Operating Manual

for

System 3000

Universal Programming Station

Headquarters
Western Regional Office
Stag Microsystems, Inc.
1600 Wyatt Drive
Santa Clara, CA95054
USA
Tel: (408) 988-1118

Eastern Regional Office
Stag Microsystems Inc.
3 Northern Boulevard
Amherst, NH 03031
Tel: (603) 673 4380

Address-01
803 1033 Rev 1 APĆ

System 3000 — Contents

Logic Device Programming Sections 9-15

Trademarks

IBM, PC and AT are trademarks of International Business Machines.
PAL is a trademark of Monolithic Memories Inc.
GAL is a trademark of Lattice.
DEC is a trademark of Digital Equipment Corp.
Stag Hex and Master Pack are tradenames of Stag.

1

SECTION 1

stag

Sophisticated systems for the discerning engineer.

1 GENERAL INTRODUCTION

The System 3000 is a Universal programmer designed to program any programmable part in any technology.

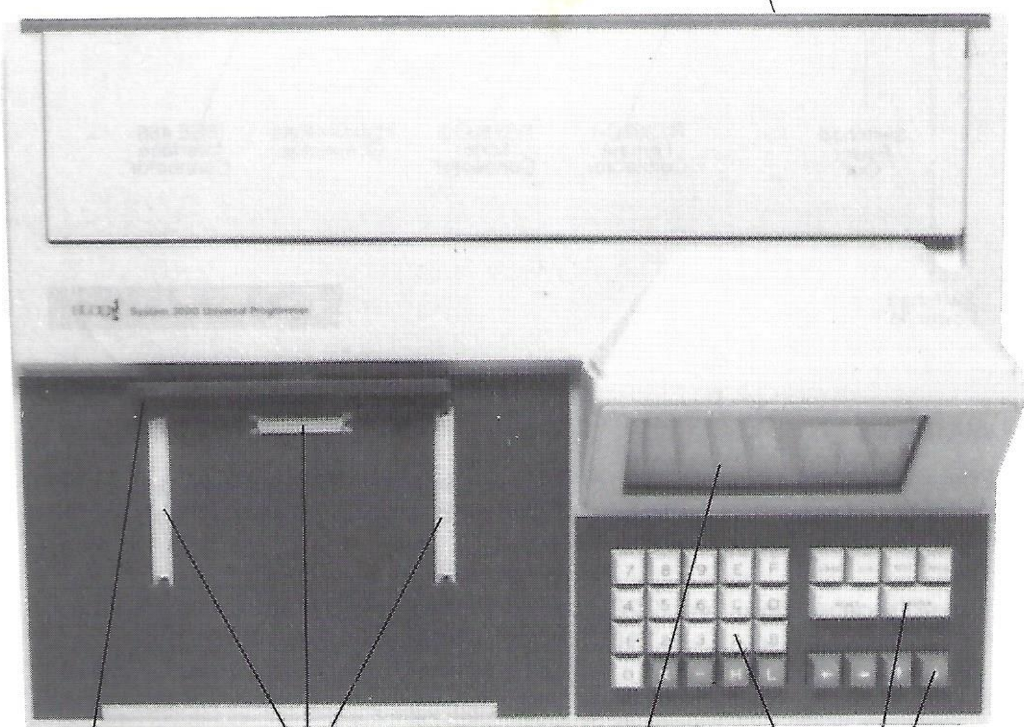
1.1 SYSTEM 3000 MAINFRAME

THE SYSTEM 3000 HAS THE FOLLOWING COMPONENTS:

Z-PACK RACK: THIS HOUSES THE Z-PACKS FOR SPECIAL SOFTWARE ENHANCEMENTS

MODE SWITCH: THREE POSITION KEY SWITCH GIVING, LOCAL MODE, REMOTE MODE AND EDIT LOCK-OUT MODE

TOP FACE OF SYSTEM 3000 (UNDERNEATH THE HINGED FLAP)



THREE SOCKETS TO MATE WITH PROGRAMMING STATION

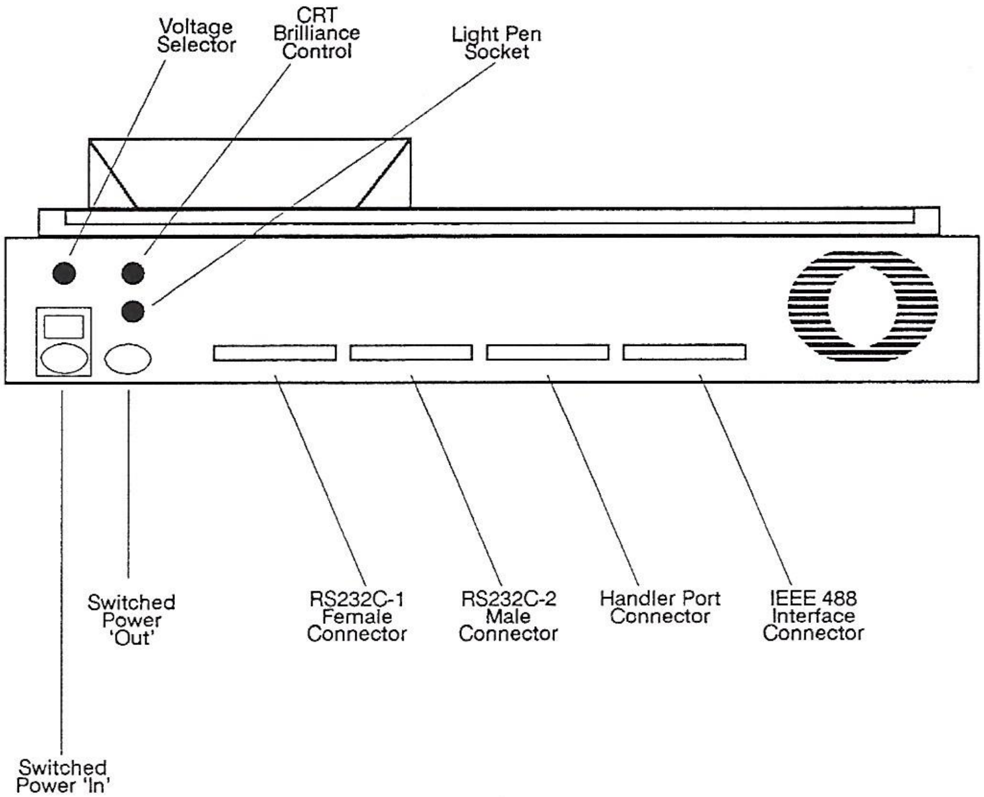
RELEASE LEVER

ALIGNMENT BAR

KEYBOARD: FOR DATA ENTRY AND OPERATING PROGRAMMER FUNCTIONS.

DISPLAY: 80 CHARACTERS x 18 LINES GREEN PHOSPHOR C.R.T.

System 3000 Rear Panel

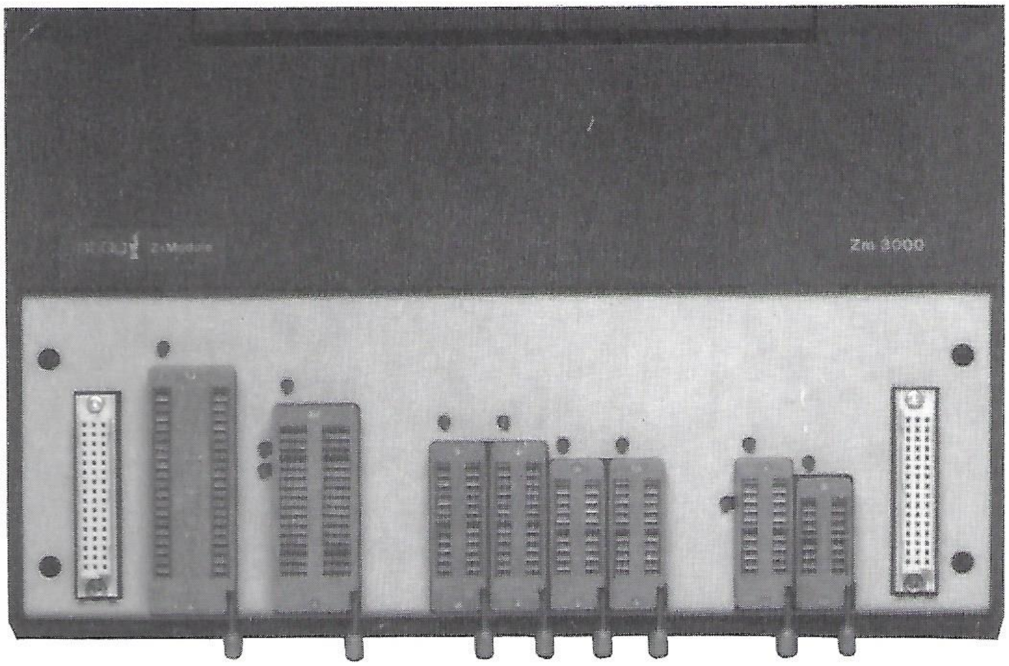


WARNING

Ensure that the voltage selector on the rear panel of the System 3000 is indicating the correct voltage. Failure to do this may result in irreparable damage to the System 3000.

The voltage settings are: 220V -10%/+20% (nominal)
110V -10%/+20% (nominal)

1.1.1 Zm3000 Universal Programming Station

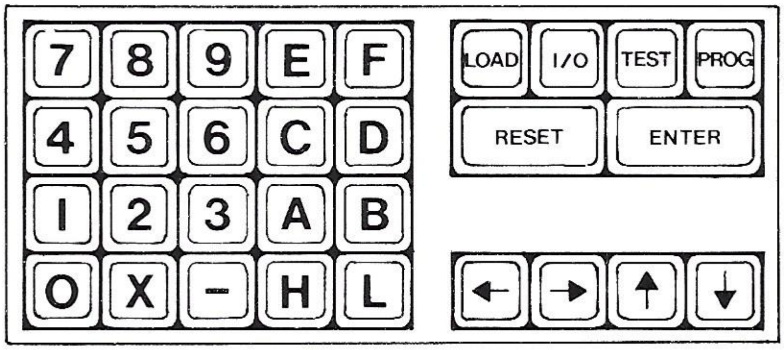


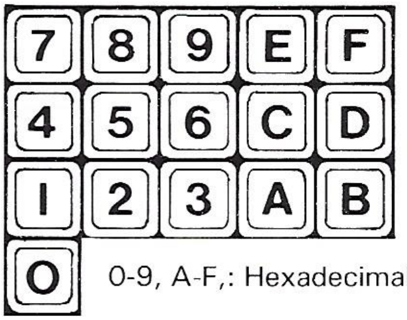
The Zm3000 Universal Programming Station is equipped with 8 ZIF sockets which enable it to program the vast majority of programmable devices in every technology. An optional SMD Chip Station is available to allow programming of PLCC and LCC equivalents of DIP devices supported by the Zm3000. The SMD Chip Station plugs into the DIN connectors at either end of the Zm3000.

A unique feature of the Zm3000 is the 'Master Pack' IC card which stores the controlling software including device libraries and programming algorithms. Software upgrades are now performed easily and quickly by just removing the existing 'Master Pack' and inserting a new one. This can be done by the user, thus removing the need to return units to the supplier for an upgrade.

A further benefit of the 'Master Pack' is the greater speed it provides. It has an access time which is far superior to that of a floppy disk and therefore reduces the time required to carry out any function.

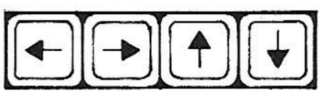
1.2 **KEYBOARD—for data entry and operating programmer functions**



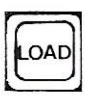


0-9, A-F,: Hexadecimal Keyboard.

X—HL: Entry keys for PAL* and IFL*



Four cursor control keys

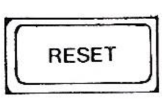


LOAD key—to load the programmer RAM

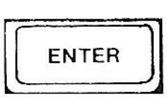
I/O Key—to execute input or output via the I/O ports.

TEST Key—to initiate a test function (empty, verify etc.)

PROGRAM Key—to initiate a sequence of tests and the program routine.



RESET Key—to return the programmer to its initial state.



ENTER Key—to enter a selected parameter or to execute a pre-selected function.

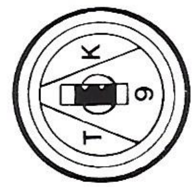
1.3 SELECTING THE OPERATING MODE

Selection of the operating mode is made by use of the three position key-switch located underneath the hinged flap. By inserting the supplied key one of these modes can be chosen.

LOCAL MODE (arrow vertical)

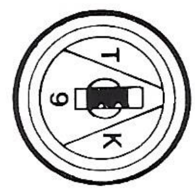
In LOCAL mode all functions of the System 3000 are controlled from its own keyboard. All the functions operate interactively with the CRT and Input/Output transmissions take place between the System 3000 and the peripheral equipment under 'Local Control'.

REMOTE MODE (arrow pointing to the left)



In REMOTE the System 3000 operates under remote control from a computer or a terminal. The keyboard of the System 3000 is inoperative at this time and the CRT will only display information as requested under remote command.

EDIT LOCK-OUT MODE (arrow pointing to the right)



The Edit Lock-out mode ensures all editing key functions are inoperative but that all other functions operate normally under LOCAL MODE conditions.

In this mode data corruption or changes to the RAM data cannot take place and security of information is ensured.

Interchange between LOCAL and REMOTE MODE can be accomplished by turning the key-switch to the required direction and pressing RESET.

Interchange between LOCAL and EDIT LOCK-OUT MODE can be made by simply turning the key to the required direction. (Should an edit key function be requested under Edit Lock-Out Mode the CRT will display: EDIT LOCKED OUT)

1.4 **Initial Setting-up Procedure and Self-Test**

Before attempting to power-up your System 3000 ensure that it is set to the correct operating voltage in line with your power supply. The voltage setting is printed on the rear panel. Use a suitable flat bladed instrument to adjust the setting if necessary.

1. Plug the supplied power-cord into the rear panel socket.

2. Connect power-cord to power supply.

3. Power-up the System 3000 using the ON/OFF switch on the rear panel.

4. Wait a few seconds for the CRT to warm up and then adjust the picture brightness using the brilliance control on the rear panel.

After "POWER-UP" and without the programming station inserted, the display will read:

```
— POWER UP ———————————————— MAINFRAME Rev X —

                 POWER DOWN
       INSERT SYSTEM MODULE or PACK
                and POWER UP
```

The CRT has a display protection mechanism to save the screen from burning. Therefore, the display will shut-off automatically if the screen is not in use for a period in excess of 4 minutes. (To restore the picture, press any of the cursor keys.)

Setting-up Procedure

1. Power down the System 3000 with the on/off switch on the rear panel.
2. Plug in the Zm3000 making sure the sockets on the programming station and the mainframe are properly aligned. Use the alignment bar on the front of the mainframe for lateral positioning. Make sure the programming station is as far forwards as possible and resting against the alignment bar before pressing down to engage the connectors. The lever to the rear is only used to release the programming station and will automatically find its rest position when the programming station is properly engaged.
3. Check that the 3-position key switch under the hinged flap is indicating the required mode. (For example: Local Mode—Arrow Vertical).
4. Power-up using the on/off switch on the rear panel. On initial power-up, the display will show the 'Set Family' menu. PROMs, PLDs or Micros can be selected by pressing the appropriate key.

```
— Set Family ———————————————————————— Zm3000 Rev X —

   Family Type:

      1 . . . Prom/Eprom

      2 . . . Logic Device

      3 . . . Microcomputer


                                    Select Family:
```

On any subsequent power-up, the display will show the Main System Menu relating to the last used device type.

Self Test

On Power-up, Self Test routines are carried out within the module. If no faults are found, the programmer will initialize in the normal manner. In the event of a fault being located the keyboard will become inoperative and an error message will be displayed.

Should this situation arise it will be necessary to arrange with your supplier to have the fault(s) rectified.

1.5 **Horn Control**

To determine whether the Zm3000 should operate with the warning horn activated or silent:

Press KEY A followed by KEY 4, to display:

— SET SYSTEM ——————————————— Zm3000 Rev X —

4. Miscellaneous

Horn Control: Off On

Remote I/O: RS232C 1 RS232C 2 IEEE

Pass Through: Off On

RESET . . . System Menu
ENTER . . . Set System

The arrow keys can be used to move the cursor to select 'off' or 'on'.

Press ENTER

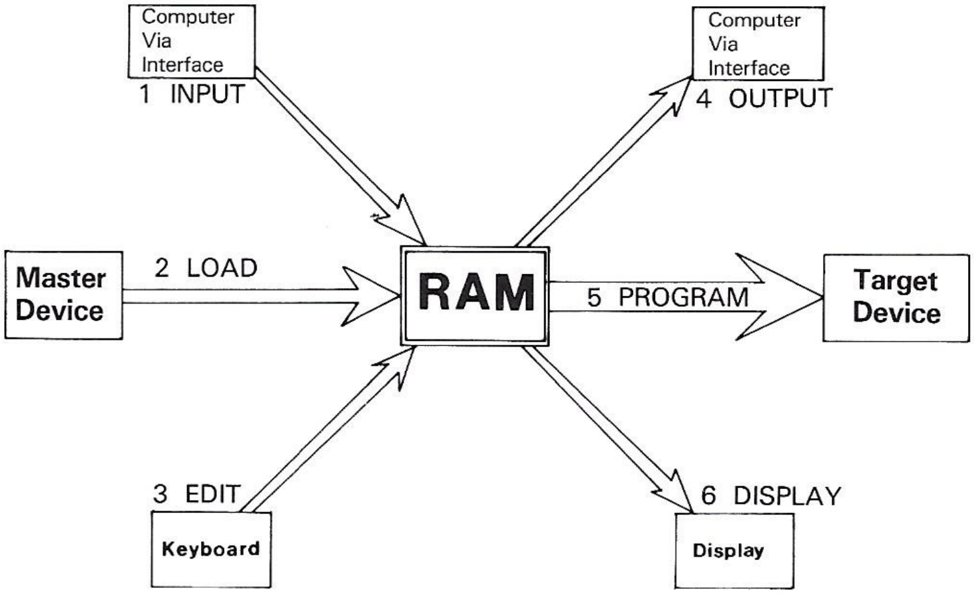
to confirm selection

The Set System Menu is re-displayed

Press RESET

to return to the System Menu.

