down insertion in the programmer socket. Take extra care that PROMs are inserted correctly. The "Remove PROM" light on some IM personality modules indicates backward PROMs.

4. Never leave a PROM in the programmer socket while turning the programmer ON or OFF.

5. Make sure that UV erasable PROMs are correctly erased before use. Partially erased PROMs are a common failure with erasable type PROMs. We recommend the Ultra Violet Products erase lights which are available directly from International Microsystems. These lights have the correct wavelength and intensity to correctly erase UV eraseable PROMs.

3.1 PROM DIFFERENCES

Bipolar PROMs by different manufactures are often pin for pin compatible in operation. The programming method, however, is most often different. Another source of confusion in the PROM market is different manufacturers using the same part number. For example, the 2716 device made by Intel is significantly different from the 2716 device manufactured by Texas Instruments. When selecting PROMs for second sourcing, be sure to check the programming specifications before trying to program a device with a personality module designed for another manufacturer's device. Check IM's Module Selection Guide for compatibility or call the factory and ask for technical assistance.

3.2 FAILURE RATES

PROMs can have an initial failure rate of 5-15%. Especially high failure rates are seen on devices new to the market. However, if you believe that the failure rates that you experience with the IM1010 are too high, please don't hesitate to call the factory.

4.0 INTRODUCTION

The IM1010 PROM programmer is a versatile instrument which may be used in many different system configurations. Figure 3 on page 9 shows some typical systems.

4.1 In order to perform these various interface tasks, in addition to programming many different types of PROMs, the IM1010 incorporates a microcomputer. The block diagram of the IM1010 is seen in Figure 4, on page 10.

4.2 As seen in Figure 4, the PROM firmware for the CPU is segmented into three parts. Bank "0" contains the housekeeping and serial interface software. Bank "1" contains PROM driver software and Bank "2" contains special driver software such as the Intellec software option and the IC handler software option.

4.3 MAIN OPERATING MODES

The main operating modes are the keyboard mode, terminal mode and remote mode. The keyboard mode commands are initiated from the IM1010 keyboard. The terminal mode commands are entered via the serial terminal port on the back of the IM1010. Finally, the remote mode involves either the IM1010 keyboard or the terminal port, and also the IM1010 serial modem port. There are many similarities between the keyboard and the terminal modes. It is therefore suggested that new users of the IM1010 learn the keyboard functions before proceeding to the terminal functions.

4.4 KEYBOARD OPERATING MODES

The keyboard operator of the IM1010 interacts via 29 keys and a 14 digit gas discharge display.

4.5 DISPLAY MESSAGES

Although numerous types of messages are displayed, depending upon the operation being performed, there are two basic display messages which are always used:

1. A dash in left most position indicates opertion is completed and a new command line may be entered.

2. The first and second left most digits are used to display the function being performed.

Example: `- VP  DONE`

"-" indicates IM1010 ready for new command
"VP" indicates VERIFY PROM function was being performed

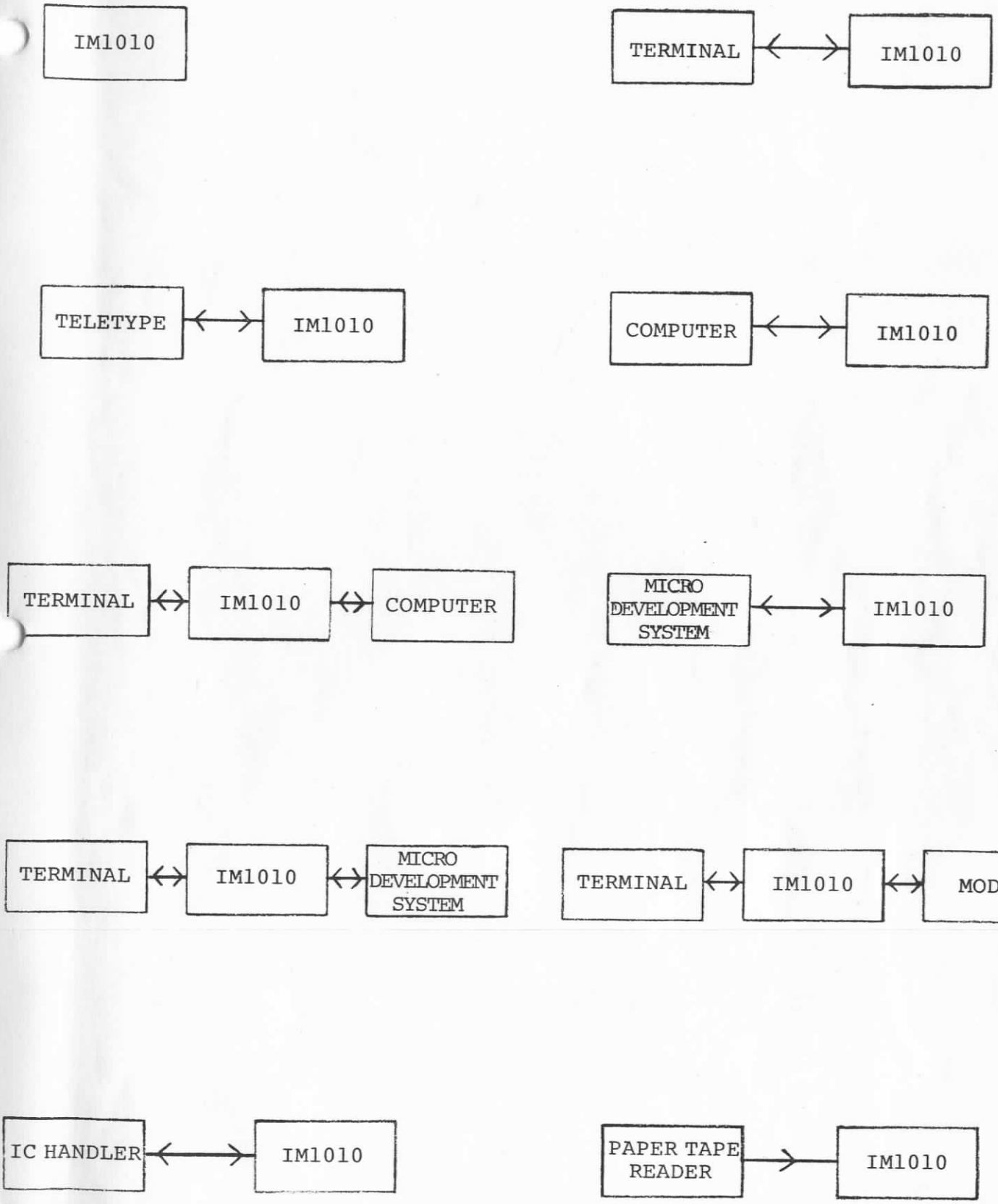

IM1010
TERMINAL
IM1010
TELETYPE
IM1010
COMPUTER
IM1010
TERMINAL
IM1010
COMPUTER
MICRO DEVELOPMENT SYSTEM
IM1010
TERMINAL
IM1010
MICRO DEVELOPMENT SYSTEM
TERMINAL
IM1010
MODEM
IC HANDLER
IM1010
PAPER TAPE READER
IM1010

Figure 3

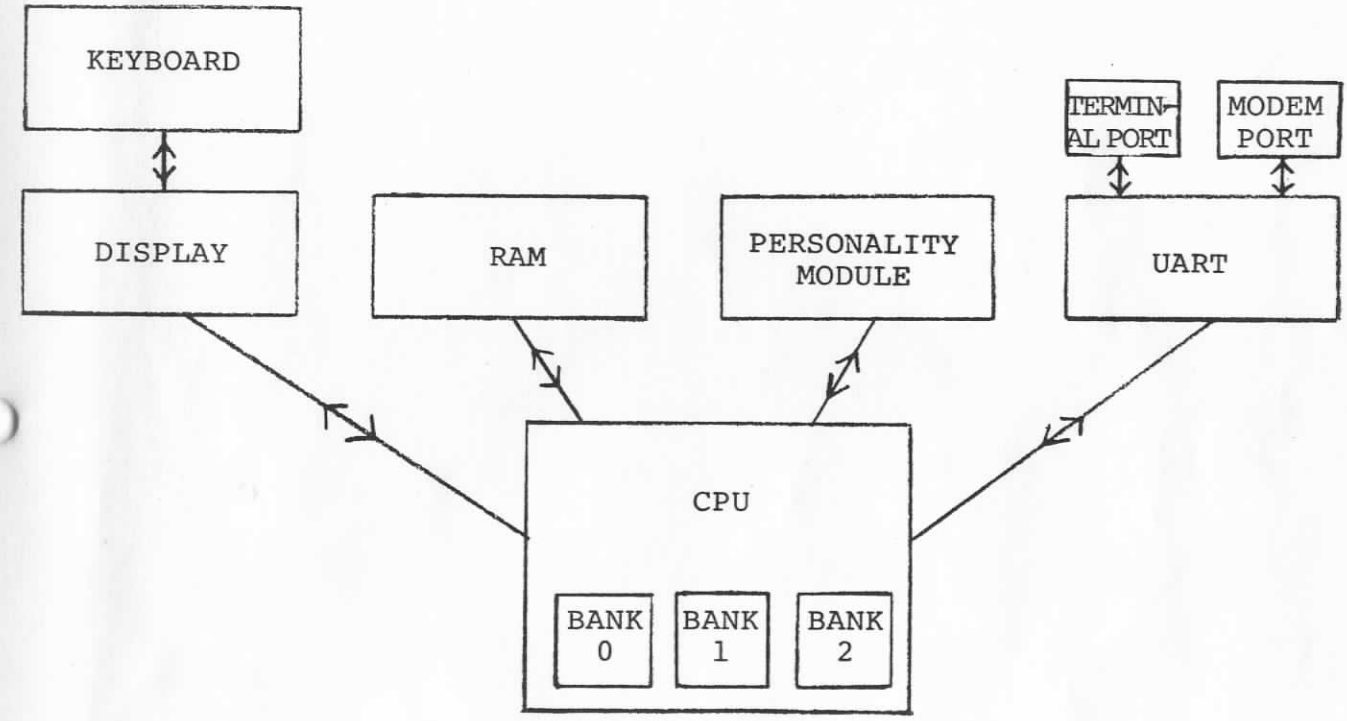

KEYBOARD
DISPLAY
RAM
PERSONALITY MODULE
TERMIN-AL PORT
MODEM PORT
UART
CPU
BANK 0
BANK 1
BANK 2

Figure 4

4.6 THE KEYBOARD

The IM1010 keyboard consists of 29 keys which are divided into three groups as shown in Figure 5 on page 12.