1.6 RAM OPERATING STRUCTURE



There are six RAM data transfer functions on the System 3000.

Data can be loaded into the RAM from three different sources, these are:

1. **INPUT:** From a computer, terminal or microprocessor development system through the IEEE-488 interface port or either of the RS232C interface ports.

2. **LOAD:** From a master device or devices, inserted into one of the ZIF sockets.

3. **EDIT:** Controlled from the keyboard. Functions include Enter/Modify Data, Nibble Swap, Block Move, Complement, ANDing, ORing and filling the RAM with Os, Fs, Arbitrarily and the Unprogrammed State

   Data can be transfered from RAM in three ways:

4. **OUTPUT:** Through the IEEE-488 interface port or either of the RS232C interface ports to a computer, terminal or microprocessor development system.

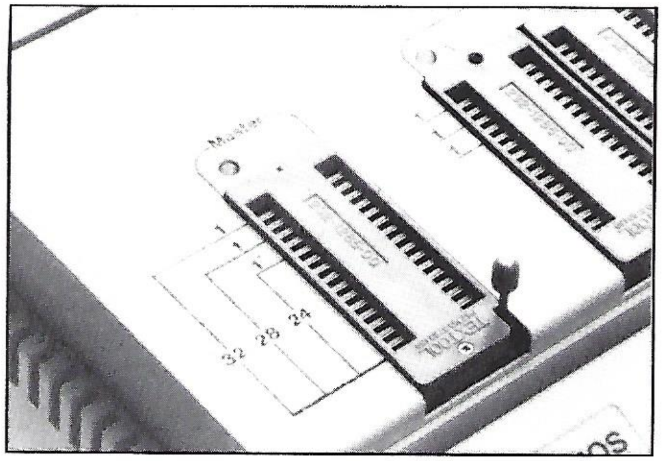
5. **PROGRAM:** The title function whereby data is finally copied from the RAM to a device in a ZIF socket.

6. **DISPLAY:** Visual confirmation of operations undertaken.

1.7 **Correct Operation of ZIF Sockets.**

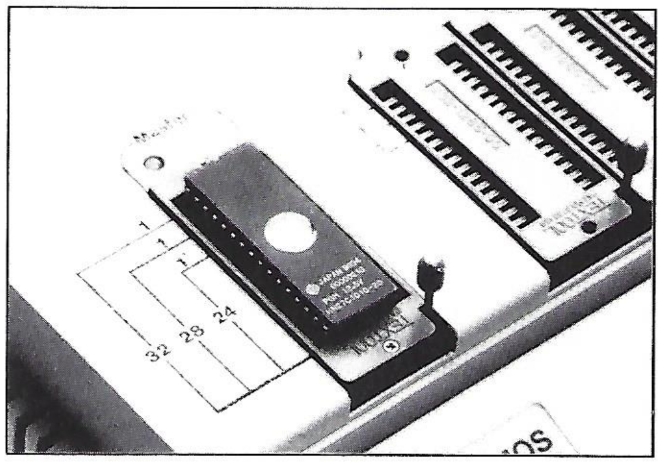
1. Empty ZIF socket.

   Lever in open (up) position.



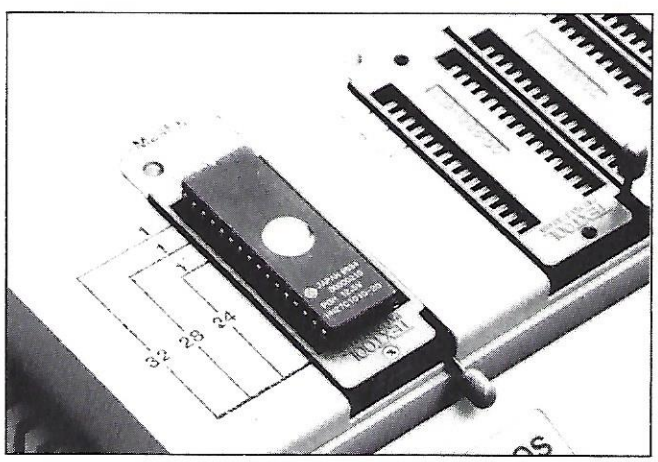
2. Device inserted.

   Socket open.



3. Device inserted.

   Lever in closed (down) position.



1.8 **Fitting 'Master Pack' Software Upgrades**

1. Power-down System 3000

2. Remove Zm3000 programming station using release lever.

3. Invert Zm3000. Hold in left hand with ZIF sockets on the right hand side. The 'Master Pack can be seen through the window in the Zm3000 base plate.

4. Place fingers of right hand under Zm3000 (finger ends resting on ZIF sockets). Place thumb of right hand on 'Master Pack'.

5. Exert slight downwards and rightwards pressure with thumb. The 'Master Pack' will disengage from its connectors and slide out through the slot in the casing.

6. Insert new 'Master Pack' into Zm3000*. Some resistance will be felt as the connectors engage. Push home firmly.

   ***Important Notice:** Ensure correct orientation of 'Master Pack'. The Stag label should be visible through window in base plate of Zm3000 and an arrow on the label denotes direction of insertion.

Memory Device Programming
Sections 2-8

for the following devices:

PROM; EPROM; EEPROM and Micros.

Logic Device Programming
Sections 9-15

for the following devices:

PLD; EPLD and EEPLD.

Common to Both
Sections 16-17

Memory Device Programming
Sections 2-8

2

SECTION 2

stag

Sophisticated systems for the discerning engineer.

2 Device Selection (Memory Devices Only)

2.1 Selecting the Device to be Programmed

Device selection on the System 3000 is very straightforward.

Step 1—Set Family

Firstly, the correct 'family' of device has to be chosen, i.e.: Prom/Eprom
Logic
Microcomputer.

If the 'family' is already correct, proceed to Step 2—Set Device.

To enter the Set Family option:

Press Key 0.

This will display the Set Family menu. Press Key 1 or 3 to select either of the memory device categories. The relevant main system menu will be displayed as well as the last chosen device in that category.

Step 2—Set Device

Press Key 1

This will display the Select Manufacturer screen. The information is displayed on several pages and is grouped according to general device type i.e. PROM, EPROM, EEPROM or Microcomputer and sub-divided within these categories alphabetically i.e. EPROM A-M, EPROM N-Z, etc.

The available categories are displayed in a column on the right hand side of the screen and the chosen category is highlighted with a static cursor. The list of manufacturers relating to the chosen device category is displayed in a column on the left hand side of the screen. The chosen manufacturer is highlighted by a flashing cursor.

The list of categories can be scrolled through until the correct selection is made by using either of the horizontal cursor keys ← and →. After selecting the required category, scroll through the list of available manufacturers using the vertical cursor keys ↑ and ↓ or enter the two digit manufacturer code from the keyboard. Press Enter.

This will display the list of devices available for this manufacturer. If more than one page of devices is available, it can be selected by using either of the horizontal cursor keys ← and →. The device can be selected by scrolling through the list using the vertical cursor keys ↑ and ↓ or entering the two digit pinout code from the keyboard.

Press Enter to confirm the selection.

Pressing Reset at any time during the selection process will cause the previously selected device to be re-instated and the main system menu will be returned.

2.2 **Device Description**

By pressing Key 2 as prompted by the System Menu a description of the selected device can be displayed.

```
— DEVICE DESCRIPTION ———————————— Zm3000 Rev X ——

   AMD Interactive

   EPROM                          64K x 8 Bits

   27512                          Tristate

   Address Limit . . . OOFFFF     Package:
   Empty State . . .       FF     28 Pins x 0.6 in DIL.

   Manufacturer Code  9F
   Pinout Code . . .    4D        RESET . . . System Menu
```

By pressing RESET the System 3000 will return to the System Menu.

2.3 **Electronic Identifier (for EPROMs only)**

Electronic Identifier is a term used to describe a code mask or signature programmed into a PROM which identifies the device type and manufacturer. The code is stored outside the normal memory array and is accessed by applying 12 volts to address line A9. This allows direct identification of any device containing an Electronic Identifier and thus eliminates the need for the user to select the device type.

Important Note: Devices which do not contain an Electronic Identifier may be irreparably damaged if the Electronic Identifier function is used.

The Electronic Identifier can be selected from the EPROM A-M category. The selection process is the same as for any device manufacturer as detailed in Section 2.1, Step 2.

SECTION 3

stag

Sophisticated systems for the discerning engineer.

3 Address Increment

3.1 Address Increment of 8-bit devices

This is a feature related to the device functions of:

PROGRAM
LOAD
TEST (VERIFY/EMPTY)

and the RAM functions of:

FILL RAM
LOGICAL 'OR' DATA
LOGICAL 'AND' DATA
NIBBLE SWAP
COMPLEMENT
CYCLIC REDUNDANCY CHECK

Increment is an increase in the amount of data bytes that constitute the size of a word within the RAM (1 byte = 8 bits).

The 8-Bit Mode corresponds to the default state of the increment as displayed on the CRT.

i.e.: Address Increment . . . 1

8, 16 and 32-Bit Modes are the most widely used configurations, however the Zm3000 has an inbuilt flexibility whereby other bit-modes can be applied if required by the user.

8 Address Increments are available for selection via the keyboard

ADDRESS INCREMENT	BIT MODE	BYTES PER WORD	No. of Devices required for Load & Program
1	**8-BIT MODE**	**1**	**1**
2	**16-BIT MODE**	**2**	**2**
3	24-BIT MODE	3	3
4	**32-BIT MODE**	**4**	**4**
5	40-BIT MODE	5	5
6	48-BIT MODE	6	6
7	56-BIT MODE	7	7
8	64-BIT MODE	8	8

3.2 **8-Bit Mode (Address Increment . . . 1)**

PROGRAM

Data at every location within limits that automatically default to the device size, will be programmed from the RAM to the device.

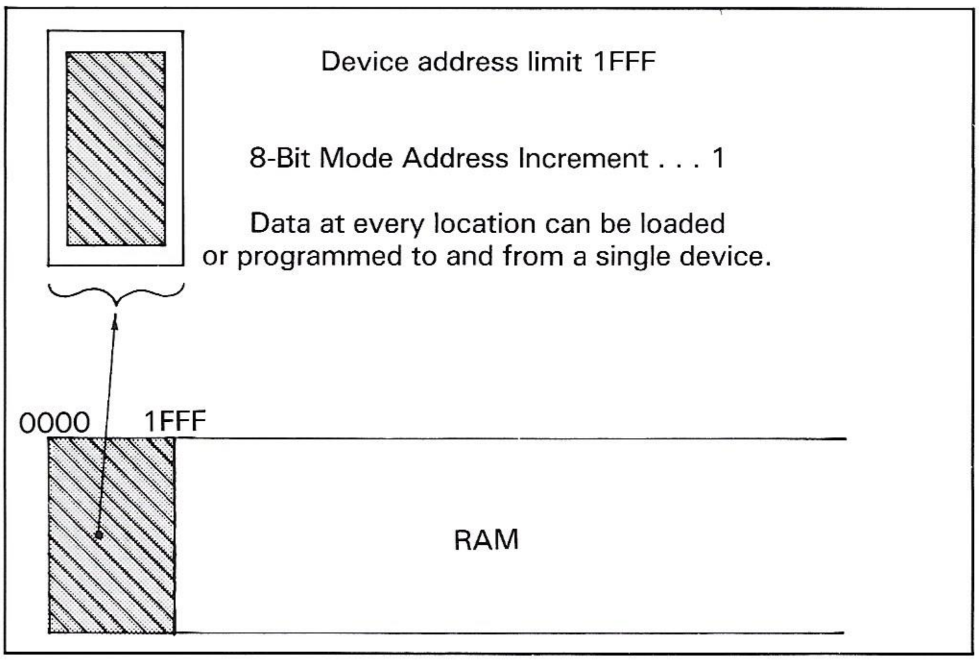
After program is complete the device will contain identical information to that of the RAM.

LOAD

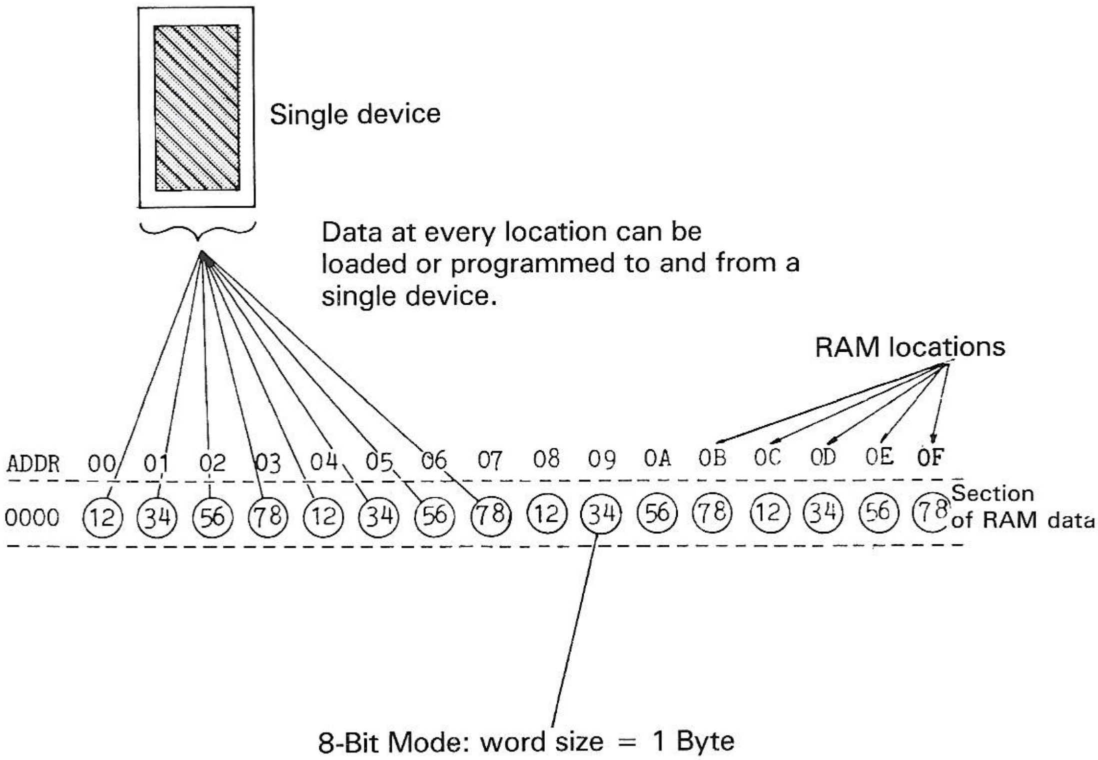
Data at every device location will be loaded into the RAM.

After load is complete the RAM will have the same address limits and will contain identical information to that of the device.

For clarity the following example will show a fixed device size:
1FFF (2764 EPROM)



Individual bytes of data make up one word in the 8-Bit Mode:



Verify (Test Mode)

Data transmitted can be verified with its source after either load or program is complete, providing stop and start addresses and an address increment remain consistent.

3.3 16-Bit Mode (Address Increment . . . 2)

PROGRAM

Two 8-Bit devices can be programmed in two operations from a common address range within the RAM. The address data will be 'split' during this programming sequence into two separate bytes of data: Byte 1 and 2.

FIRST DEVICE
Address Increment . . . 2
Byte 1
RAM Start address 0000

Byte 1 will be programmed into the first device from the RAM. It represents the (MSB) ''most significant byte'' of a 16-Bit parallel word.

In this way the address increment . . . 2, with a start address of 0000 selects alternate bytes of data at even numbered locations within the RAM's address range and programs them into the device.

SECOND DEVICE
Address Increment . . . 2
Byte 2
RAM Start address 0001

Byte 2 will be programmed into the second device from the RAM. It represents the (LSB) ''least significant byte'' of a 16-Bit parallel word.

In this way the address increment . . . 2 with a RAM start address of 0001 selects alternate bytes of data at odd numbered locations within the RAM's address range and programs them into the device.

LOAD

Two 8-Bit devices can load in two operations into a common address range within the RAM.

The address data to be loaded is split between the devices into two separate bytes of data: Byte 1 and Byte 2.

FIRST DEVICE
Address Increment . . . 2
Byte 1
RAM Start address 0000

Byte 1 will be loaded from the first device, into the RAM. It represents the (MSB) ''most significant byte'' of a 16-Bit parallel word.

In this way the address increment . . . 2, with a RAM start address of 0000, selects data bytes from every 'first device' location and loads them alternately into even numbered RAM locations.

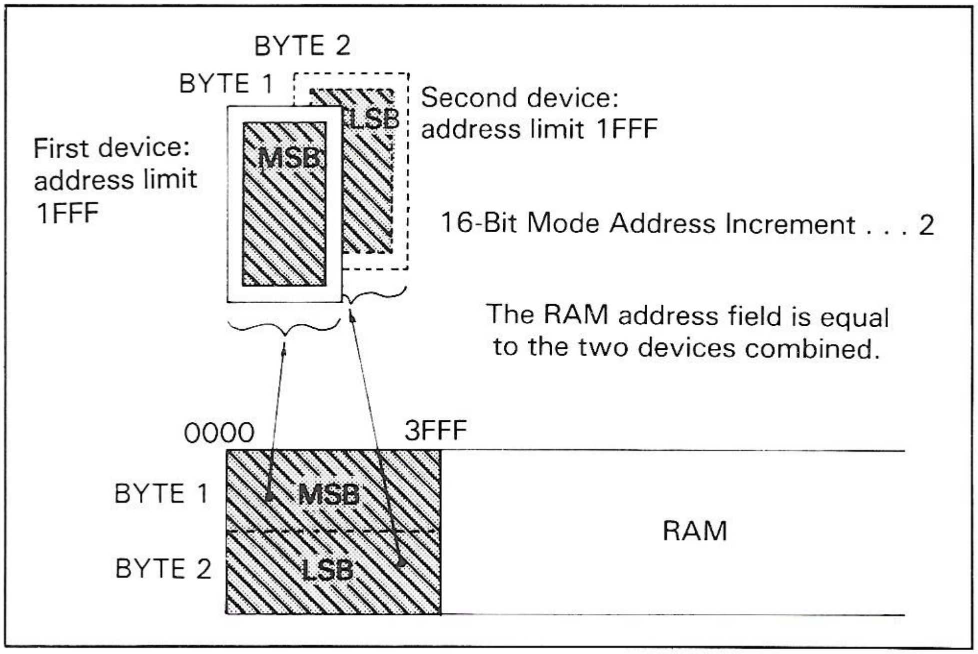
SECOND DEVICE
Address Increment . . . 2
Byte 2
RAM start address 0001

Byte 2 will be loaded from the second device, into the RAM. It represents the (LSB) ''least significant byte'' of a 16-Bit parallel word.

In this way the address increment . . . 2, with a RAM start address of 0001, selects data bytes from every 'second device' location and loads them alternately into odd numbered RAM locations.

A graphic example of how the 16-Bit Mode (Increment . . . 2) works on both LOAD and PROGRAM is shown below:

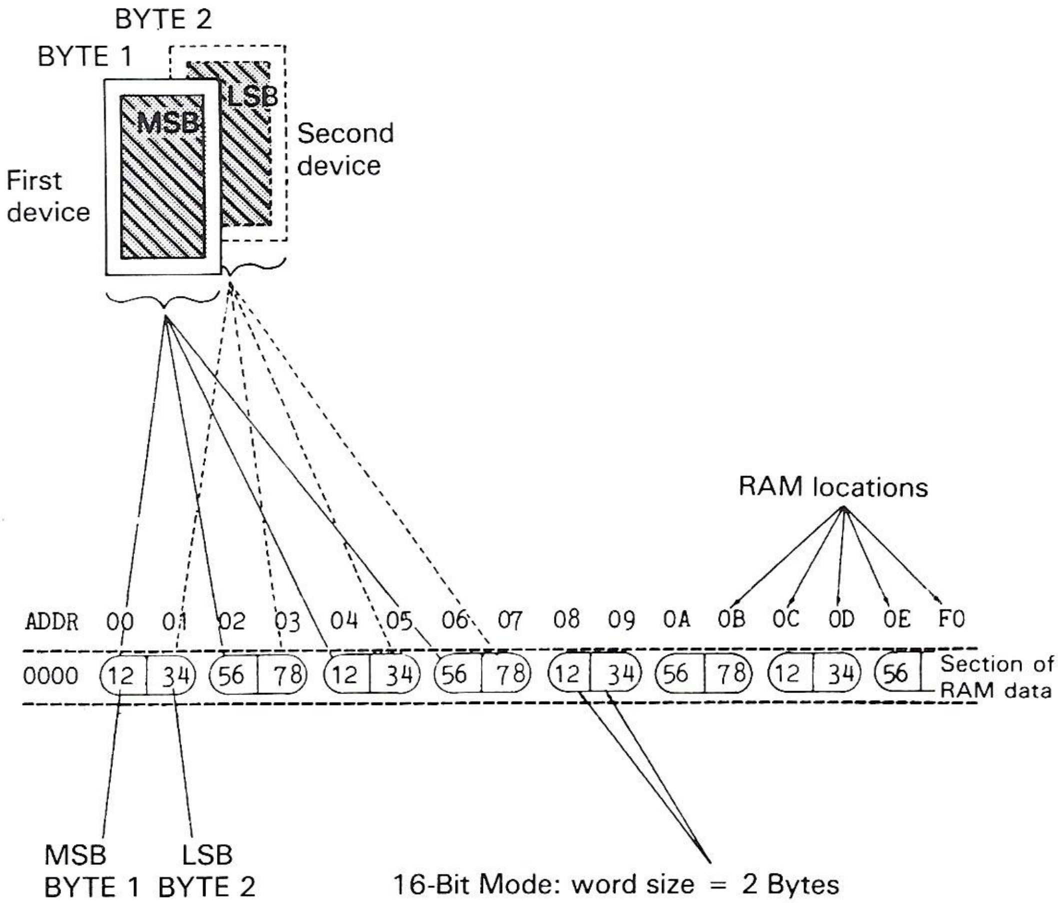
For clarity the following example will show a fixed device size: 1FFF (2764 EPROM)



In the first operation: PROGRAM or LOAD is made to and from a single 8-Bit device with only the most significant byte ''MSB'' of a 16-Bit Word undergoing data transfer.

In the second operation: PROGRAM or LOAD is made to and from a single 8-Bit device with only the least significant byte ''LSB'' of a 16-Bit word undergoing data transfer.

Two bytes of data make up one word in the 16-Bit Mode:



MSB—The "most significant BYTE" in binary code
LSB—The "least significant BYTE" in binary code

Verify (Test Mode)

Data transmitted can be verified with its source after both load or program are complete, providing that stop and start addresses and an address increment remain consistent.

3.4 **32-Bit Mode (Address Increment . . . 4)**

PROGRAM

Four 8-Bit devices can be programmed in four operations from a common address range within the RAM. The address data will be divided during this programming sequence into four separate bytes of data: Byte 1, Byte 2, Byte 3 and Byte 4.

FIRST DEVICE
Address Increment . . . 4
Byte 1
RAM Start Address 0000

Byte 1 will be programmed into the first device from the RAM. It represents the (MSB) ''most significant byte'' of a 32-Bit parallel word.

In this way the address increment . . . 4, with a RAM start address of 0000 selects every fourth byte of data within the RAM's address range (0000, 0004, 0008, 000C etc) and programs them into the device.

SECOND DEVICE
Address Increment . . . 4
Byte 2
RAM Start Address 0001

Byte 2 will be programmed into the second device from the RAM. It represents the 'second most significant byte' of a 32-Bit parallel word.

In this way the address increment . . . 4, with a pre-selected RAM start address of 0001 selects every fourth byte of data within the RAM's address range (0001, 0005, 0009, 000D etc) and programs them into the device.

THIRD DEVICE
Address Increment . . . 4
Byte 3
RAM Start Address 0002

Byte 3 will be programmed into the third device from the RAM. It represents the ''second least significant byte'' of a 32-Bit parallel word.

In this way the address increment . . . 4, with a pre-selected RAM start address of 0002 selects every fourth byte of data within the RAM's address range (0002, 0006, 000A, 000E etc) and programs them into the device.

FOURTH DEVICE
Address Increment . . . 4
Byte 4
RAM Start Address 0003

Byte 4 will be programmed into the fourth device from the RAM. It represents the (LSB) ''least significant byte'' of a 32-Bit parallel word.

In this way the address increment . . . 4, with a pre-selected RAM start address of 0003 selects every fourth byte of data within the RAM's address range (0003, 0007, 000B, 000F etc) and programs them into the device.

LOAD

Four 8-Bit devices can load in four operations from a common address range within the RAM.

The address data to be loaded is divided between the devices into four separate bytes of data: Byte 1, Byte 2, Byte 3 and Byte 4.

FIRST DEVICE
Address Increment . . . 4
Byte 1
RAM Start Address 0000

Byte 1 will be loaded from the first device, into the RAM. It represents the (MSB) ''most significant byte'' of a 32-Bit parallel word.

In this way the address increment . . . 4, with a RAM start address of 0000 selects data bytes from every 'first device' location and loads them into the RAM at every fourth location: 0000, 0004, 0008, 000C etc.

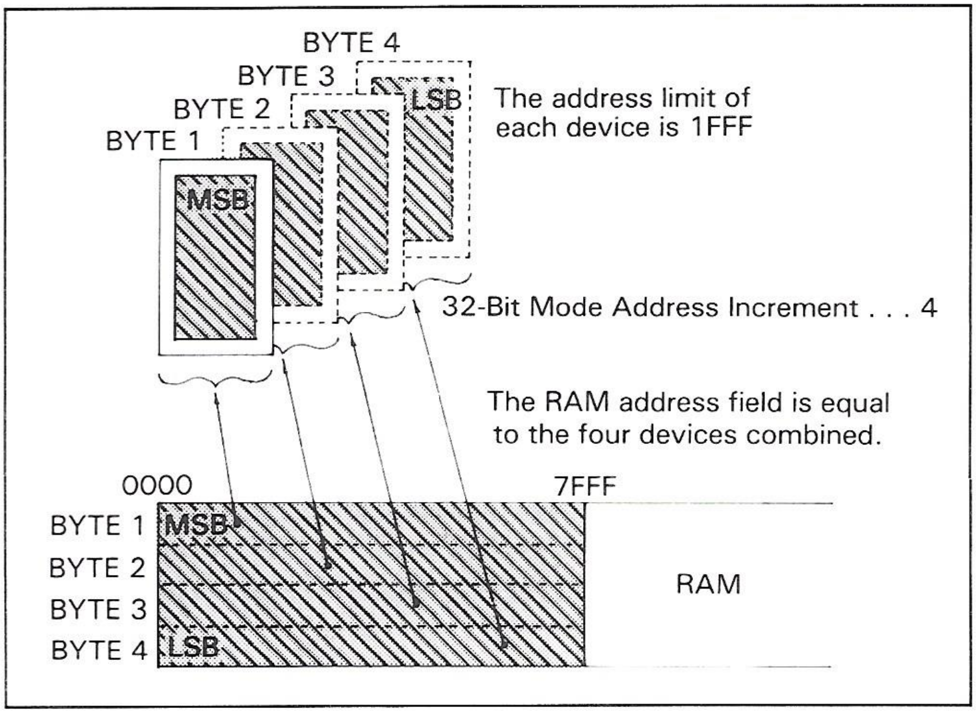
SECOND DEVICE
Address Increment . . . 4
Byte 2
RAM Start Address 0001

Byte 2 will be loaded from the second device, into the RAM. It represents the 'second most significant byte' of a 32-Bit parallel word.

In this way the address increment . . . 4, with a pre-selected RAM start address of 0001 selects data bytes from every 'second device' location and loads them into the RAM at every fourth location: 0001, 0005, 0009, 000D, etc.

A graphic example of how the 32-Bit Mode (increment . . . 4) works on both LOAD and PROGRAM is shown below:

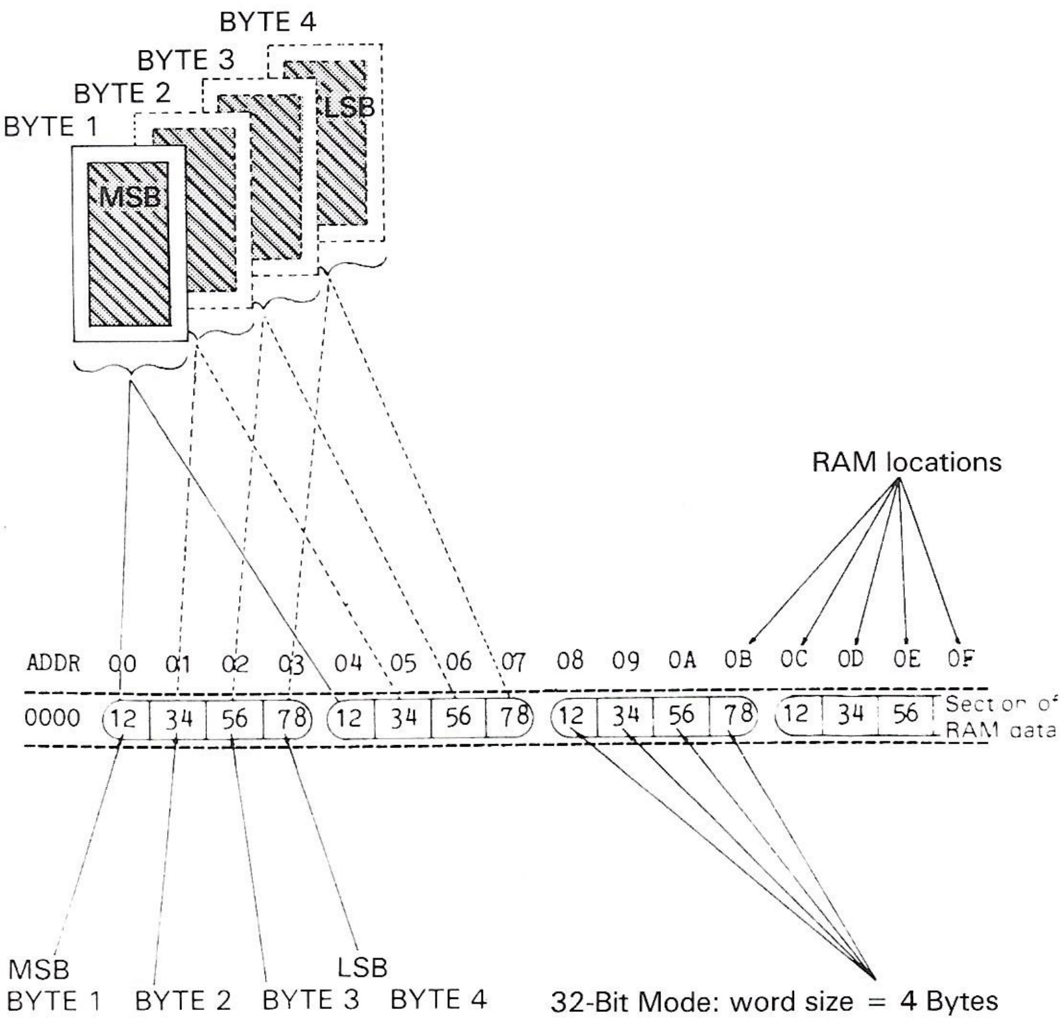
For clarity the following example will show a fixed device size: 1FFF (2764 EPROM)



In the first operation: PROGRAM or LOAD is made to and from a single 8-Bit device with only BYTE 1 the "most significant byte" of a 32-Bit Word undergoing data transfer.

This sequence is repeated for the next three devices with bytes 2, 3 and 4 respectively undergoing data transfer.

Four bytes of data make up one word in the 32-Bit Mode:



MSB—The "most significant BYTE" in binary code
LSB—The "least significant BYTE" in binary code

Verify (Test Mode)

Data transmitted can be verified with its source after both LOAD or PROGRAM are complete, providing stop and start addresses and an address increment of 4 remain consistent with those specified in either operation.

Empty Test

The "address increment" feature, is included in the EMPTY TEST Mode. However, as an EMPTY TEST is applied to the device alone, the "increment function" plus "RAM start address", become superfluous features.

3.5 Nibble Swap (Increment on 4-Bit devices)

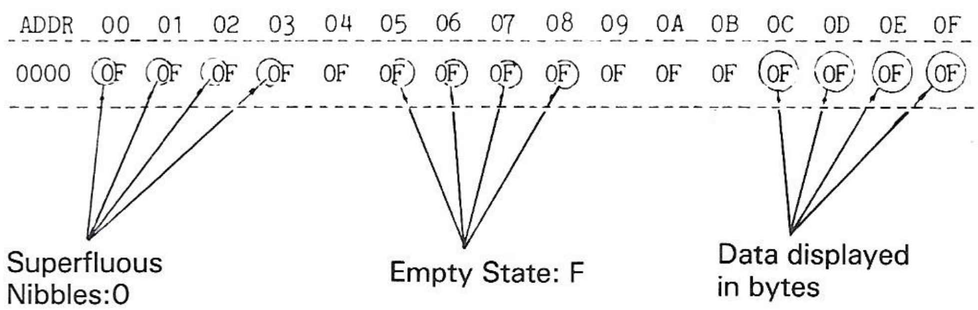
Two 4-Bit devices can be programmed in two operations from a common address range within the RAM, in such a way as to contain 8 Bit Data when combined.

The address data will be 'split' during this programming sequence into two separate nibbles of data.

4-Bit Device. Initial Display

The 4-Bit data within a particular device can be displayed (with the device inserted into its appropriate ZIF socket) by pressing Key 6.

This example shows a section of the display from locations 0000 to 000F.



As can be seen from the above section, the data displayed in each individual location is in the form of bytes. However, the left nibbles containing 'O's' are superfluous throughout the entire device address range. Only the right nibble reflects the condition of the 4-Bit device. In this case the empty state—'F'.

N.B.—In other examples the empty state may be '0' in which case both nibbles would look alike however, the above conditions would still apply.

Programming of 8-Bit Data into two 4-Bit Devices

8-Bit data can be loaded into the RAM in one of three ways:

i From peripheral equipment via the interface ports.
ii From a master 8-Bit Device.
iii By use of the Edit functions.

By pressing key 5 the 8-Bit RAM data can be displayed.

This example shows a section of RAM, from locations 0000 to 000F.

ADDR	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21

1 BYTE = 8 BITS   NIBBLE 2   NIBBLE 1

Two 4-Bit devices can be programmed in two operations from a common address range within the RAM. The address data will be 'split' during this programming sequence into two separate nibbles of data: Nibble 1 and Nibble 2.

For clarity the following examples will show a fixed 4-Bit device size: 00FF (Bipolar PROM) and will be without preselected program offsets.

First 4-Bit Device
Nibble 1

Only the right hand nibble of each data byte will be selected. The RAM will automatically program these nibbles into every location of the device.