- Function Keys: PROM, MOVE, INSERT (INS), DELETE (DELE), FILL, PROGRAM (PROG), BLANKCHECK (BLCK), CHECKSUM (CSUM), VERIFY (VER), FUNCTION (FUNC)

The function keys are used during the entry of the command functions.

- Data Keys: (0 - F)

These 16 keys are used to enter the hexadecimal address and data, and also to select various functions when used in conjunction with the function key.

- Control Keys: (RESET, CLEAR, ENTER)

  RESET is used to exit from any function.
  CLEAR is used to clear the last entry made.
  ENTER is used to initiate a command or increment a command to the next address.

4.7 KEYBOARD FUNCTIONS

There are four types of functions which may be initiated from the IM1010 keyboard. These are:

1. RAM edit functions: Program, List, Move, Insert, Delete, Checksum, Blankcheck, Invert, Search, Nibble Swap

   Example: Checksum RAM from H"123" to H"234"
   User enters: CSUM 123 234 ENTER (4kx8 RAM)

2. PROM functions: LIST, BLANKCHECK, CHECKSUM

   Example: CHECKSUM PROM from H"123" to H"234"
   User enters: PROM 123 234 ENTER

3. Interaction between PROM and RAM functions: PROGRAM, VERIFY

   Example: Program PROM with RAM images from H"123" to H"234"
   User enters: PROG PROM 123 234 ENTER (4kx8 RAM)
   PROG PROM 0123 0234 ENTER (16kx8 RAM)

4. Special functions:

   Example: Change modules without affecting RAM data
   User enters: FUNC 04 ENTER

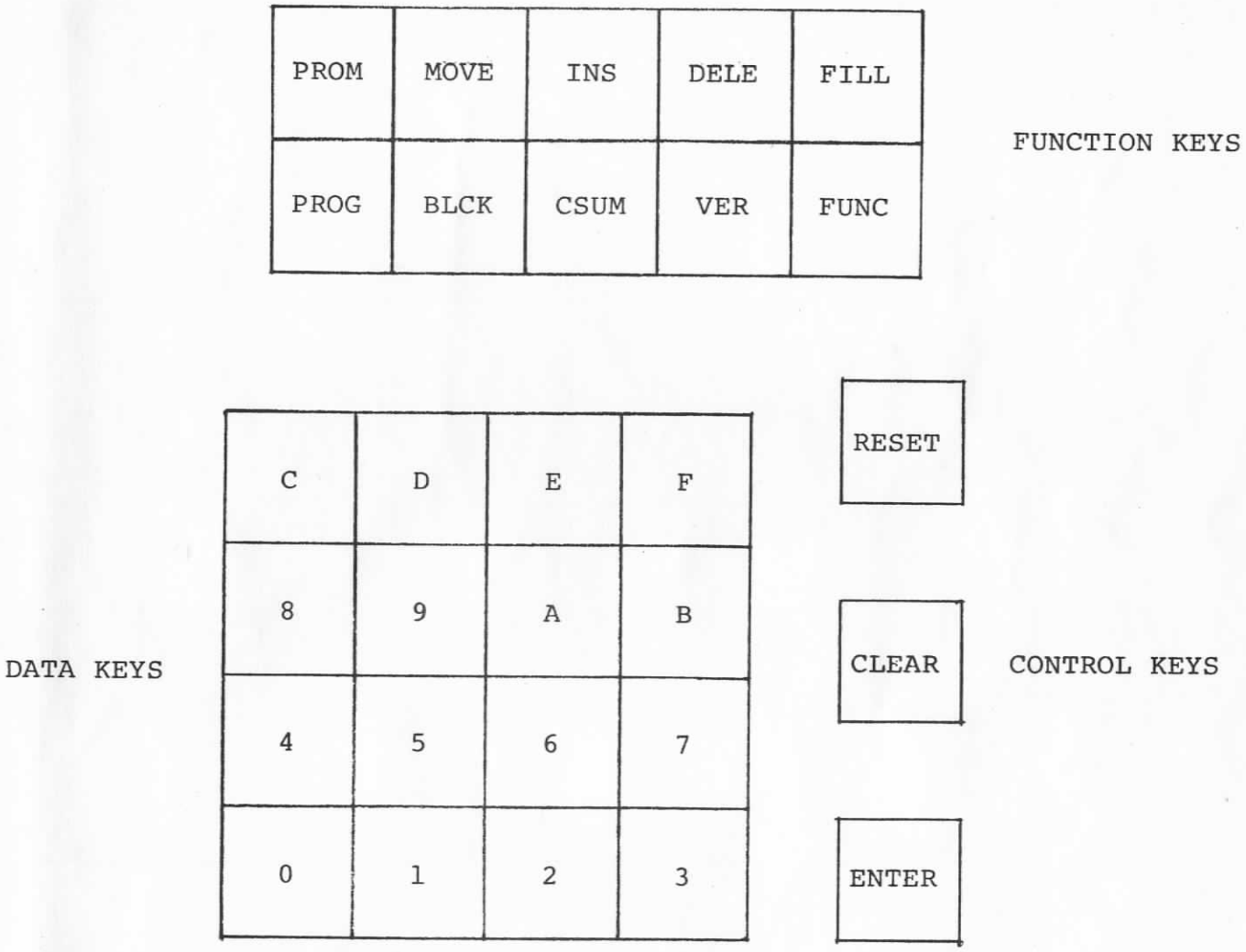

PROM
MOVE
INS
DELE
FILL
PROG
BLCK
CSUM
VER
FUNC
FUNCTION KEYS
C
D
E
F
8
9
A
B
4
5
6
7
0
1
2
3
DATA KEYS
RESET
CLEAR
ENTER
CONTROL KEYS

Figure 5

One important point which is demonstrated in the above examples using the Checksum function is the use of the PROM key. When the user wishes to perform a PROM function he/she must explicitly follow the entry of the desired function with the PROM key. This is not the case with RAM functions. Thus, unless the PROM key is used, the IM1010 assumes that the desired function is to be performed on the RAM.

In the following two sections there is a sample operating session at the keyboard and detailed descriptions of all the commands. We suggest that the new IM1010 user go through the sample session using the programmer.

4.8 TERMINAL MODE -- GENERAL CONSIDERATIONS

For those IM1010 operators who desire to use a terminal to interact with the programmer, there is the IM1010 terminal mode. Normally the user connects the terminal via the terminal port connector on the back of the IM1010. Section 7.4 on serial interfacing shows how the internal switches in the IM1010 may be set to the desired baud rate, etc.

Some important points to be emphasized are given below:

Half Duplex Mode: The terminal should be set for half duplex. That is, it should print its own entries.

ASCII Format: The IM1010 assumes that terminal commands are entered in 7-bit ASCII format and are in upper case.

Terminate Key: The ASCII character "#" (H"23") will generally cause early termination of a terminal function.

Command/Field Divider: The ASCII character ":" (H"3A") is used to divide the function command from address and data field in the terminal mode.

Clear Entry: The ASCII character "/" (H"2F") is used to clear digits on a terminal.

Enter: The ASCII character for carriage return (H"OD") corresponds to ENTER on the IM1010 keyboard.

Use of spaces between beginning and end addresses and data is recommended but not required. On functions that require more than 16 hex digits, do not use spaces.

4.9 EXAMPLES OF ASCII CONTROL CHARACTERS

```
#
>LR: 000 01F
ADDR  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F

 000 89  89  89  89  89  89  89  89  89  89  89 #89  89  89  89  89
>


FR: 000 005 12//34

>LR: 000   00F
ADDR  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F

 000 34  34  34  34  34  34  89  89  89  89  89  89  89  89  89  89
>
```

5.0 SAMPLE PROGRAMMING SESSION

Situation: User wants to program three 2708 PROMs from an assembly language listing and make changes to one of the PROMs. Note: If you are using a PROM other than the 2708 the following displays and entries may vary.

Put module in mainframe:

Key In:	Display	Function
1. RESET	-	Sets parameters for module in use and puts IM1010 in a "ready for instruction" state
2. FILL 000 FFF ENTER	F 000 FFF	Fills all of RAM with the Erased condition
3. PROG ENTER	P 000 FF	Initiates program RAM function from 000 to size of PROM. The IM1010 returns with first address and data currently in RAM.
4. Type in data. Follow each entry with ENTER to store data and increment address.		
5. After all data is entered, or RESET is depressed, list contents of RAM		
6. FUNC ENTER	L 000 XX	Lists contents of RAM from 000 to size of PROM. Note: XX is the data in the RAM.
7. Each time ENTER is depressed, the address will increment one (1) location.		
8. Or depress 0, 1, 2 or 3 on hexpad for slew rate control. Depressing ENTER will stop slew. Depress RESET to exit list mode.		
9. Place PROM in socket.		
10. To program PROM:		
11. PROG PROM ENTER	PP DONE = EN	Initiates program PROM function from 000 to size of PROM and waits for second ENTER as a safety precaution.
12. ENTER	PP 01	Programming cycle starts and counts off, after programming the IM1010 automatically goes into a verify cycle, if no errors occur, the display reads -VP DONE .