When the program is complete the contents of the first device can be displayed by pressing RESET followed by KEY . . . 6

Section of display 0000 to 000F (first device)

ADDR	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01

Superfluous
Nibbles:0

NIBBLE 1

Before the second device can be programmed, the RAM data within the specified limits of 0000 to 00FF must first undergo NIBBLE SWAP:

To enter NIBBLE SWAP:

PRESS KEY 3

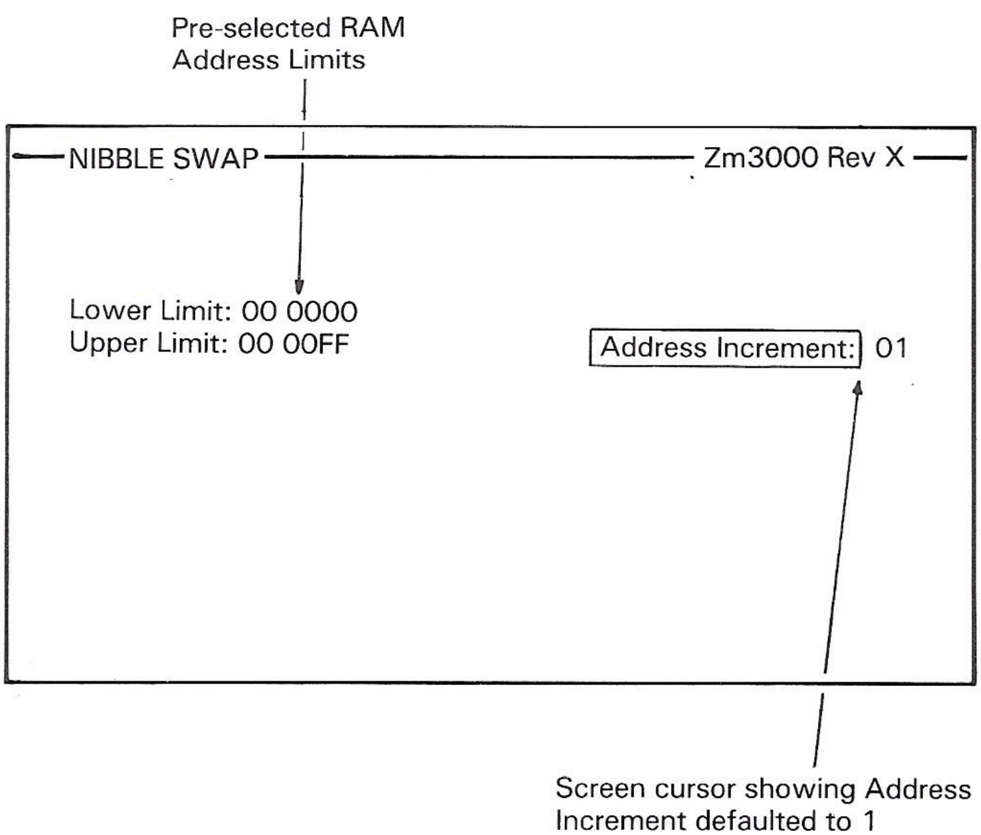
This will display the EDIT RAM DATA sub-menu, then

PRESS KEY 6

The screen will now display the default states of the RAM address limits to undergo the nibble swap. The screen cursor will show the lower RAM address limit defaulted to zero, and the upper limit will be defaulted to the size of the pre-selected device. Interchange between the two limits is made by use of the vertical cursor keys (↑ and ↓), and the new selection can be entered from the hex-keyboard. To confirm these selections:

PRESS ENTER

This will display the following:



If required by the user, the nibble swap function on 4-bit devices can be integrated with Address Increment to handle data in the 16-bit mode and beyond.

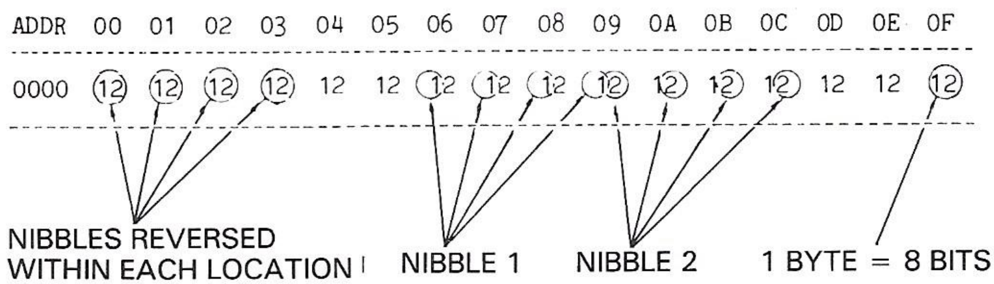
The Address Increment can take values from 1 to 8 which can be entered from the keyboard. To confirm the Address Increment value and implement the nibble swap:

PRESS ENTER

To display rearranged RAM data:

PRESS RESET followed by KEY 5

Section of RAM data 0000 to 000F



To return to the SYSTEM MENU:

PRESS RESET

The second 4-bit device can now be programmed.

Second 4-Bit Device

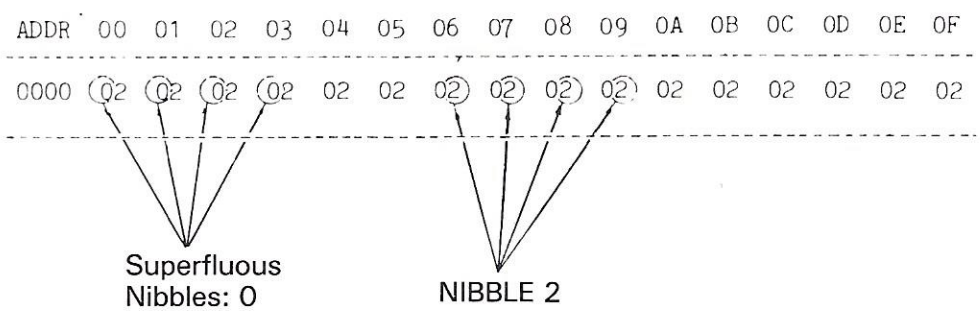
Nibble 2

Replace the first device in the ZIF socket with the second 4-bit device to be programmed.

Once again only the right hand nibble of each data byte will be selected. The RAM will automatically program these nibbles into every location within the device.

When the program is complete the contents of the second device can be displayed by pressing RESET followed by KEY . . . 6.

Section of display 0000 to 000F second device



To return to the SYSTEM MENU:

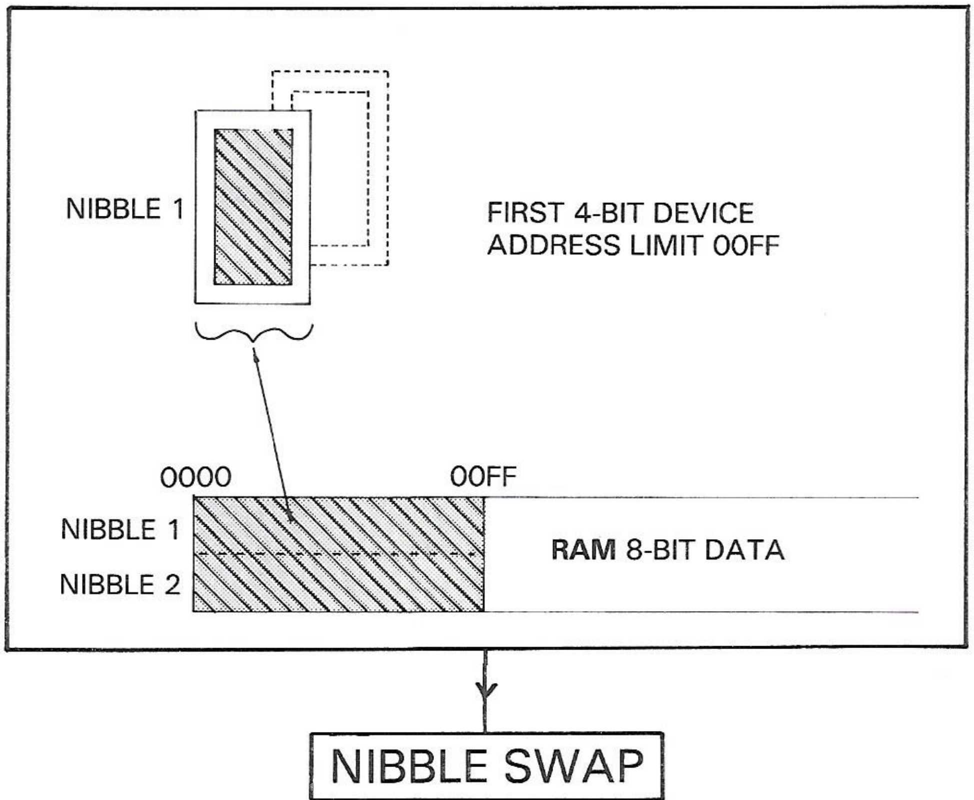
PRESS RESET

Load

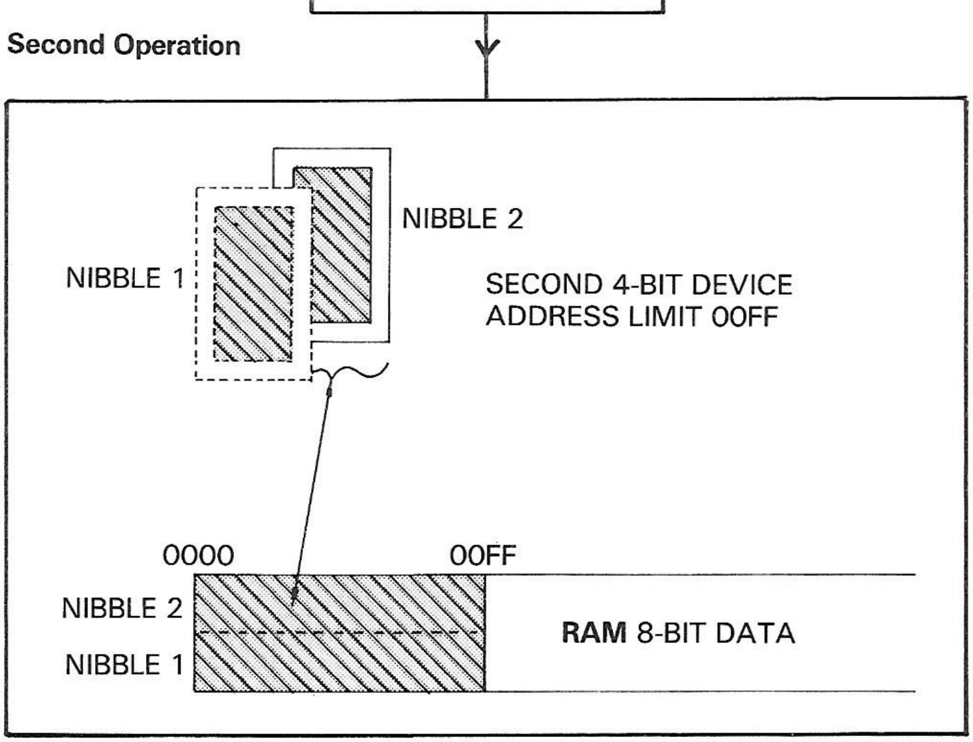
Loading of two 4-Bit devices into the RAM is a direct reversal of the programming sequence.

Increment on Two 4-Bit Devices by use of the Nibble Swap Function

First Operation



Second Operation



SECTION 4

4

stag

Sophisticated systems for the discerning engineer.

4 Device Functions

4.1 Load

Loading the RAM from a master device (8-bit).
For details on handling 16 and 32-bit data, see Section 3—Address Increment.

Insert master device into the appropriate socket as indicated by the red LED adjacent to Pin 1. Close socket. If in doubt as to the correct operation of ZIF sockets, refer to Section 1.7—Correct Operation of ZIF Sockets.

Press LOAD. The following will be displayed:

```
— LOAD MODE ————————————————————————— Zm3000 Rev X —

                                Device: INTEL INTELLIGENT
                                        2764

  RAM Start Address:   00 0000    PROM Start Address: 000000
  Address Increment:         1    Stop Address:       001FFF

                                  RESET . . . System Menu
                                  ENTER . . . Load Device
```

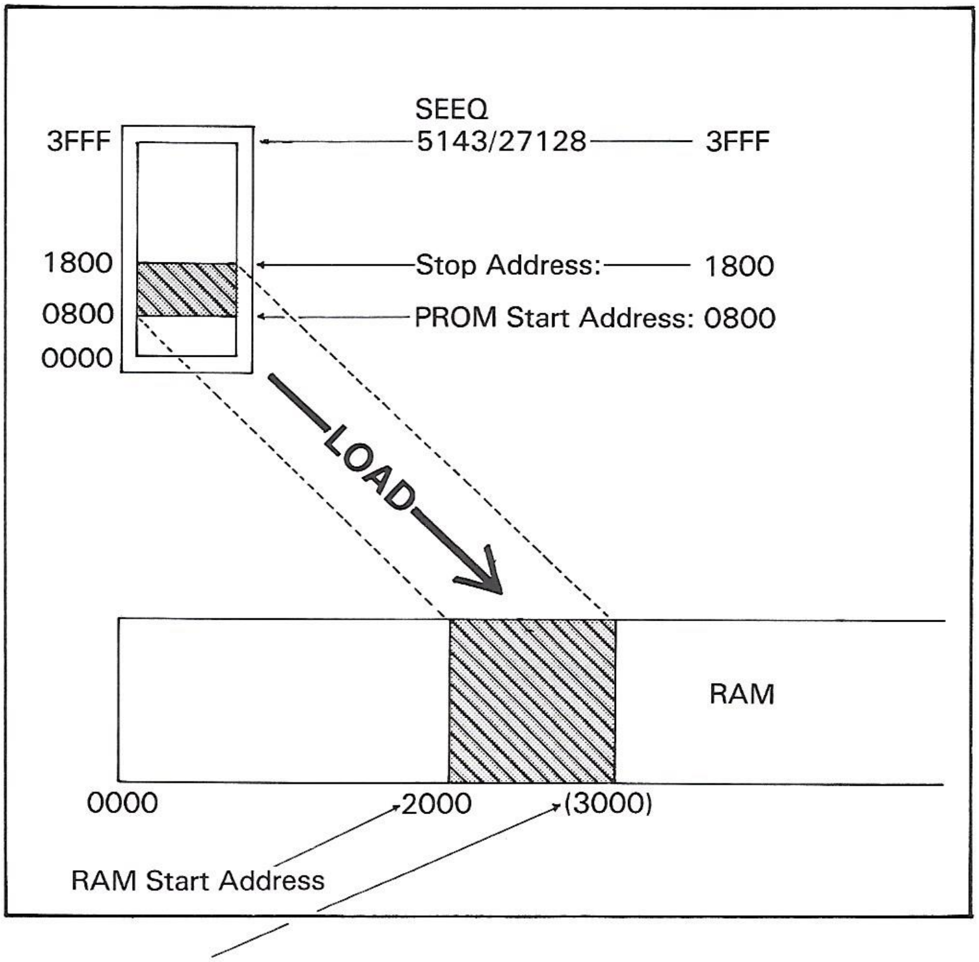
The display shows the RAM and PROM start addresses defaulted to zero, the PROM stop address defaulted to the size of the selected device and an address increment of 1. Any of these parameters can be selected using the four cursor keys and new values can be entered from the keyboard. To implement the load:

Press Enter.

Tests will be carried out to check for a reversed or faulty part and a connect test will be executed. If any of these fail, an error message will be displayed. Correct the fault and press Enter to carry out the Load using the same set of parameters.

On successful completion of the Load, the message 'Device Loaded' and a checksum will be displayed.

Examples of the address limits for the master device and the RAM can be shown in diagram form:



No stop address is entered for the RAM. Data loaded will automatically default to the size of the data block specified within the device.

4.2 Test Operations

The program sequence incorporates all the various test modes that may be required by the user prior to, during and after programming of a device.

The device tests fall into three categories:

(i) MANUAL
EMPTY and VERIFY—using the TEST key.

(ii) **OPTIONAL PRE-PROGRAM**
ILLEGAL BIT, EMPTY or NONE—select in SET SYSTEM mode.

(iii) **AUTOMATIC IN-PROGRAM**
VERIFY, PROGRAM FAIL and CHECKSUM—always performed at the end of a device PROGRAM operation.

4.3 Empty Test (Manual)

An EMPTY TEST can be applied to the device in the appropriate ZIF socket at any time. To do this:

PRESS TEST

This will show the TEST MODE display:

```
— TEST MODE ——————————————————————— Zm3000 Rev X —

                                 Device: INTEL INTELLIGENT
                                              2764

   Test Type:  [EMPTY]      VERIFY

   RAM Start Address:   00 0000    PROM Start Address: 000000
   Address Increment:         1    Stop Address:       001FFF


                                   RESET . . . System Menu
                                   ENTER . . . Test Device
```

Either the VERIFY or the EMPTY TEST will be indicated by a flashing cursor. Interchange between these two test modes is made by use of the horizontal cursor keys.

If the EMPTY TEST is applied to the whole device then stop and start addresses can remain in their default state as above.

To implement the EMPTY TEST:

PRESS ENTER

The device will be examined for its unprogrammed state throughout the specified area as defined by the start and stop addresses. The unprogrammed state could be either 'FF' or '00' for an 8-bit device or 'F' or '0' for a 4-bit device.

During the EMPTY TEST a two digit counter will increment to show that the operation is in progress.

On completion the display will show a checksum

For example:

```
— TEST MODE ————————————————————————— Zm3000 Rev X —


                                  Device: INTEL INTELLIGENT
                                               2764

  Test Type:   [EMPTY]       VERIFY

  RAM Start Address:   00 0000    PROM Start Address: 000000
  Address Increment:         1    Stop Address:       001FFF



  Device Empty                       RESET . . . System Menu
  Checksum: 2000                     ENTER . . . Test Device
```

To return to the SYSTEM MENU:

PRESS RESET

Empty Test: Fail

If a device were to fail the EMPTY TEST the display would show:

i. up to eight locations where the discrepancy occurs (these will be the first eight locations in the device where data was detected).
ii. the unblown state of that device.
iii. the actual data within each of the device locations.

For example:

```
— TEST MODE ———————————————————————————— Zm3000 Rev X —

                                            Device: FUJITSU
                                                    27C256

  Address     Empty State      PROM Data
   0047           FF              AC
   0084           FF              BD
   0164           FF              F3
   034A           FF              12
   035A           FF              02
   0597           FF              88
   05DD           FF              CE
   0600           FF              44     RESET  . . . System Menu
                                         ENTER  . . . Test Device
   [Not Empty]                             ↓    . . . Next Failure
```

Pressing the down cursor key will show the next fail locations if any are present. By continually pressing this key the user can scan every location in the device.

Empty Test: Stop and Start Address

The EMPTY TEST is applied to the device alone therefore, the "Address Increment" and the "RAM Start Address" are superfluous features.

Only the "PROM Start Address" and "Stop Address" are relevant. In this way the 'empty state' scan of the PROM can be made within pre-selected limits.

These can be entered by use of the cursor keys to select the parameters and by use of the hex-keyboard to enter the new address.

For instance:

```
— TEST MODE ——————————————————————————— Zm3000 Rev X —

                                   Device: INTEL INTELLIGENT
                                                2764

   Test Type:    EMPTY        VERIFY

   RAM Start Address:   00 0000    PROM Start Address: 000100
   Address Increment:         1    Stop Address:       000500



   Device Empty                       RESET . . . System Menu
   Checksum: FCFF                     ENTER . . . Load Device
```

To implement the EMPTY TEST:

PRESS ENTER

To return to the System Menu:

PRESS RESET

4.4 **Verify Test (Manual)**

The manual VERIFY TEST can be applied at any time, e.g. after LOAD as well as PROGRAMMING.

It can be entered by first pressing the TEST key to show the TEST MODE display:

— TEST MODE ————————————— Zm3000 Rev X —

Device: INTEL INTELLIGENT 2764

Test Type: EMPTY [VERIFY]

RAM Start Address: 00 0000   PROM Start Address: 000000
Address Increment: 1   Stop Address: 001FFF

Bit Mask: FF

RESET . . . System Menu
ENTER . . . Load Device

Either the EMPTY or the VERIFY test will be indicated by a flashing cursor. Interchange between these two modes is made by use of the horizontal cursor keys.

To implement the VERIFY TEST:

PRESS ENTER

During the check a two digit counter will increment to show that the operation is in progress.
On completion the display will show a checksum of all verified data within the device.

For example:

```
— TEST MODE ————————————————————————— Zm3000 Rev X —

                                   Device: INTEL INTELLIGENT
                                                2764

   Test Type:   EMPTY      [VERIFY]

   RAM Start Address:   00 0000    PROM Start Address: 000000
   Address Increment:         1    Stop Address:       001FFF

   Bit Mask: FF

   Device Verified                    RESET . . . System Menu
   Checksum: 6A65                     ENTER . . . Load Device
```

Verify Test: Address increment and stop and start addresses.

These can be entered prior to the test by use of the cursor keys to select the parameters and by use of the hex-keyboard to enter the new address.

Verify Test: Fail

If a verify fail were to occur the display would show similar information to that of the in-program verify:

TEST MODE — Zm3000 Rev X

Device: FUJITSU
27128

Address	RAM data	PROM Data
0030	A6	FF
0033	F6	FF
0053	26	22
0056	19	11
00B5	36	BC
00BA	68	E8
00BF	68	D4
0104	6E	66

Verify Fail

RESET . . . System Menu
ENTER . . . Test Device
↓ . . . Next Failure

Continually pressing the down cursor key will implement this scan of the entire device address range.

Verify "Bit Mask"

This is a feature whereby one or more bits of the 8 bit data can be masked (made irrelevant) during a device/RAM VERIFY scan of what would normally be 8-Bit words.

The BIT MASK is entered as a two digit hexadecimal number, which is converted by the PPZ into its fundamental binary state. This can be translated as the BIT MASK key.

This is best shown by using an example for instance: 3F

The Bit MASK is entered as a two digit hexadecimal figure: → 3F

The PPZ converts this figure into its binary state: → 0011 1111

The binary state can be translated into the key: → I I RR RRRR

I = Irrelevant bits
R = Relevant bits

Whereby 0's represent irrelevant bits and 1's represent relevant bits

3F effectively reduces the normal 8-Bit word to 6-Bits.

This means that the two most significant bits of the 8-Bit word can be either 0 or 1.

Therefore, the VERIFY TEST, with a BIT MASK of 3F will pass, so long as the remaining 6-bits are identical for all corresponding device/RAM locations within the specified address limits.

For example:

(A) **BIT MASK: 3F**

Device Data: → 01 | 11 0110 = 7 6
Corresponding RAM Data: → 10 | 11 0110 = B 6
} PASS

Irrelevant bits

Identical device/RAM data therefore verify pass.

(B) **BIT MASK: 3F**

Device Data: → 11 | 00 1101 = C D
Corresponding RAM Data: → 10 | 00 1101 = 8 D
} PASS

Irrelevant bits

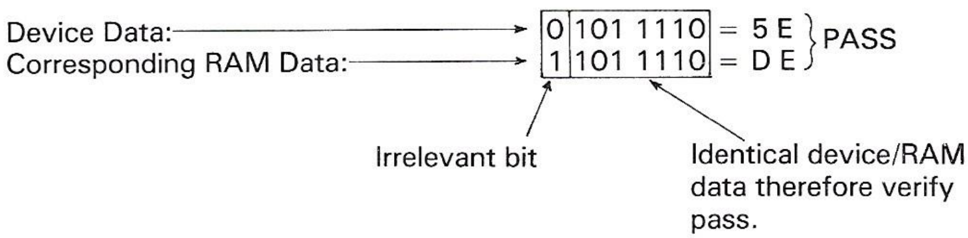
Identical device/RAM data therefore verify pass.

7F tends to be the most commonly required BIT MASK as it is associated with the ASCII code.

7F = 0111 1111 = IRRR RRRR = 7-BIT WORD
I = Irrelevant Bits R = Relevant Bits

For example:

BIT MASK: 7F

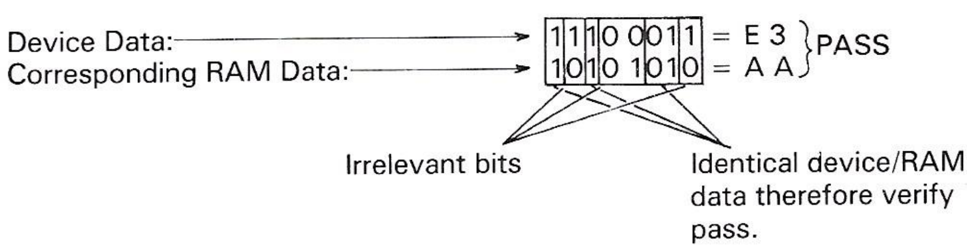


More unorthodox BIT MASKS can be entered if required for instance: A6

A6 = 1010 0110 = RIRI IRRI
I = Irrelevant bits R = Relevant bits

For example:

BIT MASK: A6



The most extreme version of a BIT MASK is 00, whereby every data bit in an 8-Bit word becomes irrelevant. Therefore VERIFY will always pass whatever the differences are between the RAM data and the device data.

00 = 0000 0000 = IIII IIII
I = Irrelevant bits

For example:
BIT MASK: 00

Device Data: 0101 1101 = 5 D
Corresponding RAM Data: 1010 0010 = A 2
PASS

Irrelevant bits

The bit mask can be selected prior to a verify by using the cursor keys and entering a new value from the keyboard.

4.5 Security Fuses and Encryption (Micros Only).

Security Fuses

Certain devices such as the 8751H have a security fuse. If this fuse (Fuse 1) is blown as part of the programming sequence, the device will function, but the data cannot be read and no further programming of the device can be carried out. For devices with two security fuses such as the 87C51, operation is slightly different. Blowing Fuse 1 will allow the device data to be read but will inhibit further programming and blowing Fuse 2 will not allow the device data to be read.

To select the security fuse status:

Press Key A followed by Key 3. This will display the following:

```
— Program/Test Control ———————————————— Zm3000 Rev X —

                                        INTEL Intelligent
                                        87C51

  Pre-Program check:      NONE     EMPTY     [ILLEGAL BIT]
  Verify Bit Mask:        FF

  Security Fuse State:
       [Fuse 1,2 Intact]      Fuse 1 Blown Fuse 2 Intact
        Fuse 1,2 Blown        Fuse 1 Intact Fuse 2 Blown

  Program Encryption Table:   [NO]    YES

                                   RESET . . . System Menu
                                   ENTER . . . Set System
```

Press the down cursor key ↓ until the cursor on 'Fuse 1,2 Intact' is flashing. This is the default state. Selection of any of the options* is made using the right cursor key →. To back track, use the left cursor key ←.

Press Enter to confirm the selection.

Press Reset to return the System menu.

***Note:**

Devices with one fuse

Fuse 1,2 Intact = Device can be read and programmed.
Fuse 1 Blown Fuse 2 Intact = Device cannot be read or programmed.
Fuse 1,2 Blown = Not Applicable.
Fuse 1 Intact Fuse 2 Blown = Not Applicable

Devices with two fuses

Fuse 1,2 Intact = Device can be read and programmed.
Fuse 1 Blown Fuse 2 Intact = Device can be read but not programmed.
Fuse 1,2 Blown = Device cannot be read or programmed.
Fuse 1 Intact Fuse 2 Blown = Not Applicable

Encryption (Micros Only).

Devices such as the 87C51 and 8752BH support a data encryption facility. This enables data within the device to be 'exclusively NORed' with a 32 byte encryption table before being read. The encryption code is entered into the programmer's memory immediately after the data to be programmed.

For example: 8752BH

Device Address Lo:	0000h
Device Address Hi:	1FFFh
RAM Address Lo:	0000h
32 Byte Encryption Table:	2000h - 2031h (inclusive)

or:

Device Address Lo:	0000h
Device Address Hi:	0FFFh
RAM Address Lo:	0000h
32 Byte Encryption Table:	1000h - 1031h (inclusive).

To select encryption:

Press Key A followed by Key 3. This will display the following:

```
— Program/Test Control ——————————————————— Zm3000 Rev X—

                                          INTEL Intelligent
                                          87C51

   Pre-Program check:     NONE    EMPTY   [ILLEGAL BIT]
   Verify Bit Mask:       FF
   Security Fuse State:
        [Fuse 1,2 Intact]      Fuse 1 Blown Fuse 2 Intact
         Fuse 1,2 Blown        Fuse 1 Intact  Fuse 2 Blown

   Program Encryption Table:   [NO]   YES

                                    RESET . . . System Menu
                                    ENTER . . . Set System
```

Press the down cursor key ↓ until the cursor on 'Program Encryption Table: NO' is flashing. This is the default state. To select YES* use the right cursor key →. To deselect encryption, use the left cursor key ←.

Press Enter to confirm the selection.

Press Reset to return the System menu.

***Note:**

After the encryption table has been blown into a device, the device will fail any subsequent attempts to verify it.

4.6 **Pre-Program Checks**

Pre-Program checks are tests carried out on a device prior to programming. These checks should not be confused with the checks for a reversed or faulty device and the connect test.

There are three options available in the category of Pre-Program Checks. These are:

1) An illegal bit test (the default test)
2) An empty test
3) A third option where no Pre-Program Check is applied i.e. NONE.

The Pre-Program Checks are detailed on the following pages.

Illegal Bit: Pre-Program Check

The System 3000 automatically checks that the pattern already within the device is able to be programmed with the intended data from the RAM.

If a device were to fail the ILLEGAL BIT TEST the display would show:

i. up to eight locations where this discrepancy occurs (these will be the first eight locations in the device where a fault is detected).
ii. the data from the RAM destined for each location and
iii. the data within the device that has caused the failure.

For example:

```
— PROGRAM MODE ———————————————————— Zm3000 Rev X —

                                         Device: HITACHI
                                                 27256

  Address     RAM Data     PROM Data
   0003          91           59
   0023          47           AE
   0043          66           44
   004C          6A           AA
   0056          98           F6
   005D          50           21
   00B5          4F           87
   0129          35           98        RESET  . . . System Menu
                                        ENTER  . . . Test Device
   [ Illegal Bit ]                        ↓    . . . Next Failure
```

Pressing the down cursor key will show the next fail locations if any are present. By continually pressing this key the user can scan every location in the device.

Empty: Pre-Program Check

This is an automatic scan to ensure the EMPTY STATE of the device prior to programming.

It is basically the same as the previously described 'manual' empty test but differs in two ways:

i. the device will automatically undergo programming should it pass the test, therefore no checksum of the unblown state is displayed and

ii. the stop and start addresses are dictated by those entered in the PROGRAM MODE.

None: No Pre-Program Check

The third option is a lack of any automatic pre-program checks. However a 'PROGRAM FAIL' will still be indicated if incompatible data is received by the device at a programmable location.

If a part to be programmed is faulty or badly socketed, the display will indicate the malfunction.

For example:

```
— PROGRAM MODE ——————————————————— Zm3000 Rev X —

                               Device: HITACHI
                                       27256


          | REVERSED OR FAULTY PROM |


                               RESET  . . . System Menu
                               ENTER  . . . Test Device
                                 ↓    . . . Next Failure
```

Program Fail

If a PROGRAM FAILURE occurs the display will show:

i. the first location where a discrepancy has occurred (unlike other fails only one location at a time will be displayed).
ii. the data within the RAM at that location and
iii. the data within the PROM at that location

For example:

```
— PROGRAM MODE ——————————————— Zm3000 Rev X —

                                   Device: HITACHI
                                           27256

  Address     RAM Data     PROM Data

   01F2          39           20


                                   RESET  . . . System Menu
  [Program Fail]                   ENTER  . . . Test Device
                                     ↓    . . . Next Failure
```

Selection of Pre-Program Checks

From the main System Menu:

Press Key A followed by Key 3 to display:

```
— Program/Test Control ———————————————— Zm3000 Rev X —

                                         INTEL Intelligent
                                         87C51

  Pre-Program check:      NONE    EMPTY   [ILLEGAL BIT]
  Verify Bit Mask:        FF

  Security Fuse State:
       [Fuse 1,2 Intact]     Fuse 1 Blown Fuse 2 Intact
        Fuse 1,2 Blown       Fuse 1 Intact  Fuse 2 Blown

  Program Encryption Table:   [NO]    YES

                                    RESET . . . System Menu
                                    ENTER . . . Set System
```

This display is for microcomputers. As far as selecting Pre-Program Checks is concerned, this display and the selection process and the one for Proms and Eproms are identical.

A flashing cursor will be seen on the default setting 'Illegal Bit'. Interchange between the options can be made using the horizontal cursor keys ← and →.

Press Enter to confirm the selection.

Press Reset to return the System Menu.

4.7 Program

Programming a target device from RAM data (8-bit).
For details on handling 16 and 32-bit data, see Section 3—Address increment.

Insert target device into the appropriate socket as indicated by the red LED adjacent to Pin 1. Close socket. If in doubt as to the correct operation of ZIF sockets, refer to Section 1.7—Correct Operation of ZIF Sockets.

Press PROG. The following will be displayed:

```
— PROGRAM MODE ————————————————————— Zm3000 Rev X —

                                  Device: INTEL INTELLIGENT
                                               2764

  Check:   NONE     EMPTY     [ILLEGAL BIT]

  RAM Start Address:   00 0000     PROM Start Address: 000000
  Address Increment:         1     Stop Address:       001FFF


                                     RESET . . . System Menu
                                     ENTER . . . Load Device
```

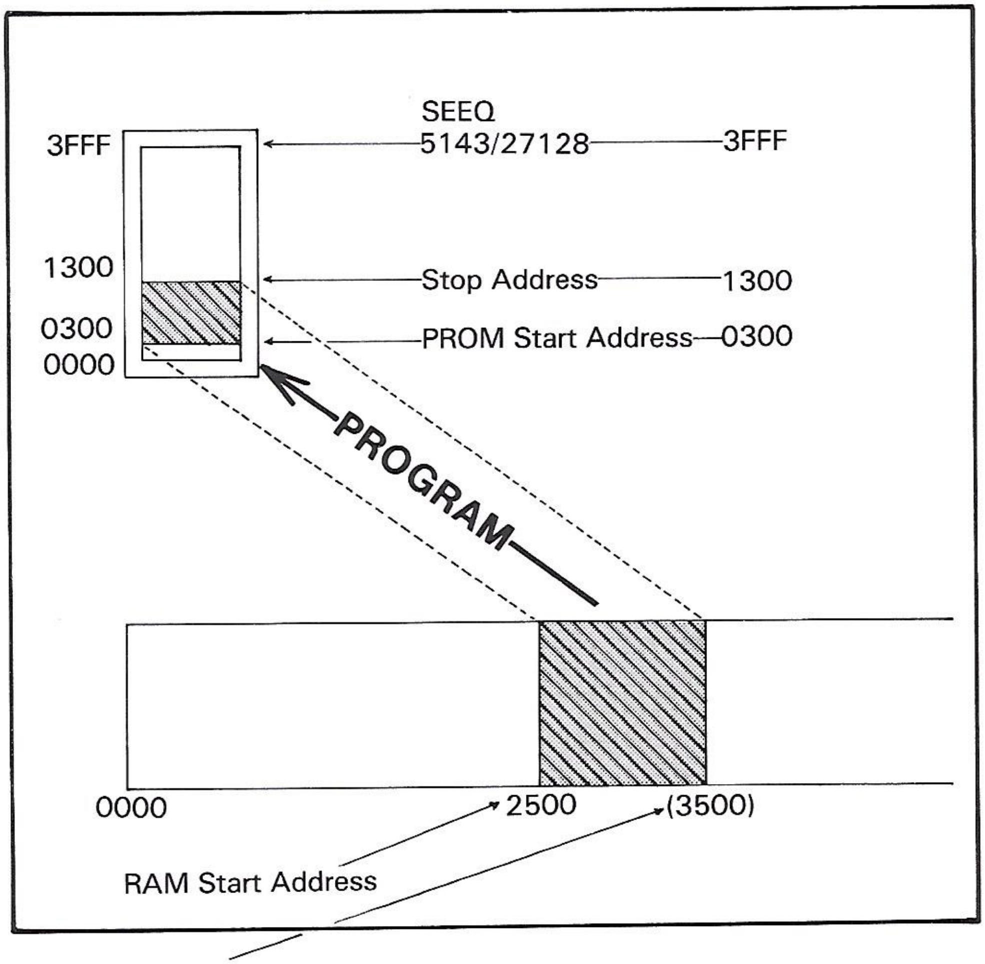
The display shows the RAM and PROM start addresses defaulted to zero, the PROM stop address defaulted to the size of the selected device and an address increment of 1. Any of these parameters can be selected using the four cursor keys and new values can be entered from the keyboard. To implement the Program:

Press Enter.