Key In:	Display	Function
13. To duplicate the PROM place a clean PROM in the socket and follow steps 11 and 12.		
14. To make some changes in the RAM and still save the original copy:		
15. MOVE 000 3FF 400 ENTER	M 000 3FF 400 -M DONE	Moves RAM data 000 to 3FF and puts it in at 400 Move is done creating two copies in the RAM.
16. To insert data in the second copy starting at 450:		
17. INS 450 7FF ENTER	I 450	Initiates INSERT function at 450 with top of block at 7FF
18. Enter data following each entry with ENTER to store data and increment address.		
19. To program a PROM from the second copy:		
20. RESET	-	Resets IM1010 to be ready for new function
21. PROG PROM 400 7FF ENTER	PP DONE = EN	Initiates program PROM function from 400 to 7FF. Waits for second ENTER as a safety precaution.
22. ENTER	PP 01	Programming cycle starts and counts off; after programming, the IM1010 automatically goes into a Verify cycle, if no errors occur, the display reads -VP DONE.
23. When programming cycle is complete, programming session is finished.		

6.0 DETAILED FUNCTION DESCRIPTION

This section contains a detailed description of the various functions that can be performed by the IM1010. The functions are in alphabetical order and contain the terminal formats, the IM1010 keyboard formats and some examples.

6.1 BLANKCHECK

Checks PROM or RAM over any field for erased condition.

Formats:

- Terminal

BP: (CR)	Checks PROM from 000 to size of PROM for the erased condition
BP: XXX YYY (CR)	Checks PROM from XXX to YYY for the erased condition
BR: (CR)	Checks RAM from 000 to size of PROM for the erased condition
BR: XXX YYY (CR)	Checks RAM from XXX to YYY for the erased condition

- IM1010 Keyboard

BLCK PROM ENTER	Checks PROM from 000 to size of PROM for the erased condition
BLCK PROM XXX YYY ENTER	Checks PROM from XXX to YYY for the erased condition
BLCK ENTER	Checks RAM from 000 to the size of the PROM for the erased condition
BLCK XXX YYY ENTER	Checks RAM from XXX to YYY for the erased condition

Notes:

1. When checking PROMs, the Y-X distance may not be larger than the PROM

Examples:

The following examples were done in the Terminal Mode of operation and show Blankchecking the RAM over two different fields; one with the erased condition, the other with data present. A listing of the RAM is given for clarity. Also shown is the format for blankchecking the PROM from 000 to the top.

```
LR: 000 01F
ADDR  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F

 000 FF  FF  FF  FF  FF  FF  FF  FF  FF  FF  FF  FF  FF  FF  FF  FF
 010 4B  66  4B  0D  4B  0D  4B  0D  40  00  40  00  4C  A2  4C  13
>

BR: 000 00F
ERASED
>

BR: 000 01F
NOT ERASED
>

BP:
ERASED
>
```

6.2 CHANGE MODULE

Prevents data in the RAM from changing while changing personality modules.

Formats:

- IM1010 Keyboard

FUNC 04 ENTER	Accesses change module function Personality module may be changed.
RESET	Sets parameters for new module and returns to main system.

Notes:

1. Display may light all segments but RAM will be undisturbed. Pushing RESET will clear display and return to main system.

6.3 CHECKSUM

Performs a straight 8-bit add on the data in the RAM or PROM,and displays the last 8-bits of the sum in hexadecimal.

Formats:

- Terminal

CP: (CR)	Checksums entire PROM
CP: XXX YYY (CR)	Checksums PROM from XXX to YYY
CR: (CR)	Checksums RAM from 000 to size of PROM
CR: XXX YYY (CR)	Checksums RAM from XXX to YYY

- IM1010 Keyboard

CSUM PROM ENTER	Checksums entire PROM
CSUM PROM XXX YYY ENTER	Checksums PROM from XXX to YYY
CSUM ENTER	Checksums RAM from 000 to size of PROM
CSUM XXX YYY ENTER	Checksums RAM from XXX to YYY

Notes:

1. When checksuming PROM, Y-X distance may not be larger than PROM.

Examples:

The following examples were done in the Terminal Mode of operation and show checksuming the RAM over two different fields. A listing of the RAM is given for clarity. Also shown is the format for checksuming a PROM from 000 to top of PROM.

```
LR: 000 01F
ADDR  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

 000 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
 010 4B 66 4B 0D 4B 0D 4B 0D 40 00 40 00 4C A2 4C 13
>

CR: 000 00F
CKSUM = F0
>

CR: 010 01F
CKSUM = 86
>

CP:
CKSUM = 00
>
```

6.4 CONTROL KEYS

- **CLEAR**

Terminal

/ — Entry of a "/" clears the last character entered from the terminal. "/" may be entered as often as needed to clear back to a desired character.

IM1010 Keyboard

CLEAR — Entry of a "CLEAR" clears the last character entered from the terminal. "CLEAR" may be entered as often as needed to clear back to a desired character. "CLEAR" also can be used for early termination of a program PROM cycle.

- **ENTER**

Terminal

(CR) — Used to enter data or initiate a function

IM1010 Keyboard

ENTER — Used to enter data or initiate a function

- **RESET**

Terminal

— Used for early termination of most functions from a terminal

IM1010 Keyboard

RESET — Used for early termination of all functions from the IM1010 keyboard

Examples:

The following examples were done in the Terminal Mode of operation and show each of the above in operation.

```
#
>LR: 000 01F
ADDR  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F

 000 89  89  89  89  89  89  89  89  89  89  89  89  89  89  89  89
>

#
>FR: 000 005 12//34

>LR: 000    00F
ADDR  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F

 000 34  34  34  34  34  34  89  89  89  89  89  89  89  89  89  89
>
```

6.5 DELETE

Deletes word or block in RAM and backfills with the erased condition.

Formats:

- Terminal

DR: XXX YYY ZZZ CR	Deletes RAM block XXX to YYY and moves block YYY+1 to ZZZ down to XXX backfilling from ZZZ with the erased condition

- IM1010 Keyboard

DELE XXX YYY ZZZ ENTER	Deletes RAM block XXX to YYY and moves block YYY+1 to ZZZ down to XXX backfilling from ZZZ with the erased condition

Notes:

1. This is a destructive move. ZZZ may not be in the range of XXX to YYY.

Examples:

The following example was done in the Terminal Mode of operation and shows the format for a DELETE instruction and a RAM image before and after the DELETE.

```
LR:000 02F
ADDR  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F

 000 44  44  44  11  11  11  11  11  11  11  22  22  22  22  22  22
 010 22  22  22  22  22  22  00  00  00  00  00  00  00  00  00  00
 020 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
>


DR: 001 008 014

>


LR: 000 02F
ADDR  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F

 000 44  11  22  22  22  22  22  22  22  22  22  22  22  FF  FF  FF
 010 FF  FF  FF  FF  FF  22  00  00  00  00  00  00  00  00  00  00
 020 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
>
```

6.6 FILL

Fills RAM with any specified character over any field

Formats:

- Terminal

F: (CR)	Fills RAM from 000 to size of PROM with the erased condition
F: XXX YYY (CR)	Fills RAM from XXX to YYY with the erased condition
F: XXX YYY Z (CR)	Fills RAM from XXX to YYY with Z for 4-bit PROMs
F: XXX YYY ZZ (CR)	Fills RAM from XXX to YYY with ZZ for 8-bit PROMs

- IM1010 Keyboard

FILL ENTER	Fills RAM from 000 to size of PROM with the erased condition
FILL XXX YYY ENTER	Fills RAM from XXX to YYY with the erased condition
FILL XXX YYY Z ENTER	Fills RAM from XXX to YYY with Z for 4-bit PROMs
FILL XXX YYY ZZ ENTER	Fills RAM from XXX to YYY with ZZ for 8-bit PROMs

Examples:

The following examples were done in the Terminal Mode of operation and show some formats for the FILL instruction, and before and after RAM images.

```
LR: 000 010/F
ADDR  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F

 000 FF  33  37  3F  F1  11  77  73  E8  B8  D8  C8  68  A8  78  58
 010 4B  66  4B  0D  4B  0D  4B  0D  40  00  40  00  4C  A2  4C  13
>

F:005 00B

>

LR: 000 01F
ADDR  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F

 000 FF  33  37  3F  F1  FF  FF  FF  FF  FF  FF  FF  68  A8  78  58
 010 4B  66  4B  0D  4B  0D  4B  0D  40  00  40  00  4C  A2  4C  13
>

F: 006 009 55

>

LR: 000 01F
ADDR  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F

 000 FF  33  37  3F  F1  FF  55  55  55  55  FF  FF  68  A8  78  58
 010 4B  66  4B  0D  4B  0D  4B  0D  40  00  40  00  4C  A2  4C  13
>
```

6.7 INSERT

Inserts a word at a time into RAM and moves rest of data up one location at a time to a specified top; data above specified top is not affected.

Formats:

- Terminal

  I: XXX YYY (CR) — Insert data starting at XXX with top at YYY

- IM1010 Keyboard

  INS XXX YYY ENTER — Insert data starting at XXX with top at YYY

Notes:

1. Address may be redirected in the terminal mode

Examples:

The following example was done in the Terminal Mode of operation and shows the INSERT function with a redirected address, and before and after RAM images.

```
LR: 000 050
ADDR  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

 000 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
 010 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
 020 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33
 030 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44
 040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 050 00
>

I: 014 03F

 014 99
 015 88
 016 77
 017 66
 018 007 011
 007 AA
 008 BB
 009 #
>

LR: 000 050
ADDR  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

 000 11 11 11 11 11 11 11 AA BB 11 11 11 11 11 11 11
 010 11 11 22 22 99 88 77 66 22 22 22 22 22 22 22 22
 020 22 22 22 22 33 33 33 33 33 33 33 33 33 33 33 33
 030 33 33 33 33 44 44 44 44 44 44 44 44 44 44 44 44
 040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 050 00
>
```

6.8 INVERT

Inverts data in RAM over any field

Formats:

- Terminal

H 03 (CR)	Accesses INVERT function
(CR)	Inverts data in RAM from 000 to size of PROM
XXX YYY (CR)	Inverts data in RAM from XXX to YYY

- IM1010 Keyboard

FUNC 03 ENTER	Accesses INVERT function
ENTER	Inverts data in RAM from 000 to size of PROM
XXX YYY ENTER	Inverts data in RAM from XXX to YYY

Examples:

The following example was done in the Terminal Mode of operation and shows the INVERT function, and before and after RAM images.

```
FR: 000 01F 55

>

LR: 000 01F
ADDR  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

 000 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
 010 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
>

H: 03

INVERT  03
>003 00E
DONE
>

LR: 000 01F
ADDR  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

 000 55 55 55 AA AA AA AA AA AA AA AA AA AA AA AA 55
 010 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55
>
```

6.9 LIST

Lists data in RAM or PROM over any field. In the terminal mode, two types of listings are available: one outputs the data in a format of sixteen addresses across the page, the other outputs the data in a serial string in a format recognized by the IM1010 input mode.