Tests will be carried out to check for a reversed or faulty part and a connect test will be executed. If any of these fail, an error message will be displayed. Correct the fault and press Enter to carry out the Program using the same set of parameters.

On successful completion of the Program, the message 'Device Programmed' and a checksum will be displayed.

Examples of the address limits for the master device and the RAM can be shown in diagram form:



No stop address is entered for the RAM as data programmed will automatically default to the size of the data block specified within the device.

In-Program Verify

A feature whereby each device location programmed is immediately checked to see that it is identical to the incoming data bytes in the RAM. However, only after the two digit program counter has incremented over the entire pre-selected address range will VERIFY FAIL data be displayed.

If this were to happen the display would show:

i up to eight locations in the device where a discrepancy has occured. These will be the first eight locations where the device and the RAM data are incompatible.
ii the data from the RAM in comparison to data in the device locations and
iii the data in the device locations.

For instance:

```
— PROGRAM MODE ——————————————————————— Zm3000 Rev X —

                                   Device: INTEL INTELLIGENT
                                                2764

  Address        RAM Data        PROM Data
   0500             FF              00
   0501             FF              00
   0502             FF              00
   0503             FF              00
   0504             FF              00
   0505             FF              00
   0506             FF              00
   0507             FF              00
  [ Verify Fail ]                       RESET  . . . System Menu
                                        ENTER  . . . Test Device
                                          ↓    . . . Next Failure
```

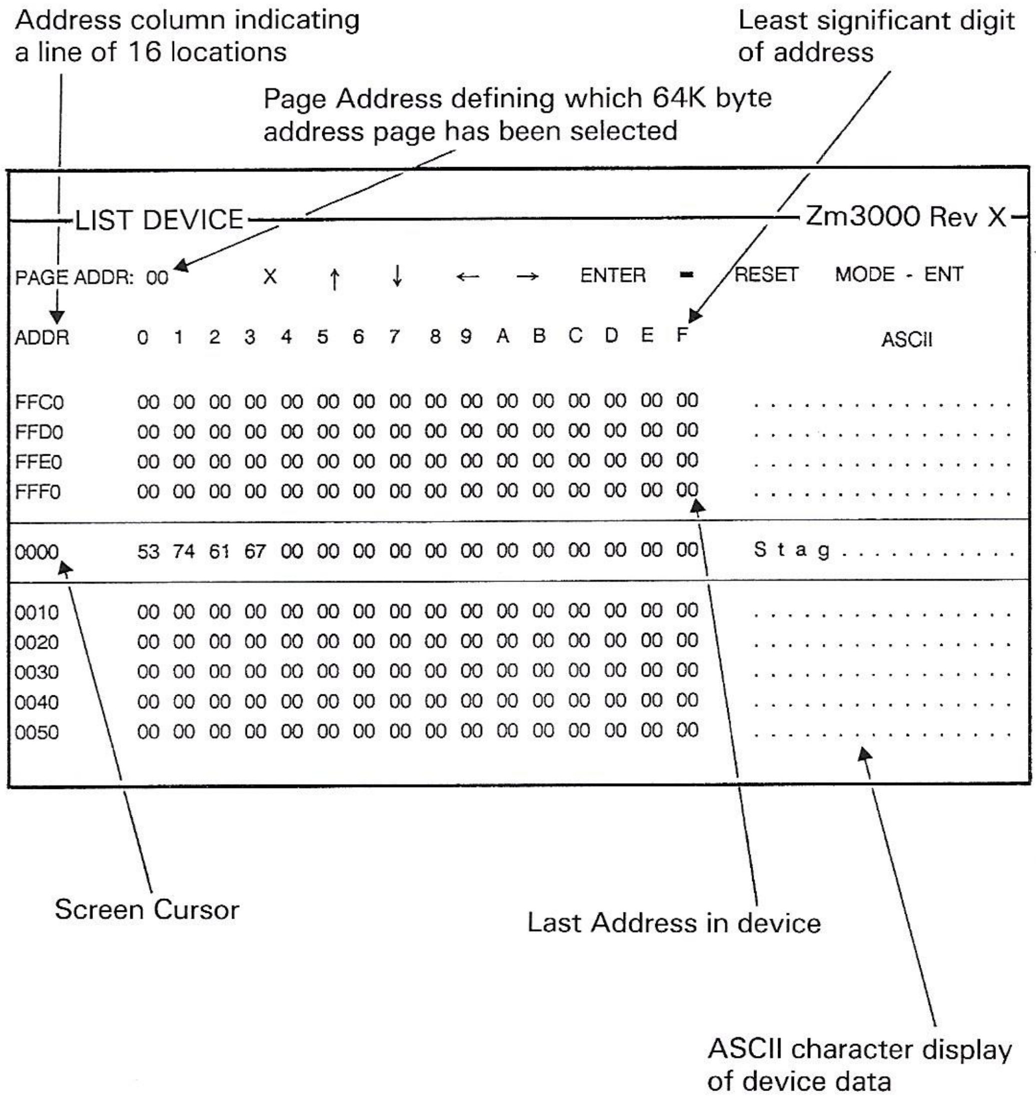
Pressing the down cursor key will re-start PROGRAMMING from a new start address defaulted to the next location above the last found VERIFY FAIL. Only after the counter has fully incremented over the remainder of the pre-selected address field will any new VERIFY FAILS be displayed.

4.8 **List Device**

This feature allows the data content of the device in the socket to be displayed and scanned on the screen.

From the System Menu:

Press Key 6. The following will be displayed:



Scanning a device data

To scan the data:

Press ↑ or ↓

The address will increment or decrement sixteen locations (one line) at a time. Alternatively an address can be entered from the keyboard and upon pressing Enter the specified area of device data will be displayed.

'Paged' Devices

To specify the required page, press Key H and enter in the page number, i.e. 01, 02, etc. Press Enter. The new page will be displayed.

List device displays data in hexadecimal form only. Pressing Key X will not display octal or decimal forms.

SECTION 5

5

stag

Sophisticated systems for the discerning engineer.

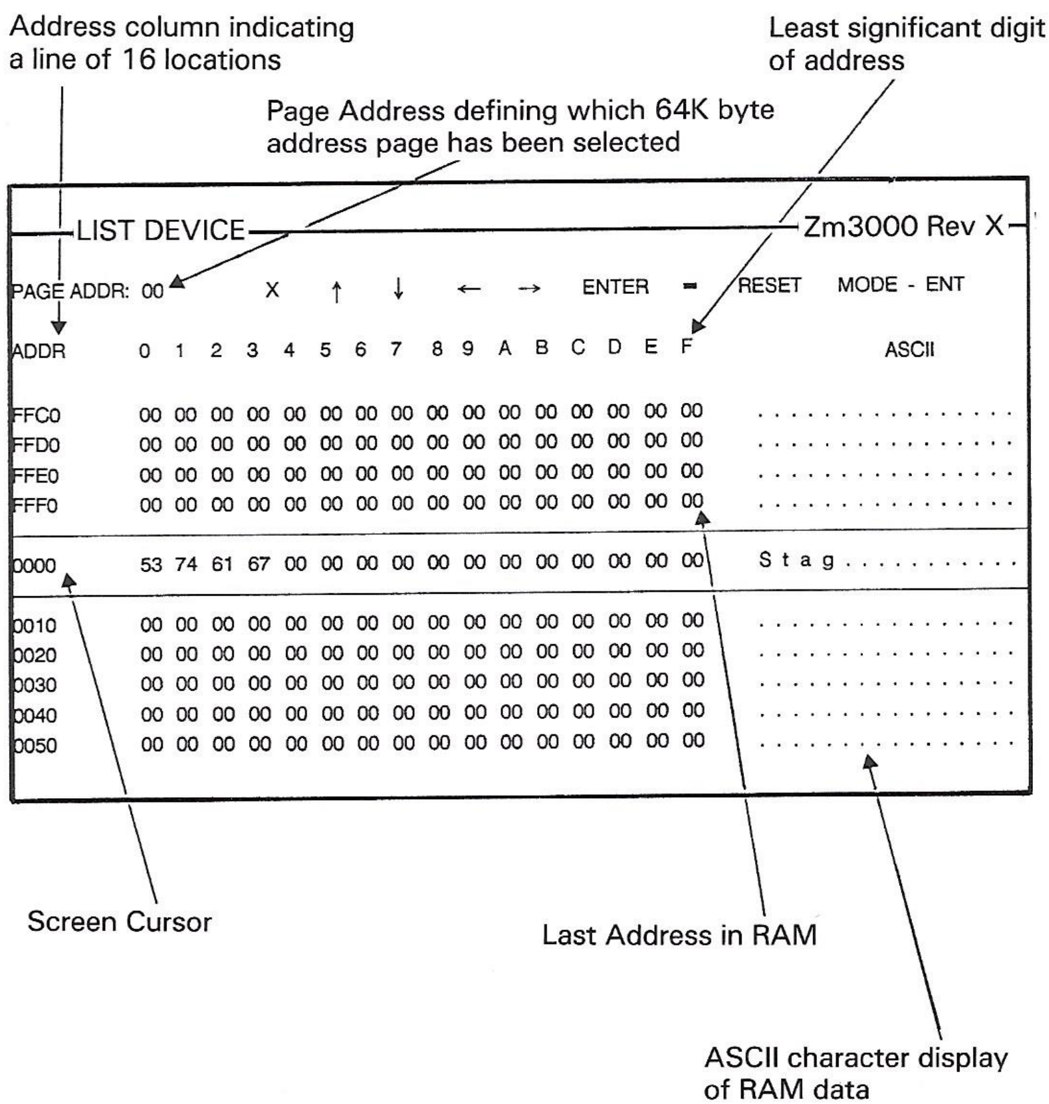
5 Ram Functions

5.1 List RAM

This feature allows the data content of the RAM to be displayed and scanned on the screen.

From the System Menu:

Press Key 5. The following will be displayed:



5.2 **Edit RAM**

This is a feature whereby the actual content of the RAM can be directly modified by input from the keyboard.

There are nine EDIT functions available. To display the options:

PRESS KEY 3

```
—— EDIT RAM DATA ——————————————— Zm3000 Rev X ——


   1 . . . Hex Editor               6 . . . Nibble Swap

   2 . . . Decimal Editor           7 . . . String Search

   3 . . . Octal Editor             8 . . . Block Move

   4 . . . Logical OR Data          9 . . . Complement

   5 . . . Logical AND Data         RESET . . . System Menu
```

Entry into a particular EDIT function from the sub-menu (shown above) is made by pressing the appropriate key.

For instance to enter into the String Search function:

PRESS KEY 7

5.2.1 **Hexadecimal, Decimal and Octal Editors**

Hexadecimal Editor

To enter the Hex. Editor, from the System Menu:

Press Key 3 followed by Key 1

The display will show a page of RAM data expressed in hexadecimal form on an address matrix 10 lines by 16 bytes in size. A screen cursor will show the default location 0000 on the sixth line up between the two parallel dash lines running horizontally across the screen (the edit line). At the right hand side of the screen there is an ASCII character representation of the data displayed in the adjacent matrix.

```
--- HEX. ENTER MODIFY DATA ----------------------------- Zm3000 Rev X ---

PAGE ADDR: 00     X   ↑   ↓   ←   →   ENTER   ▬   RESET   MODE - ENT

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

FFC0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
FFD0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
FFE0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
FFF0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
------------------------------------------------------------------------
0000   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
------------------------------------------------------------------------
0010   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
0020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
0030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
0040   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
0050   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
```

As with the List RAM data function, the vertical cursor keys will increment the address by one line (sixteen locations) at a time. To specify a further page of RAM data, press Key H and enter in the page number, i.e. 01, 02, etc. Press Enter. The new page will be displayed.

Individual bytes can be highlighted with the screen cursor. The cursor can be transferred from the address column to the data matrix by pressing Enter or →, and then moved around using any of the four cursor keys. To return the cursor to the address column, press Key L.

An alternative to using the cursor keys is direct entry of the address to be viewed. To do this, with the cursor on the address column, enter the address from the keyboard and press Enter.

The Hex Editor in common with the Decimal and Octal Editors has three operating modes:

Overwrite Mode	—	ENT
Delete Mode	—	DEL
Insert Mode	—	INS

Overwrite Mode is the default setting. It allows any selected byte of data to be overwritten by new data input from the keyboard.

Delete Mode allows the data byte at the specified location to be deleted and all data above that location to be repositioned one location lower within the RAM.

Insert Mode allows all data at and above a selected location to be repositioned one location higher in the RAM. This therefore leaves a 'blank' location in which to enter new data.

To select any of these modes, the cursor must be initially on the address column. Interchange between the modes is made by pressing Key L. The selected mode is indicated in the top right hand corner of the screen, i.e. ENT, INS or DEL.

To implement an INSert or DELete: after selecting the address, press Enter.

To display the RAM data in either Decimal or Octal form: Press Key X (once for decimal and twice for octal).

To return to the Edit RAM Data sub-menu: Press Key -.

To return to the Main System Menu: Press Reset.

Decimal Editor

To enter the Decimal Editor from the System Menu:

Press Key 3 followed by Key 2.

The display will show a page of RAM data expressed in decimal form on an address matrix 10 lines x 10 columns in size. The information can be scanned and altered in a similar manner to the Hex Editor.

Octal Editor

To enter the Octal Editor from the System Menu:

Press Key 3 followed by Key 3

The display will show a page of RAM data expressed in octal form on an address matrix 10 lines x 8 columns in size. The information can be scanned and altered in a similar manner to the Hex and Decimal Editors.

5.2.2 **Logical OR Data**

The 'OR' function

This is a modification function where one of three 'OR' equations is applied to an individual bit within a RAM byte. The type of equation applied is dependent upon the pattern discrepancy of bits that constitute the entered 'OR MASK', compared to corresponding bits within the RAM bytes.

OR equations

NB: In logic equations the OR function is usually represented by a plus sign.

(i) Remain as one

If a RAM bit was 'one' it will stay as 'one' no matter what the corresponding OR MASK might be, ie:

RAM bit + corresponding OR MASK bit = Same RAM bit

1	+	1	=	1
1	+	0	=	1

(ii) Remain as zero

A RAM bit will remain as 'zero' only if the corresponding OR MASK bit is also a 'zero', ie:

RAM bit 2 corresponding OR MASK bit = Same RAM bit

0	+	0	=	0

(iii) Changed to one

A RAM bit can be changed to a 'one' from a 'zero' if the corresponding OR MASK bit is a 'one', ie:

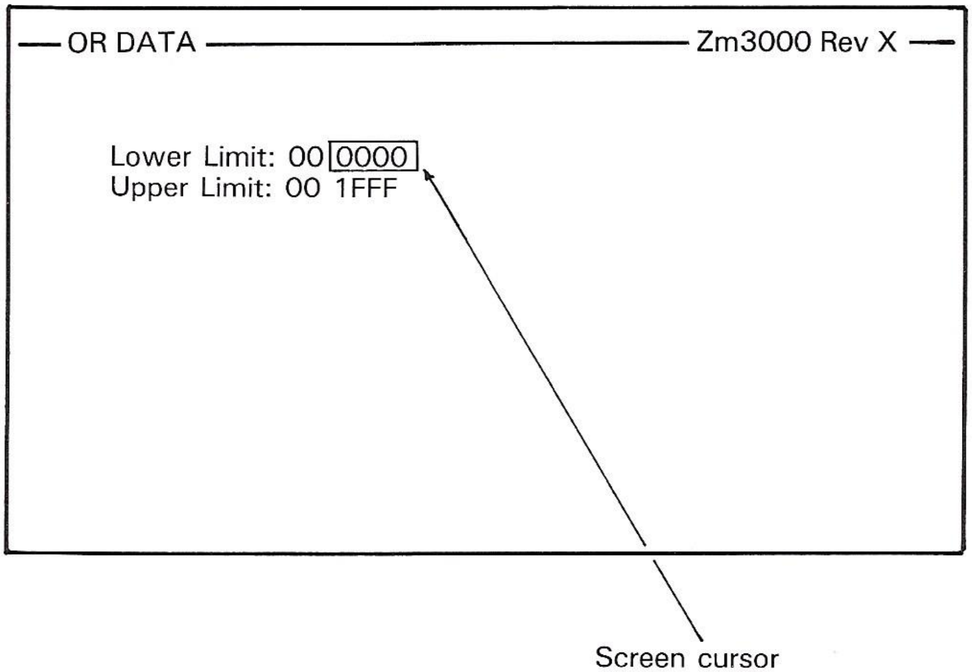
RAM bit + corresponding OR MASK bit = Changed RAM bit

0	+	1	=	1

Therefore point (iii) alone is the cause of any modification to existing RAM data bytes.