Formats:

1. Terminal

LP: (CR)	Lists all data in PROM
LP: XXX YYY (CR)	Lists data in PROM from XXX to YYY
LR: (CR)	Lists data in RAM from 000 to size of PROM
LR: XXX YYY (CR)	Lists data in RAM from XXX to YYY

2. Terminal

HP: (CR)	Lists PROM to terminal in IM standard format
HP: XXX YYY (CR)	Lists PROM from XXX to YYY to terminal in IM standard format
HR: (CR)	Lists RAM from 000 to size of PROM to a terminal in the IM standard format
HR: XXX YYY (CR)	Lists RAM from XXX to YYY to terminal in IM standard format

1. Keyboard

FUNC PROM ENTER	Lists entire PROM
FUNC PROM XXX YYY ENTER	Lists PROM from XXX to YYY
FUNC ENTER	Lists RAM from 000 to size of PROM
FUNC XXX YYY ENTER	Lists RAM from XXX to YYY

Notes:

1. Terminal mode -- LIST function may be terminated early by entry of a "#".
2. IM1010 Keyboard -- LIST may be slew controlled by pushing any key, 0 through 3. Pushing ENTER stops the slew.
3. When listing a PROM, the Y-X distance must be less than the size of the PROM.

Examples:

1. This example shows listing from the terminal of the PROM from 100 to 115.

```
#
>LP: 100 115


ADDR  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F

 100 FF  FF  FF  FF  FF  FF  FF  FF  FF  FF  FF  FF  FF  FF  FF  FF
 110 FF  FF  FF  FF  FF  FF
>
```

2. This example shows listing from the terminal of RAM data 100 to 115.

```
LR

#
>LR: 100 115


ADDR  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F

 100 A3  14  0B  F8  14  1B  F8  14  1F  48  5E  58  5E  F0  B8  24
 110 00  59  1B  56  C7  54
>
```

6.10 MOVE

Moves data in PROM to RAM or RAM to RAM

Formats:

- Terminal

MP: (CR)	Moves data in PROM from 000 to size of PROM to RAM
MP: XXX YYY (CR)	Moves data in PROM from XXX to YYY to RAM
MR: XXX YYY ZZZ	Moves data in RAM from XXX to YYY and puts it in, starting at ZZZ

- IM1010 Keyboard

MOVE PROM ENTER	Moves data in PROM from 000 to size of PROM and puts it in RAM
MOVE PROM XXX YYY ENTER	Moves data in PROM from XXX to YYY and puts it in RAM
MOVE XXX YYY ZZZ	Moves data in RAM from XXX to YYY and puts it in starting at ZZZ

Notes:

1. MOVE PROM XXX to YYY function allows for Y-X distance to be greater than PROM size. This allows for multiple copies of PROM to be placed in RAM.
2. ZZZ may not be within the range of XXX to YYY.

Examples:

The following example was done in the Terminal Mode of operation and shows the format for a MOVE instruction and a RAM image before and after the move.

```
>LR:000050
ADDR  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

 000 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
 010 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
 020 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33
 030 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44
 040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 050 00
>

MR:023 029 003

>LR: 000 050
ADDR  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

 000 11 11 11 33 33 33 33 33 33 33 11 11 11 11 11 11
 010 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
 020 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33
 030 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44
 040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 050 00
>
```

6.11 NIBBLE SWAP

Swaps nibbles (4-bit words) of byte data

Formats:

- Terminal

H: 02 (CR)	Accesses Nibble Swap
(CR)	Swaps nibbles from 000 to size of PROM
XXX YYY (CR)	Swaps nibbles from XXX to YYY

- IM1010 Keyboard

FUNC 02 ENTER	Accesses Nibble Swap
ENTER	Swaps nibbles from 000 to size of PROM
XXX YYY ENTER	Swaps nibbles from XXX to YYY

Notes:

1. Nibble Swap is a RAM function only.

Examples:

The following example was done in the Terminal Mode and shows the Nibble Swap function with a before and after RAM image.

```
LR: 000 01F
ADDR  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

 000 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
 010 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
>
```

```
H:02

SWAP  02
>005 00D
DONE
>
```

```
LR: 000 01F
ADDR  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

 000 12 12 12 12 12 21 21 21 21 21 21 21 21 21 12 12
 010 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
>
```

6.12 PROGRAM

Programs PROM or RAM

Formats:

- Terminal

PP: (CR)(CR)	Programs PROM with RAM data from 000 to top of PROM
PP: XXX YYY (CR)(CR)	Programs PROM with RAM data from XXX to YYY
PR: (CR)	Programs RAM from 000 to size of PROM
PR: XXX YYY (CR)	Programs RAM from XXX to YYY

- IM1010 Keyboard

PROG PROM ENTER ENTER	Programs PROM with RAM data from 000 to top of PROM
PROG PROM XXX YYY ENTER ENTER	Programs PROM with RAM data from XXX to YYY
PROG ENTER	Programs RAM from 000 to size of PROM
PROG XXX YYY ENTER	Programs RAM from XXX to YYY

Notes:

1. When programming a PROM, the Y-X distance must be less than or equal to the size of the PROM.
2. PROGRAM PROM functions automatically enter a VERIFY mode after programming.
3. PROGRAM PROM cycle will stop is error is detected. Push (CR) or ENTER to continue.
4. PROGRAM RAM function is subject to redirection in the terminal mode.

Examples:

The following examples were done in the Terminal Mode and show the PROGRAM PROM function over the entire PROM and over a field. The PROGRAM RAM function is shown over a field with redirected addressing.

```
#
>PP:

DONE=CR
ADDR RAM PROM


ADDR RAM PROM

 000 56  FF
#001 35  FF
  002 37  FF
>


#
>PP: 400 7FF

DONE=CR
ADDR RAM PROM
#

ADDR RAM PROM

>



#
>PR: 123 456

ADDR RAM NEW

 123 2A  34
 124 88
 125 57
 126 EF  45/4
 127 58  BD
 128 89  34
 129 58  234567
ADDR RAM NEW

 234 FD  33
 235 C0  22
 236 52  #
>
```

6.13 REMOTE LIST

Outputs data in the IM standing format to the terminal port and the modem port.

Formats:

- IM1010 Keyboard

FUNC 00 ENTER	Accesses Remote List function
PROM ENTER	Dumps PROM from 000 to top of PROM
PROM XXX YYY ENTER	Dumps PROM from XXX to YYY
ENTER	Dumps RAM from 000 to size of PROM
XXX YYY ENTER	Dumps RAM from XXX to YYY

Notes:

1. This function is only available from the IM1010 Keyboard.
2. A SOF character ":" and EOF character "/" are added.
3. A "T" appears on the display upon correct entry.

Examples:

The following example shows operation of the Remote List function from the IM1010 Keyboard, and a sample of the data format that is output to a terminal or other device.

Entry From IM1010 Keyboard:

```
FUNC 00 ENTER

120 125
ENTER
```

Terminal Or Other Device Receives:

```
:

 120 FF

 121 FF

 122 FF

 123 FF

 124 FF

 125 FF
/
```

6.14 REMOTE PROGRAM

Allows for input of data in the IM standard format

Entry Formats:

- Terminal

H: 01 (CR)	Accesses Remote Program function and takes IM1010 off line. User must key in ENTER from IM1010 Keyboard to continue.

- IM1010 Keyboard

FUNC 01 ENTER	Accesses Remote Program
ENTER	Initiates Remote Program

Examples:

The following example shows operation of the Remote Program function from the IM1010 Keyboard and an example of the data format that the IM1010 is waiting to be input.

Example from IM1010 Keyboard:

```
FUNC 01 ENTER
ENTER
```

IM1010 is now waiting for data entry in the following format.

```
:                   Start of File Character
XXX YY              Three Hex Address Digits  Two Or One Hex Data Digits CR
XXX YY              More Address And Data
.
.
.
.
XXX YY
/
                    End Of File Character
```

6.15 SEARCH

Searches RAM or PROM for Nibble, Byte, or 2-Byte match.

Formats:

- Terminal

SP: Z (CR)	Searches 4-bit PROM from 000 to top of PROM for Z
SP: Z1Z2	Searches 8-bit PROM from 000 to top of PROM for Z1Z2
SP: Z1Z2Z3Z4 (CR)	Searches 8-bit PROM from 000 to top of PROM for Z1Z2Z3Z4
SP: XXX YYY Z (CR)	Searches 4-bit PROM from XXX to YYY for Z
SP: XXX YYY Z1Z2 (CR)	Searches 8-bit PROM from XXX to YYY for Z1Z2
SP: XXX YYY Z1Z2Z3Z4 (CR)	Searches 8-bit PROM from XXX to YYY for Z1Z2Z3Z4
SR: Z (CR)	Searches RAM over size of PROM for Z match (4-bit PROM module)
SR: Z1Z2 (CR)	Searches RAM over size of PROM for Z1Z2 match
SR: Z1Z2Z3Z4 (CR)	Searches RAM over size of PROM for 2 byte match to Z1Z2Z3Z4
SR: XXX YYY Z (CR)	Searches RAM from XXX to YYY for Z (4-bit PROM module)
SR: XXX YYY Z1Z2 (CR)	Searches RAM from XXX to YYY for Z1Z2
SR: XXX YYY Z1Z2Z3Z4 (CR)	Searches RAM from XXX to YYY for Z1Z2Z3Z4

- IM1010 Keyboard

FUNC 05 ENTER	Initiates Search function. Display shows H SEARCH 05.
PROM Z ENTER	Searches PROM from 000 to size of PROM for Z
PROM Z1Z2 ENTER	Searches PROM from 000 to size of PROM for Z1Z2
PROM Z1Z2Z3 Z4 ENTER	Searches PROM from 000 to size of PROM for Z1Z2Z3Z4
PROM XXX YYY Z ENTER	Searches PROM from XXX to YYY for Z
PROM XXX YYY Z1Z2 ENTER	Searches PROM from XXX to YYY for Z1Z2

IM1010 Keyboard, cont'd.