To use the 'Logical OR Data' function, from the SYSTEM MENU:

PRESS KEY 3 followed by KEY 4 to display:



Interchange between the limits is made by using the vertical cursor keys and new values can be entered from the hex-keyboard.

To confirm the address limits:

PRESS ENTER

'Address Increment' defaulted to 01 will appear in reverse video. The increment may be altered up to a value of 08. (For further details, see Section 3). To confirm the 'Address Increment':

PRESS ENTER

'Select input form' will start to flash beside the choice of ASCII and Numeric.

Selection of ASCII Form

PRESS KEY ←

The cursor will move to 'ASCII'

PRESS ENTER to display:

```
— OR DATA ———————————————————————————— Zm3000 Rev X ——

Lower Limit: 00 0000      ASCII              :  Select input form
Upper Limit: 00 1FFF      Address Increment:01

[String:]

□ ! " £ $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] _ '
a b c d e f g h i j k l m n o p q r s t u v w x y z { 1 }

——— Run Function ————————————  ———————— Delete ———————————
```

Screen cursor in default position on Space character

To select a particular character, position the cursor over it by using all of the cursor keys and PRESS ENTER.

Characters will appear next to 'String' (up to 22) and may be deleted as required by moving the cursor onto the 'Delete' line and PRESSING ENTER.

To implement the function, move the cursor onto the Run Function line and PRESS ENTER. This will also return the EDIT RAM DATA sub-menu.

To return to the SYSTEM MENU:

PRESS RESET

Selection of Numeric Form

The default input form is 'Numeric':

PRESS ENTER to display:

```
—— OR DATA ——————————————————————— Zm3000 Rev X ——

   Lower Limit: 00 0000              Numeric: Select input form
   Upper Limit: 00 1FFF              Address Increment: 01

   [String:] 00

   Run Function
```

The numeric string will be in its default state of 00 and a new value may be entered by use of the hex-keyboard.

To enter a string value of more than one byte:

PRESS ENTER after each selected byte.

To delete a byte as required:

PRESS KEY ←

To implement the function:

PRESS ↓ followed by ENTER

This will return the EDIT RAM DATA sub-menu

To return to the SYSTEM MENU:

PRESS RESET

5.2.3 **Logical AND Data**

The 'AND' function

This is a modification function where one of three AND equations is applied to an individual bit within a RAM byte. The type of equation applied is dependent upon the pattern discrepancy of bits that constitute the entered AND MASK, compared to corresponding bits within the RAM bytes.

AND equations

NB: In logic equations the AND function is represented by a dot.

i. Remain as zero

If a RAM bit was 'zero' it will stay as 'zero' no matter what the corresponding AND MASK bit might be, ie:

RAM bit ● corresponding AND MASK bit = Same RAM bit

0	●	0	=	0
0	●	1	=	0

ii. Remain as one

A RAM bit will remain as 'one' if the corresponding AND MASK bit is also a 'one' ie:

RAM BIT ● corresponding AND MASK bit = Same RAM bit

1	●	1	=	1

iii. Changed to one

A RAM bit can be changed to 'zero' from a 'one' if the corresponding AND MASK bit is a 'zero', ie:

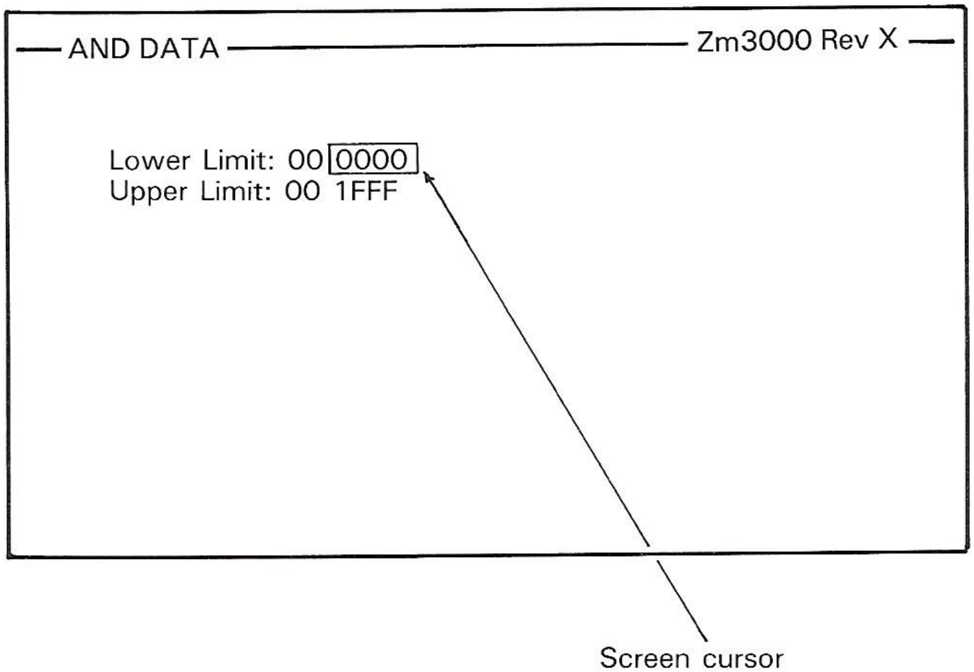
RAM bit ● corresponding AND MASK bit = Changed RAM bit

1	●	0	=	0

Therefore point (iii) alone, is the cause of any modification to existing RAM data bytes.

To use the 'Logical AND Data' function, from the SYSTEM MENU:

PRESS KEY 3 followed by KEY 5 to display:



Interchange between the limits is made by using the vertical cursor keys and new values can be entered from the hex-keyboard.

To confirm the address limits:

PRESS ENTER

'Address Increment' defaulted to 01 will appear in reverse video. The increment may be altered up to a value of 08. (For further details, see Section 3). To confirm the 'Address Increment':

PRESS ENTER

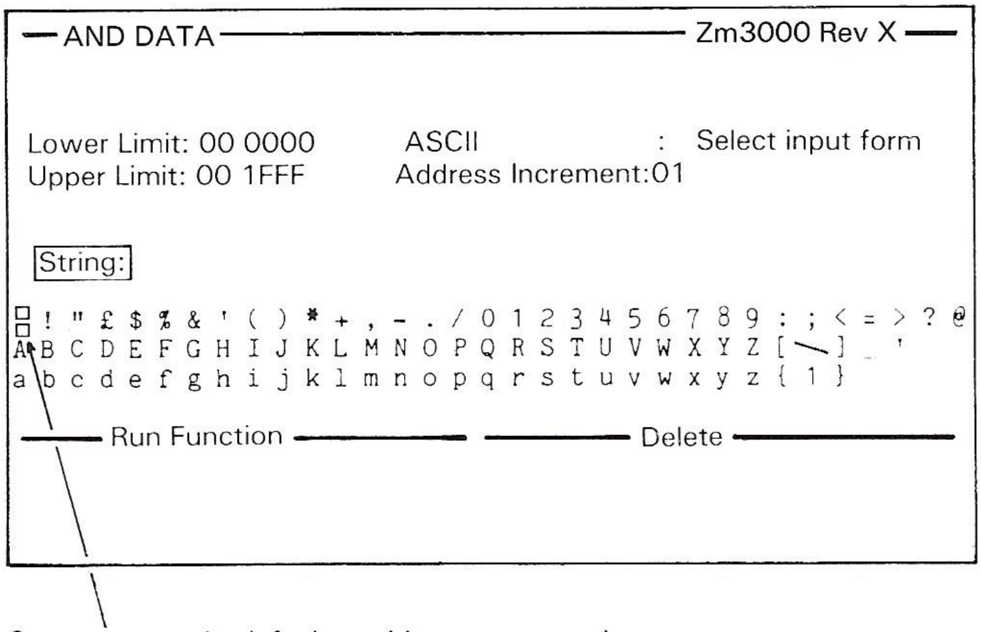
'Select input form' will start to flash beside the choice of ASCII and Numeric.

Selection of ASCII form

PRESS KEY ←

The cursor will move to 'ASCII'

PRESS ENTER to display:



Screen cursor in default position on space character

To select a particular character, position the cursor over it by using all of the cursor keys and PRESS ENTER.

Characters will appear next to 'String' (up to 22) and may be deleted as required by moving the cursor onto the 'Delete' line and PRESSING ENTER.

To implement the function, move the cursor onto the RAM function line and PRESS ENTER. This will also return the EDIT RAM DATA sub-menu.

To return to the SYSTEM MENU:

PRESS RESET

Selection of Numeric form

The default input form is 'Numeric':

PRESS ENTER to display:

```
— AND DATA ——————————————— Zm3000 Rev X —

Lower Limit: 00 0000            Numeric: Select input form
Upper Limit: 00 1FFF            Address Increment: 01

[String:] 00

Run Function
```

The numeric string will be in its default state of 00 and a new value may be entered by use of the hex-keyboard.

To enter a string value of more than one byte:

PRESS ENTER after each selected byte.

To delete a byte (as required):

PRESS KEY ←

To implement the function:

PRESS ↓ followed by ENTER

This will return the EDIT RAM DATA sub-menu.

To return the system menu:

PRESS RESET

5.2.4 **Nibble Swap** (See also Section 3.5)

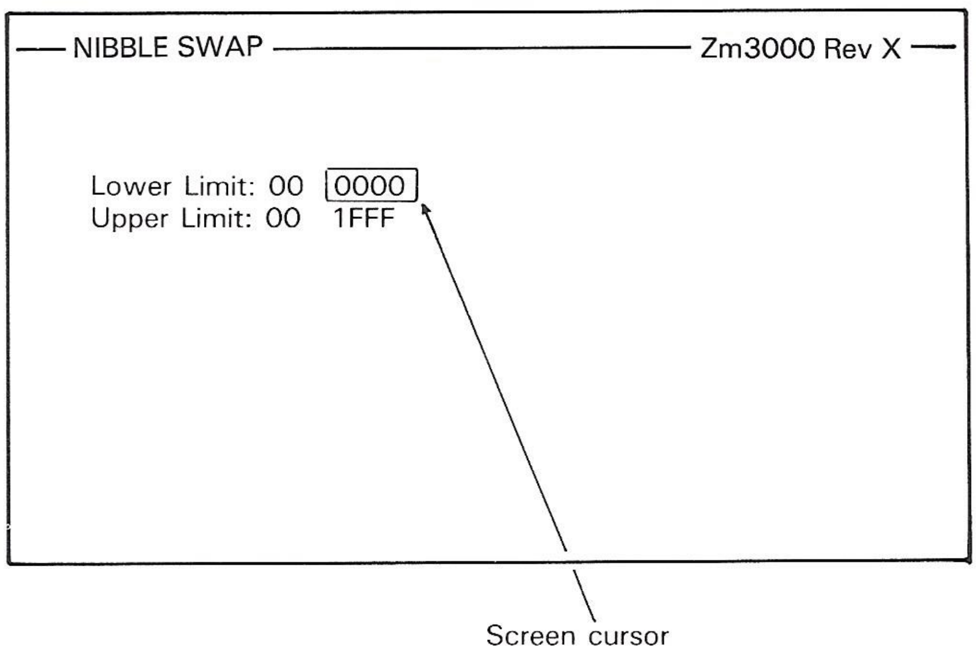
Two 4-bit devices can be programmed in two operations from a common address range within the RAM in such a way as to contain 8-Bit data when combined.

The address data will be 'split' during this programming sequence into two separate nibbles of data.

To enter the Nibble Swap mode:

PRESS KEY 3 followed by KEY 6.

The display will show the default state for the device:



Interchange between the limits is made by using the cursor keys and new values can be entered from the hex-keyboard. To confirm the address limits:

PRESS ENTER

After confirming the Address Limits, 'Address Increment' defaulted to 01 will appear in reverse video. The increment may be altered up to a value of 08.

To confirm the address increment and implement the Nibble Swap:

PRESS ENTER

This will re-display the EDIT RAM DATA sub-menu.

To return the SYSTEM MENU:

PRESS ENTER

5.2.5 **String Search**

This is a feature which enables the user to search for a pattern of data in the RAM and to determine the address at which this data is stored.

A reference pattern can be input as a string of up to 22 hexadecimal data bytes or as a string of up to 22 characters from the ASCII register.

If a string is found, there is also a facility to display the page of RAM data within which the string is located.

To enter string search mode, from the SYSTEM MENU:

PRESS KEY 3 followed by KEY 7 to display:

```
— STRING SEARCH ————————————————— Zm3000 Rev X ——

      Lower Limit: 00  [0000]
      Upper Limit: 00   1FFF
```

Interchange between the RAM address limits can be made using the vertical cursor keys, and new limits can be input from the hex-keyboard. To confirm the limits:

Press Enter

After confirming the Address Limits, 'Address Increment' defaulted to 01 will appear in reverse video. The increment may be altered up to a value of 08.

Press Enter

'Select input form' will start to flash beside the choice of ASCII and Numeric.

Selection of ASCII form

Press ←

The cursor will move to 'ASCII'

Press ENTER

The display below appears on the screen:

```
—— STRING SEARCH ———————————————————————————— Zm3000 Rev X ——


 Lower Limit: 00 0000        ASCII              :  Select input form
 Upper Limit: 00 1FFF       Address Increment:01


 String:

□ ! " £ $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ]   '
a b c d e f g h i j k l m n o p q r s t u v w x y z { 1 } ~

——— Run Function ————————  ——————— Delete ———————————
```

To select a particular character or function, position the cursor over it by use of the arrow keys and press ENTER.

Characters will appear next to 'string' and may be deleted as required using the 'Delete' function.

'Run Function' implements the ASCII string search.

If a matching string is present within the pre-selected RAM limits then the display will show the location of the first byte of the first string found:

```
—— STRING SEARCH ———————————————————————————— Zm3000 Rev X ——


 Lower Limit: 00 0000        ASCII              :  Select input form
 Upper Limit: 00 1FFF


 String: SMT
 ↑ : Adjust specification    —: Return to sub-menu RESET: System Menu
 ENTER: Next occurence       E: View in the editor
 String found at: 00 0004
```

In this case the first match has been found at 00 0004.

To search for the next match:

PRESS ENTER

To view the match at its RAM location:

PRESS KEY E (to return to the current display PRESS —)

To adjust the specification for the string search:

PRESS KEY ↑

To return the RAM EDIT DATA sub-menu:

PRESS KEY —

To return to the SYSTEM MENU:

PRESS RESET

ASCII String Search: Failure

If a matching string is not found within the pre-selected RAM limits, then the display will show:

```
—— STRING SEARCH ——————————————————— Zm3000 Rev X ——

Lower Limit: 00 0000   ASCII       :  Select input form
Upper Limit: 00 1FFF

String: PROM
↑ : Adjust specification  :  Return to sub-menu RESET: System Menu


String not found in specified limits
```

The up arrow key gives the chance to alter the limits for the search.

Press RESET

to return to the system menu.

Selection of Numeric form

The default input form is 'Numeric'.

PRESS ENTER to display:

```
— STRING SEARCH ————————————————— Zm3000 Rev X —

  Lower Limit: 00 0000            Numeric: Select input form
  Upper Limit: 00 1FFF



 [String:] 00

  Run Function
```

The numeric string will be in its default state of 00 and a new value may be entered by use of the hex-keyboard.

To enter a string value of more than one byte:

PRESS ENTER after each selected byte.

To delete a byte (as required):

PRESS KEY ←

To implement the function:

PRESS ↓ followed by ENTER

For example:

```
—— STRING SEARCH ———————————————— Zm3000 Rev X ——

   Lower Limit: 00 0000              Numeric: Select Input form
   Upper Limit: 00 3FFF

  String: 53   74   61   67

↑ : Adjust specification —: Return to sub-menu RESET: System Menu
ENTER: Next occurrence E: View in the editor

   String found at: 00 012C
```

In this case the first match was found at 012C.

To search for the next match:

PRESS ENTER

To view the match at its RAM location:

PRESS KEY E (to return to the current display PRESS —)

To adjust the specification for the string search:

PRESS KEY ↑

To return the RAM EDIT DATA sub-menu:

PRESS KEY —

To return to the SYSTEM MENU:

PRESS RESET

Numeric String Search: Failure

If a matching string is not found within the pre-selected RAM limits, then the display will show:

```
—— STRING SEARCH ————————————————— Zm3000 Rev X ——

Lower Limit: 00 0000   ASCII      Numeric: Select input form
Upper Limit: 00 1FFF

String: PROM
↑ : Adjust specification ←:Return to sub-menu RESET: System Menu


String not found in specified limits
```

The up arrow key gives the chance to alter the limits for the search.

Press RESET

to return to the system menu.

5.2.6 **Block Move**

This is a feature enabling a block of data within pre-selected address limits to be re-located at another address within the RAM, without destroying the original data.

To enter BLOCK MOVE from the SYSTEM MENU:

PRESS KEY 3 followed by KEY 8

This will display:

```
— BLOCK MOVE ————————————————————— Zm3000 Rev X —

     Lower Limit: 00  [0000]
     Upper Limit: 00   3FFF
```

The display shows the upper and lower limits in their default states. The lower limit is defaulted to zero and the upper limit is defaulted to the size of the pre-selected device. These limits can be altered to define the size and location of the block of data to be moved.

5.2.7 **Complement**

Complement is a modification of RAM data made by inverting the bits within the bytes.

Complement will change 'zero' bits to 'ones' and 'one' bits to 'zeros' therefore transforming all RAM data bytes within pre-selected address limits.