Z ENTER	Searches RAM from 000 to size of PROM for Z
Z1Z2 ENTER	Searches RAM from 000 to size of PROM for Z1Z2
Z1Z2Z3 Z4 ENTER	Searches RAM from 000 to size of PROM for Z1Z2Z3Z4
XXX YYY Z ENTER	Searches RAM from XXX to YYY for Z
XXX YYY Z1Z2 ENTER	Searches RAM from XXX to YYY for Z1Z2

Notes:

1. This function may be terminated in process from a terminal entry of a "#".

Examples:

The following examples were done in the Terminal Mode of operation and show respectively, 1) Search RAM over size of PROM for a 1-byte match; 2) Search RAM over a field for a 1-byte match; 3) Search RAM over size of PROM for a 2-byte match; and 4) Search RAM over a field for a 2-byte match.

```
#
>LR: 000 01F
ADDR  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

 000 00 22 00 22 33 00 00 00 00 00 00 00 00 00 00 00
 010 00 00 00 00 22 00 33 00 22 33 00 00 00 00 00 00
>

SR: 22
ADDR

 001
 003
 014
 018
>

SR: 000 00F 22
ADDR

 001
 003
>

SR: 2233
ADDR

 003
 018
>

SR:000 00F 2233
ADDR

 003
>
```

6.16 VERIFY

Compares PROM data to RAM data and displays discrepancies

Formats:

- Terminal

VP: (CR)	Verifies RAM from 000 to top of PROM against PROM
VP: XXX YYY (CR)	Verifies RAM from XXX to YYY against PROM
VR: (CR)	Verifies RAM from 000 to size of PROM against PROM
VR: XXX YYY (CR)	Verifies RAM from XXX to YYY against PROM

- IM1010 Keyboard

VER PROM ENTER	Verifies RAM from 000 to top of PROM against PROM
VER PROM XXX YYY ENTER	Verifies RAM from XXX to YYY against PROM
VER ENTER	Verifies RAM from 000 to size of PROM against PROM
VER XXX YYY ENTER	Verifies RAM from XXX to YYY against PROM

Notes:

1. In VERIFY RAM mode, if distance Y-X is greater than size of PROM, certain RAM data will be compared more than once to PROM data.
2. In the terminal mode, VERIFY may be terminated early by entry of a "#".
3. In VERIFY PROM mode, Y-X distance may not be larger than size of PROM.

Examples:

The following examples were done in the Terminal Mode of operation and show respectively, 1) VERIFY PROM against RAM over the size of the PROM; 2) VERIFY RAM against PROM over the size of the PROM; 3) VERIFY RAM against PROM over a field. These examples also show the use of the "#" character for early termination of a function.

```
#
>VP:

ADDR RAM PROM

 000 EC  FF
 010 30  FF
 011 F2  FF
 012 B2  FF
 #013 B2  FF
>
```

```
#
>VR:

ADDR RAM PROM

 000 EC  FF
 010 30  FF
 011 F2 # FF
>
```

```
#
>VR: 130 135

ADDR RAM PROM

 130 FB  FF
 131 B3  FF
 132 B2  FF
 133 B3  FF
 134 B2  FF
 135 B3  FF
>
```

7.0 COMPUTER INTERFACES

Interfacing the IM1010 to a computer can be done either through the terminal port or the modem port. Three common types of configurations are shown in Figure 6, on page 39. It should be noted that none of these modes prohibit the use of the IM1010 keyboard mode.

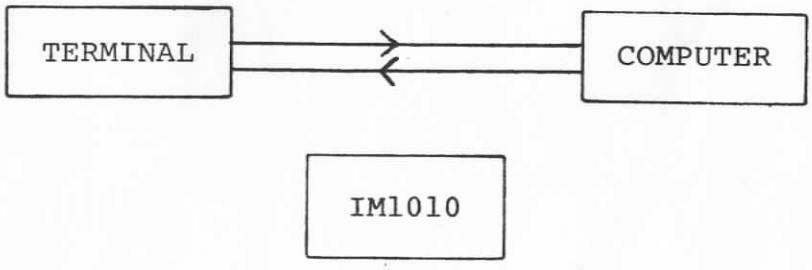
7.1 TYPE 1: COMPUTER EMULATION OF TERMINAL MODE

In this configuration, the user may have access to all the terminal mode features of the IM1010 by having the microcomputer either issue its own set of commands to the IM1010, or act as an in-between hand/shake device between the user and the programmer.

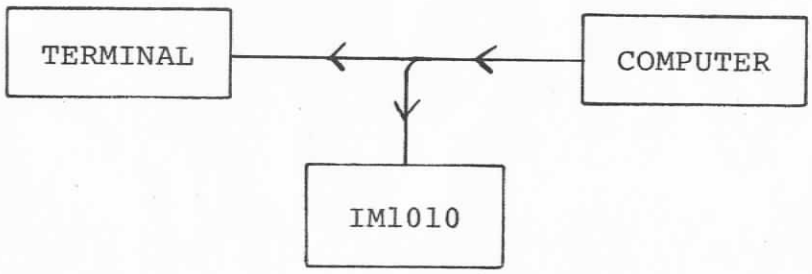
7.2 TYPE 2: DAISY CHAIN OPERATION OF THE IM1010

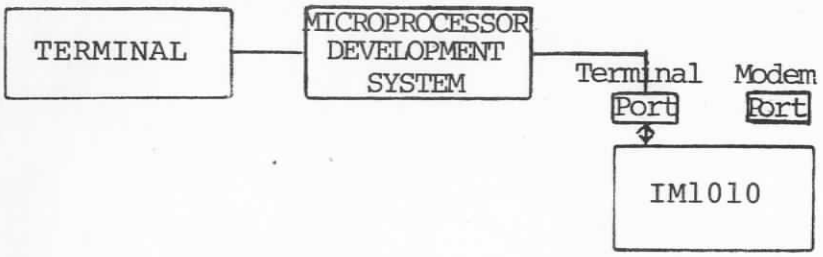
In this configuration, the user can perform the following functions without having to use two RS-232 ports on the computer or development system.

1. Place the IM1010 into the Remode Mode which takes the programmer off line and allows normal communication between the terminal and host computer.

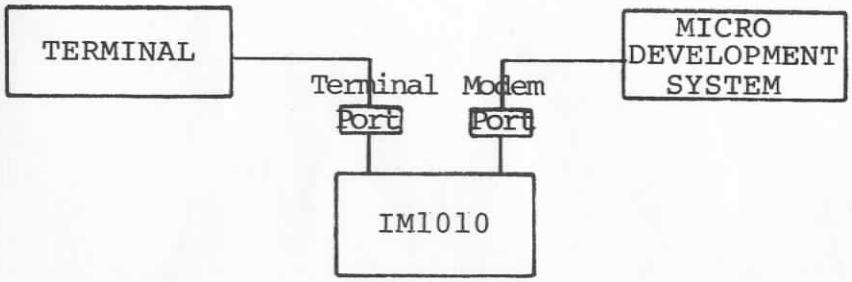


2. Place the programmer in the Remote Program Mode and cause the programmer and the terminal to receive a file transfer containing PROM data to be loaded into the IM1010 RAM.

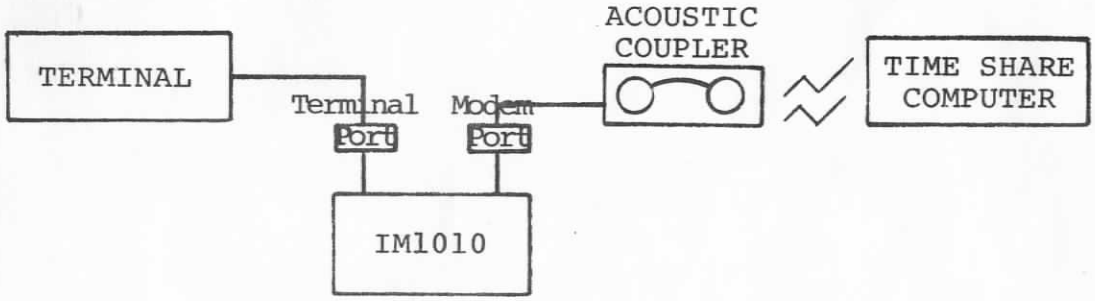




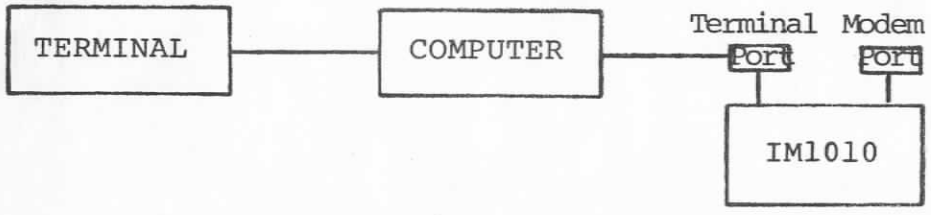
TYPE 1



or



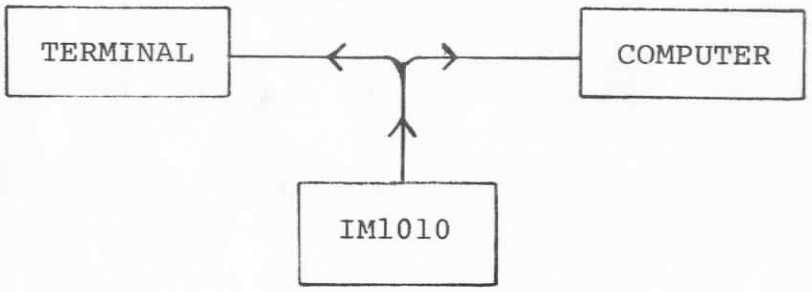
TYPE 2



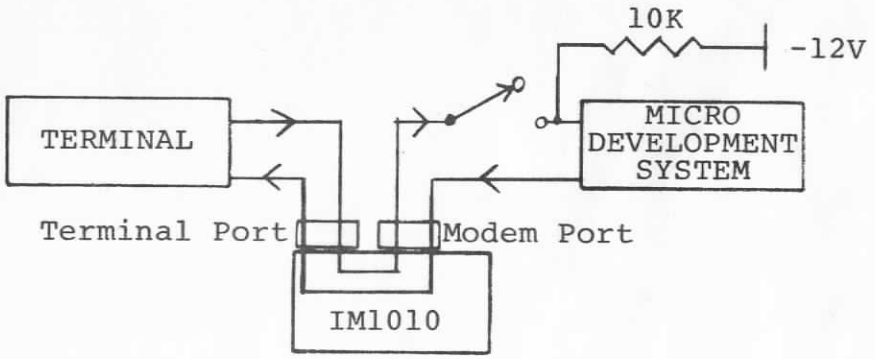
TYPE 3

Figure 6

3. Place the IM1010 in the Remote List Mode and cause a file transfer from the IM1010 to the host computer and the terminal.

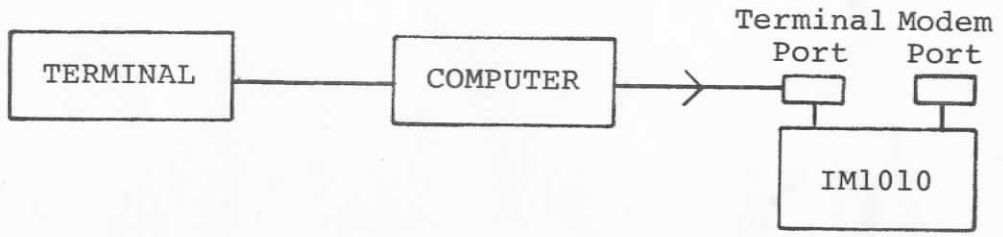


4. Using a user supplied switch on the data transmitted line to the host computer, the user may initiate all terminal functions of the IM1010 without sending data to the host computer. Note, this switch can be provided as an option to the IM1010.

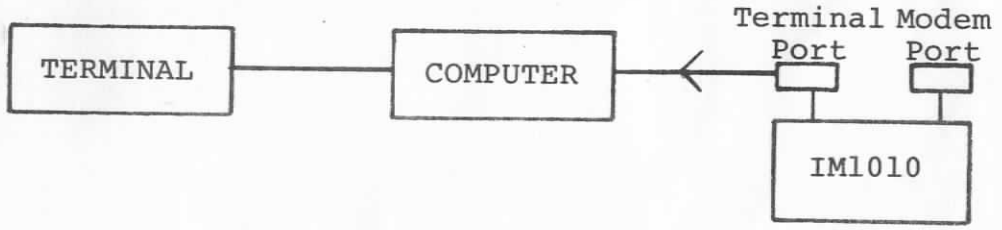


7.3 TYPE 3: SIMPLE FILE DUMP BETWEEN PROGRAMMER AND HOST COMPUTER

This mode is a special case of Type 1. The user places the IM1010 in a Remote Program Mode via the IM1010 keyboard and then using the user's terminal, directs the host computer to dump a data file to the IM1010. Alternately, the user may instruct the host computer to receive a data file from the IM1010 and then initiate this transfer from the keyboard of the IM1010.



or



OTHER PERIFERAL INTERFACES

Almost any serial ASCII instrument may be interfaced to the IM1010 in a manner similar to the ones described under Computer Interface. Several special systems in which the IM1010 has been used are seen in Figure 7, below.

The Engineering Department at IM will gladly assist you in solving your interface requirements for these applications or other specialized systems.

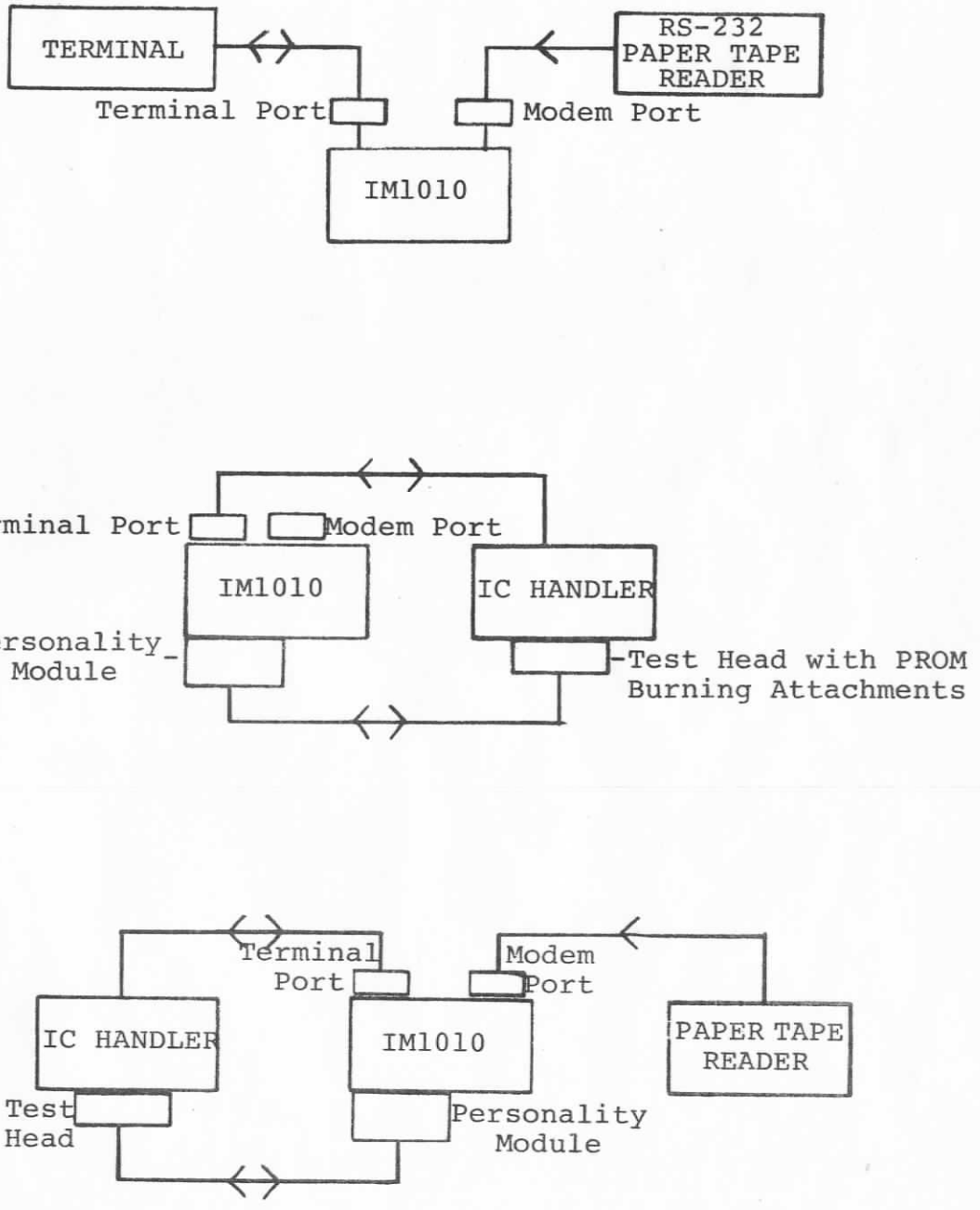


Figure 7

7.4 SERIAL INTERFACING

The IM1010 has two serial interfaces designated the terminal port and modem port. The terminal port provides a current loop and RS-232C interface. The modem port supports only an RS232C interface. The following paragraphs detail the use of these two ports.

7.5 CHARACTER ENCODING FORMATS

The serial encoding formats which are supported are seven level ASCII data and 8-bit binary. The ASCII format is normally used for terminal and computer interfacing. The standard software supports upper case letters, decimal numbers and certain control characters, (SPACE, /, :, #). Lower case characters will not be recognized.

7.6 STANDARD RS-232 PIN ASSIGNMENTS

Due to the confusion surrounding the RS-232C standard, a short review is applicable. The RS-232C standard recognizes two types of devices, a terminal and a data set. Figure 8, below, shows a typical connection. A brief description of these lines and their use follows.

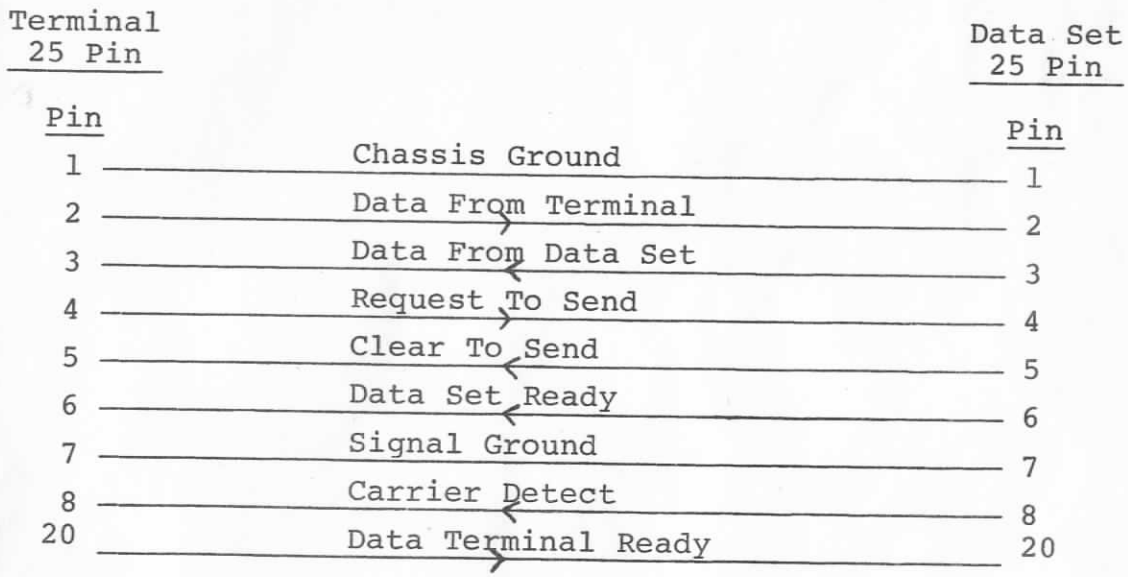


Figure 8

Grounds:

Pin 1: Chassis Ground
Pin 7: Signal Ground

Data Transmission:

Pin 2: Data from Terminal
Pin 3: Data from Data Set

On Line Status:

Pin 20: The Data Terminal Ready (DTR) line is pulled active by the terminal (> + 3V) to indicate that it is on line.

Pin 6: The Data Set Ready (DSR) line is pulled active by the data set to indicate that it is on line.

Pin 8: The Carrier Detect Line (DCD) is pulled active by the data set to indicate that a carrier signal is available.

Hand Shake Status:

Pin 4: The Request To Send (RTS) line is pulled active by the terminal to indicate it is ready to send data to the data set.

Pin 5: The Clear To Send (CTS) line is pulled active by the data set to signal the terminal that the terminal may initiate a data transfer. This line may only go high in response to an active RTS from the terminal.

The ground lines (Pin 1 and 7) should always be connected. The data lines (Pin 2 and 3) also must be connected for bi-directional data transfers. The status lines may or may not be needed depending on the periferal device. The IM1010 terminal and modem ports do not require any status lines. However, the necessary return status lines are always pulled active. Figure 9, below, shows the conditions on these two ports.

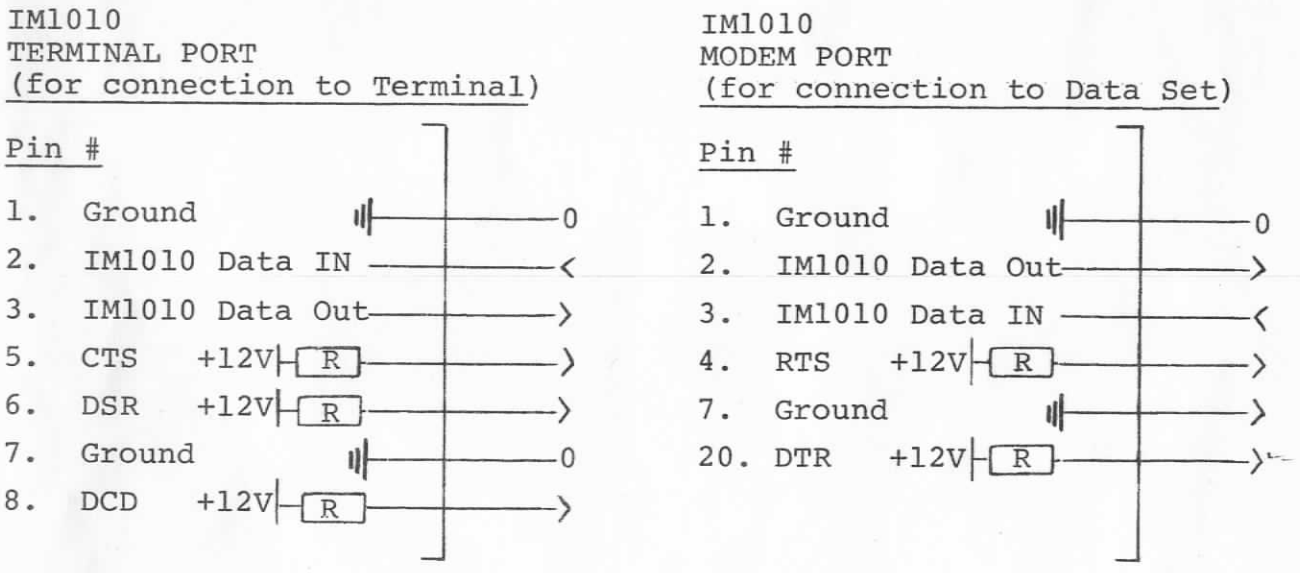


Figure 9

7.7 The connector on the Terminal Port is a female connector, and the connector on the Modem Port is a male connector. This allows the user who is daisy-chaining the programmer from a terminal to a computer to easily remove the IM1010 from the terminal to computer connection. This is shown in Figure 10, below.

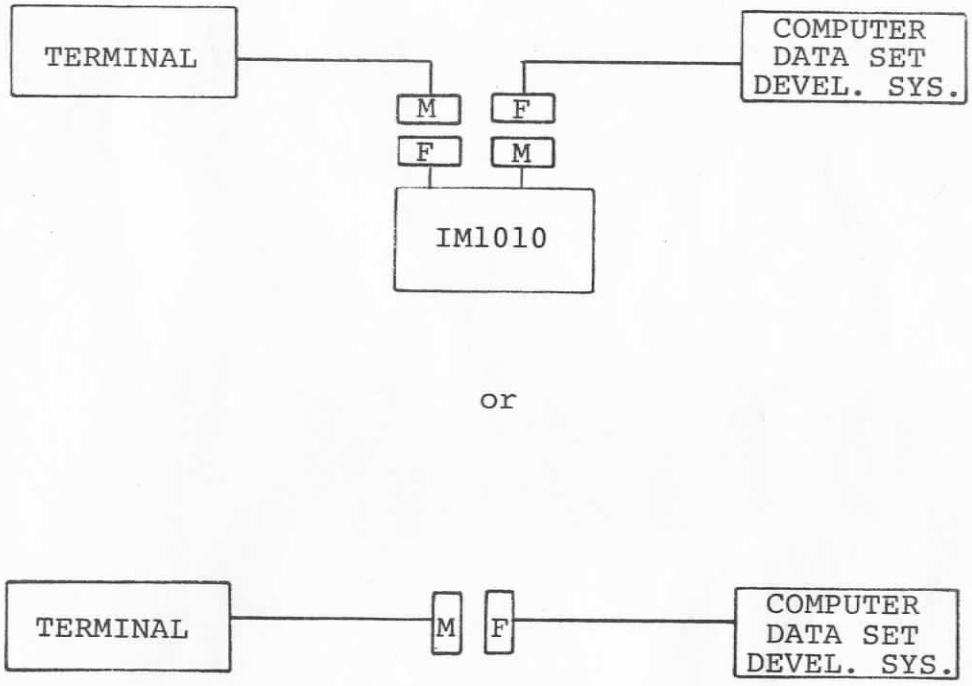


Figure 10

7.8 CURRENT LOOP INTERFACE

A 20MA current loop interface is provided on the terminal port and is shown in Figure 11, on page 46. This is primarily meant for a TTY. Although not shown in Figure 11, a 300 baud filter is incorporated on the UART interface board. If the user wishes to transmit current loop data at a higher rate than 300 baud, then C4 on the UART board must be removed. Consult the UART schematic for further aid.

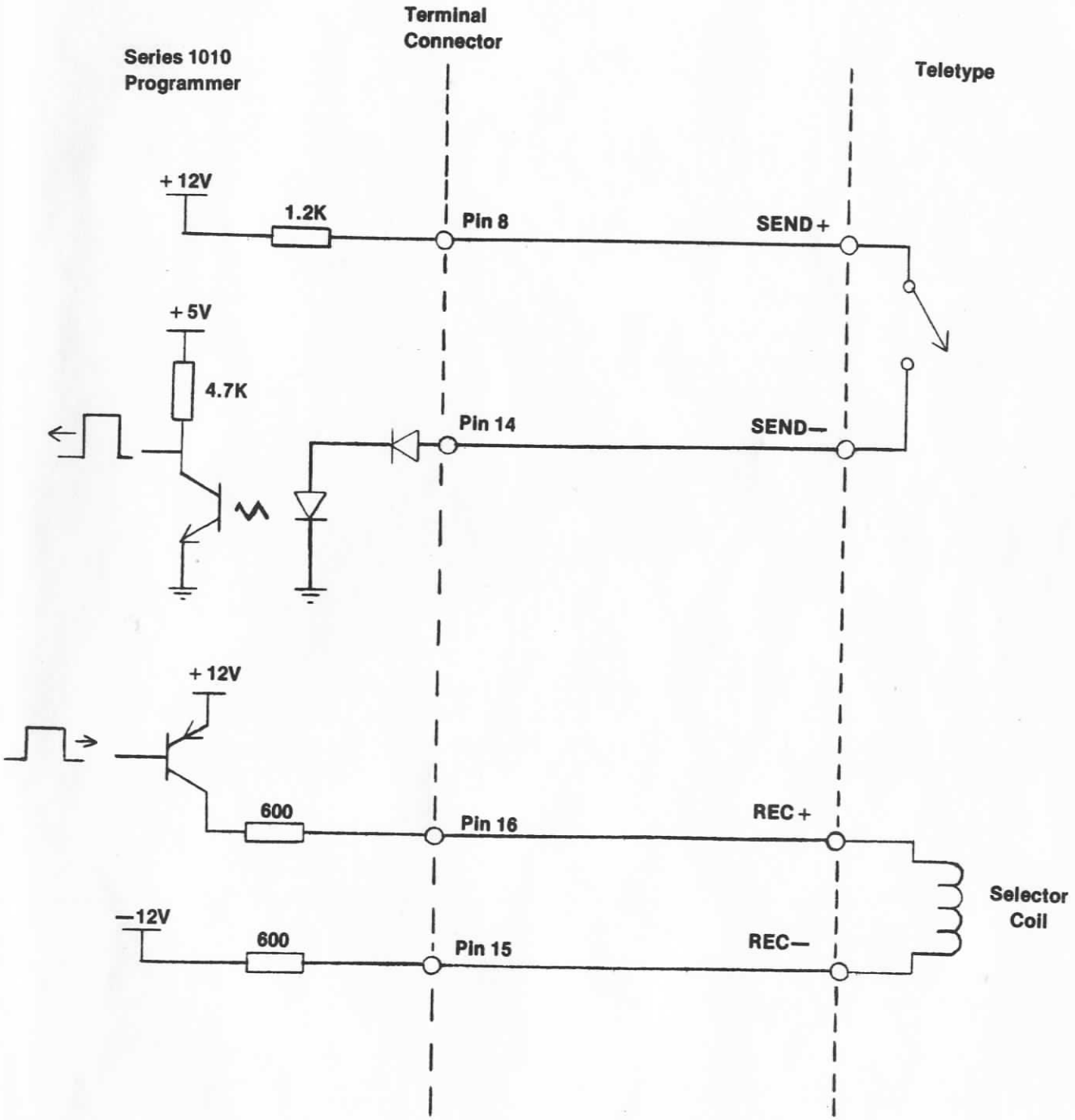


Figure 11. TTY CURRENT LOOP INTERFACE

7.9 UART BOARD SWITCH SETTINGS

Figure 12, below, shows a diagram of the switches on the UART board. There are two switches which the user must set; a six position shorting switch for the baud rate and a seven position dip switch for encoding information. Figure 13 on page 48 shows the pin assignments.

As an example, assume the user wishes to set the IM1010 for 7-bit ASCII, even parity, 2 stop bits, RS-232 and 1200 baud:

Then the 7 position dip switch will have positions 1, 4, 6 in the "ON" position and positions 2, 3, 5, 7, "OFF". The 6 position shorting switch will be set at the third position from the right.

Normally, the RS-232C user should set positions 4 and 6 "ON" (7-bit ASCII and RS-232C). This results in 2 stop bits. Then, depending on parity, set positions 1 and 2.

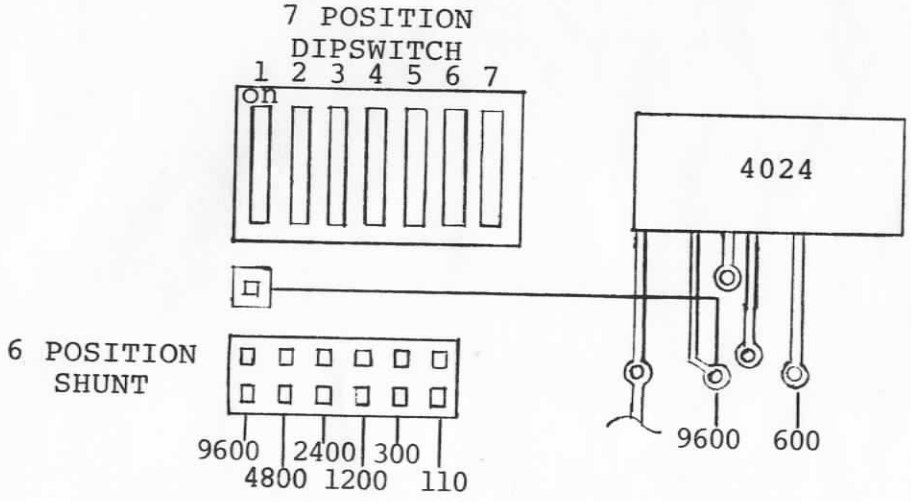


Figure 12

7 POSITION SWITCH

Position	On	Off	Name
1	Parity	No Parity	No Parity
2	1 Stop Bit	2 Stop Bits	Stop Bits
3-See Below			
4-See Below			
5	Odd Parity	Even Parity	Odd/Even Parity
6	RS-232C	20ma Current	RS-232/Current Loop
7-Unused			

Explanation of Positions 3 & 4 -- Number of Serial Data Bits

Data Bits	Position 3	Position 4
5	On	On
6	On	Off
7	Off	On
8	Off	Off

6 Position Shorting Switch

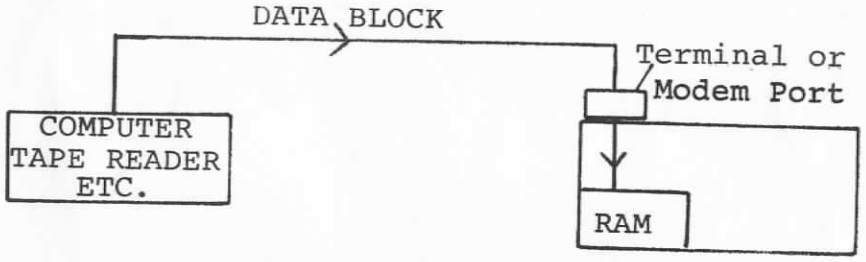
With Position 1 being situated as the right most switch:

Position	Baud Rate
1	110
2	300
3	1200
4	2400
5	4800
6	Jumper -- 600, 9600

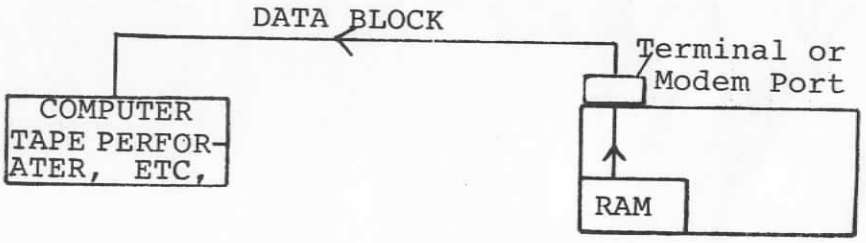
Figure 13. Switch Settings for I/O Control Board

7.10 TRANSFER OF PROM DATA VIA SERIAL INTERFACES:

The IM 1010 incorporates a RAM memory for storage of data to be programmed into the PROM. By using various functions of the IM1010, this RAM may be filled with data via the two serial I/O ports. This is called a Remote Program operation. Also, it is possible to transfer out of the IM1010 RAM a block of data. This is called a Remote List operation. Figure 14, below, illustrates the normal block transfer desired.



Remote Program



Remote List

Figure 14

7.11 There are numerous encoding methods for the transfer of these blocks. They differ in speed, error checking and readability. However, all formats must supply or imply the following information:

1. Start of block
2. Address of data
3. Data
4. End of block

Some encoding methods supply information such as end-of-line terminators and checksum digits. The standard format incorporated into the IM1010 is shown in Figure 15, below.

Other common formats supported are the Intel Intellec Hex, Motorola Micbug, and Tektronix Tek Hex. Consult an IM Price List for an updated listing. If the user has a special format, this can generally be supplied by IM at a low cost. Don't hesitate to ask.

IM STANDARD FORMAT	
Start of text	:
Carriage return (optional) Hexadecimal address	XXX
Space (optional)	
Hexadecimal data	XX
Carriage return	
.	.
.	.
End of text	/

Figure 15

7.12 HOW BLOCK TRANSFERS ARE INITIATED

Although this may be done in a number of ways depending on the particular system, an example will show a typical method. Assume the user has a terminal daisy-chained through the IM1010 to a computer. Also assume he or she wishes to program a PROM with data residing in the computer, which has been formatted in IM's standard format. Figure 16, on page 51, outlines how this may be done.

The user first places the IM1010 in the Remote Mode by keying in "H 01 ENTER" on the IM1010 keyboard. This takes the IM1010 off line. The user then instructs the host computer with the commands to dump the PROM data file to the IM1010. However, previous to the last character sent by the user to the computer to initiate the transfer (usually a carriage return), the user pushes the ENTER key on the programmer. This places the programmer in the REMOTE PROGRAM mode. Upon entry of the CR, the computer dumps the file and it is read into the IM1010 RAM. The user may now program the PROM from the IM1010 keyboard.

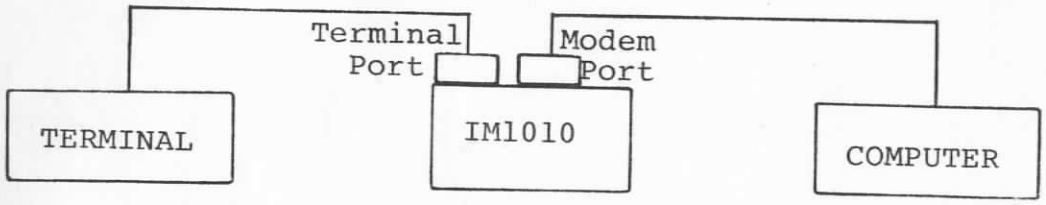


Figure 16

	Terminal	Programmer Input	Comments
1.		H 01 ENTER	IM1010 in REMOTE Mode
2.	Dump File XXX YYY		Set computer up for file dump
3.		ENTER	IM1010 ready to accept file
4.	CR		Initiate dump
5.		(H PROG 01)	File received; programmer back in REMOTE mode

7.13 TRANSMISSION ERROR CHECKING

If a transmission error occurs any time during the transfer of data to the IM1010 programmer, the programmer will display an error code after receiving the end-of-file character. The display will read:

H EX

The hexadecimal number "X" is interpreted as shown in Figure 17, below.

TRANSMISSION ERROR DISPLAY

Let X8X4X2X1 by the four binary bits of X with:

X8 = msb and X1 = isb

A one in any of these bits is interpreted as follows:

Bits	Meaning
X8 = 1	Line Data Error. Not a sufficient number of hexadecimal characters between two carriage returns.
X4 = 1	Overrun Error. Two ASCII characters have been received before one of them could be processed by the programmer.
X2 = 1	Framing Error. Incorrect number of stop bits.
X1 = 1	Parity Error. An ASCII character has been received which does not have correct parity.

The hexadecimal character "X" will be the sum of these individual errors. For example, if the display reads:

H E9

The X8 = 1 and X1 = 1 indicating that both a line data error and a parity error have occurred.

The result of these errors is only significant to the processing of X8 = 1, a line error. If a line of hexadecimal characters is started but not completed, i.e., XX YY CR sequence is received (insufficient address digits) the programmer will ignore this data and wait for a correct transmission of line data or, if appropriate, an end-of-file character.

The user may choose to ignore the error message. However, a prudent course of action would be to investigate why the error occurred and possibly retransmit the file.