The COMPLEMENT function may be entered from the SYSTEM MENU by doing the following:

PRESS KEY 3 followed by KEY 9 to display:

```
— COMPLEMENT ——————————————————— Zm3000 Rev X —

  Lower Limit: 00  [0000]
  Upper Limit: 00   3FFF
```

The address limits of the RAM to undergo the complement will be in their default states. The lower limit will be defaulted to zero and the upper limit to the size of the pre-selected device. Interchange between the two limits is made by use of the vertical cursor keys (↑ and ↓) and new selections can be entered from the hex-keyboard.

To confirm these selections:

PRESS ENTER

If an Address Increment other than 1 is required it can now be entered directly from the keyboard, and can take values from 1 to 8.

To confirm the Address Increment and implement the Complement:

PRESS ENTER

This will return the EDIT RAM DATA sub-menu. To return to the SYSTEM MENU:

PRESS RESET

5.3 Fill RAM

5.3.1 Fill RAM with 00s, FFs or the Empty State

This feature allows the RAM to be filled with 00, FF or the Empty State of the chosen device which in 8-bit devices can be 00 or FF and in 4-bit devices can be 0 or F.

To enter the Fill RAM function, from the SYSTEM MENU:

Press Key 4

This will display the Fill RAM Sub Menu.

Four features are displayed and entry into any particular one is made by pressing the appropriate key.

E.g., to fill RAM with 00s

Press Key 0 to display:

```
── FILL 00s ──────────────────────────── Zm3000 Rev X ──


    Lower Limit: 00  [0000]
    Upper Limit: 00   3FFF
```

The display will show the default state of the RAM address limits to be filled with 00.

This lower limit will be defaulted to zero and the upper limit will be defaulted to the size of the pre-selected device.

Interchange between the two limits is made by use of the vertical cursor keys (↑ and ↓) and the new selections can be entered by use of the hex-keyboard.

To confirm the Address Limits:

PRESS ENTER

If an Address Increment other than 1 is required it can now be entered directly from the keyboard and can take values from 1 to 8.

To confirm the Address Increment and fill the selected area of RAM with 00s:

PRESS ENTER

This will return the FILL RAM sub-menu. To return the SYSTEM MENU:

PRESS RESET

Fill RAM with FF or the Empty State

These two features can be selected from the Fill RAM Sub Menu by pressing F for Fill RAM with FFs or E for Fill RAM with the Empty State.

After selection, Address Limits and the Address Increment can be set up in the same way as for fill RAM with 00s.

5.3.2 **Fill RAM Arbitrarily—Numeric (Hex Data)**

This feature allows a string of up to 22 bytes of keyboard selected hex. data to be entered in such a way as to be repeated throughout the entire specified area of RAM.

To enter this feature from the SYSTEM MENU:

PRESS KEY 4 followed by KEY A

The display will show the default state of the RAM address limits to be filled with Hex Data.

```
—— FILL ARBITRARILY ——————————————— Zm3000 Rev X ——

     Lower Limit: 00  [0000]
     Upper Limit: 00   3FFF
```

The lower limit will be defaulted to zero and the upper limit will be defaulted to the size of the pre-selected device.

Interchange between the two limits is made by use of the vertical cursor keys (↑ and ↓) and the new selections can be entered by use of the hex-keyboard.

To confirm the Address Limits:

PRESS ENTER

After the Address Limits have been confirmed, the screen will display:

```
—— FILL ARBITRARILY ———————————— Zm3000 Rev X ——

     Lower Limit: 00   0000
     Upper Limit: 00   3FFF              [Address Increment:]  01
```

The screen cursor shows the Address Increment in the default state of 1. If an Address Increment other than 1 is required it can now be entered directly from the keyboard and can take values from 1 to 8.

To confirm the Address Increment:

PRESS ENTER

To select the Numeric input form (hex data):

PRESS ENTER

The screen will now display:

```
—— FILL ARBITRARILY ————————————— Zm3000 Rev X ——

   Lower Limit: 00   0000          Numeric: Select Input form
   Upper Limit: 00   3FFF            Address Increment: 01

   [String:]  00

   Run Function
```

A data string in bytes can be selected by use of the hex-keyboard. As each byte is selected, pressing ENTER will confirm the selection and allow the next byte to be chosen. Up to 22 bytes of hex data can be entered.

To delete bytes of data:

PRESS KEY ←

This will delete the last selected byte.

When the desired string of data has been selected, to fill the specified area of RAM:

PRESS KEY ↓ followed by ENTER

This will return the FILL RAM sub-menu. To return to the SYSTEM MENU:

PRESS RESET

Fill RAM Arbitrarily—ASCII

This feature allows a string of up to 22 characters from the ASCII register to be entered in such a way as to be repeated throughout the entire specified area of RAM.

To enter this feature from the SYSTEM MENU:

PRESS KEY 4 followed by KEY A.

The display will show the default state of the RAM address limits to be filled with ASCII string data.

```
—— FILL ARBITRARILY ———————————— Zm3000 Rev X ——

     Lower Limit: 00  [0000]
     Upper Limit: 00   3FFF
```

The lower limit will be defaulted to zero and the upper limit will be defaulted to the size of the pre-selected device.

Interchange between the two limits is made by use of the vertical cursor keys (↑ and ↓) and the new selections can be entered by use of the hex-keyboard.

To confirm the Address Limits:

PRESS ENTER

After the Address Limits have been confirmed, the screen will display:

```
— FILL ARBITRARILY ——————————————— Zm3000 Rev X —


    Lower Limit: 00   0000
    Upper Limit: 00   3FFF              [Address Increment:]  01
```

The screen cursor shows the Address Increment in the default state of 1. If an Address Increment other than 1 is required it can now be entered directly from the keyboard and can take values from 1 to 8.

To confirm the Address Increment:

PRESS ENTER

To select the ASCII input form:

PRESS KEY ← followed by ENTER.

The screen will now display:

```
— FILL ARBITRARILY ————————————————————— Zm3000 Rev X —

Lower Limit: 00 0000        ASCII          :  Select input form
Upper Limit: 00 3FFF

[String:]

□ ! " £ $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] _ '
a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~
——— Run Function —————————   ————————— Delete ——————————
```

The screen cursor will now be in the default position on the space character. By using all of the cursor keys, any character from within the ASCII register can be pin pointed and then selected by pressing the ENTER key.

When the required string is displayed (up to 22 characters) moving the cursor down onto the Run Function line and pressing ENTER will fill the specified RAM area with the ASCII string data, and return the FILL RAM sub-menu.

To return to the SYSTEM MENU:

PRESS RESET

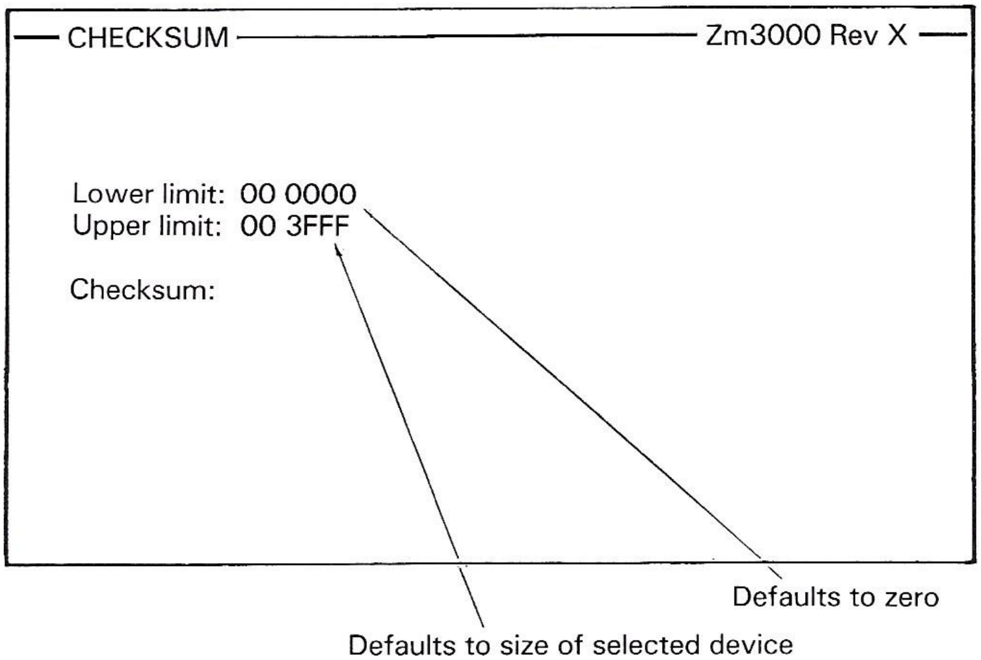
5.4 **In-Program:Checksum**

In-Program Checksum is an automatic addition of all byte size data programmed into the device.

Manual: Checksum

The Manual Checksum of the RAM can be applied at any time. To apply the checksum, from the SYSTEM MENU:

PRESS KEY 8 to display:

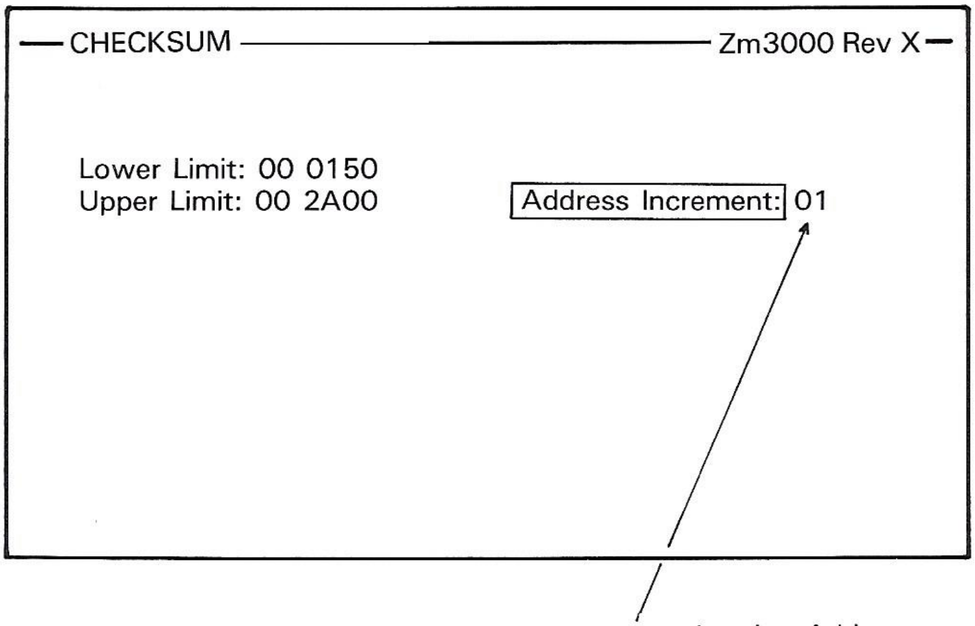


Address limits can be independently selected for the manual checksum (unlike the automatic versions which default to limits pre-selected in their respective operations i.e. Program, Load, Verify etc). Interchange between the two limits is made with the vertical cursor keys and new selections can be entered from the hex-keyboard.

To confirm the RAM address limits:

PRESS ENTER

After having confirmed the address limits the screen will display:



Screen cursor showing Address Increment defaulted to 1.

If an Address Increment other than 1 is required it can now be entered from the keyboard. (See Section 3 for further details.)

To confirm the Address Increment and implement the Checksum:

PRESS ENTER

The checksum will be displayed in the bottom left corner of the screen.

To return the SYSTEM MENU:

PRESS RESET

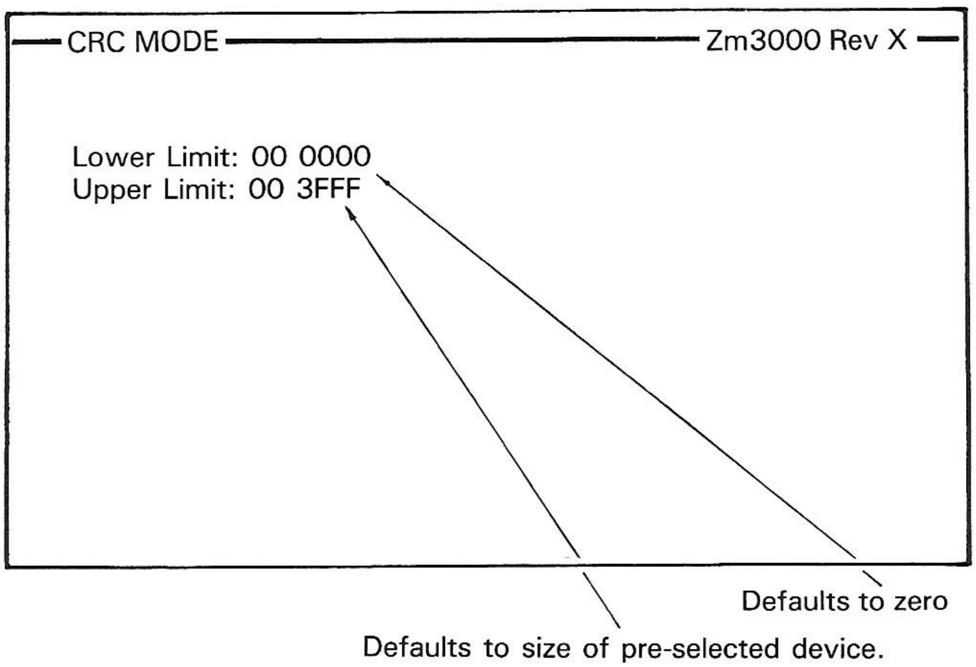
5.5 **Cyclic Redundancy Check (CRC)**

The Cyclic Redundancy Check applies a continuous process of shifting and addition to the RAM data. This yields a coded representation of the data, which is sensitive to the ordering of the data bytes unlike checksum which only considers their values.

To apply a CRC from the System Menu:

PRESS KEY 9

This will display:



Both Upper and Lower Address Limits can be changed, and also an Address Increment can be selected as with CHECKSUM.

CRC: Selecting Address Limits and Address Increment.

As with CHECKSUM, interchange between the two limits is made by use of the vertical cursor keys (↑ and ↓), and new selections can be entered directly from the hex-keyboard.

To confirm the Address Limits:

PRESS ENTER to display:

```
—— CRC MODE ——————————————————————— Zm3000 Rev X ——


    Lower Limit: 00 0000
    Upper Limit: 00 3900            Address Increment: 01
```

If an Address Increment other than 1 is required it can now be entered from the keyboard. (See section 3 for further details.)

To confirm the Address Increment and implement the CRC:

PRESS ENTER

When the check is complete, a four digit hex number indicating the result will appear on the screen.

For example:

Cyclic Redundancy Check: A768

To return to the SYSTEM MENU:

PRESS RESET

SECTION 6

stag

Sophisticated systems for the discerning engineer.

6 Interface

6.1 Setting the I/O Interface Parameters (for both RS232C ports and the IEEE)

On power-up the I/O parameters default to the last selection.

There are five RS232C categories of pre-set I/O interface parameter available on the Zm3000 and one parameter that applies to the IEEE only.

They are divided under two separate headings: ''Set Interface Hardware'' and ''Set I/O Software'' which are displayed on two separate screen pages:

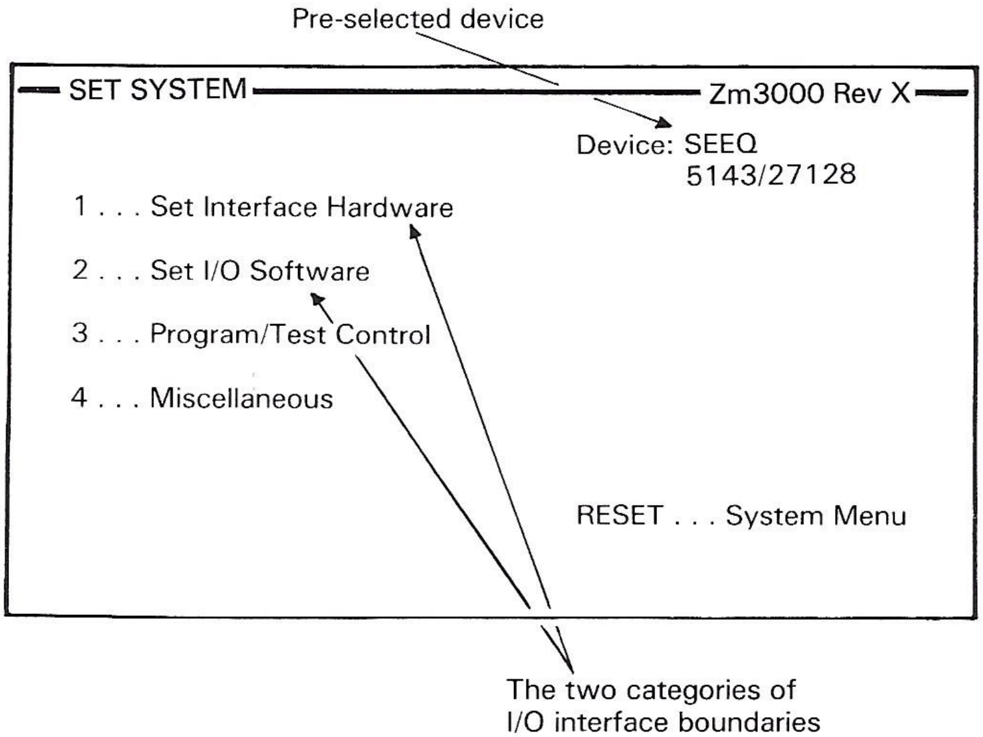
Interface parameters on Zm3000

Set Interface Hardware (Page 1)	Set I/O Software (Page 2)
Baud Rate Word Length Parity Stop Bits	Format Selection (plus control Z option)
IEEE Address Parameter	

Entry into both of these categories is first made by doing the following. From the SYSTEM MENU:

PRESS KEY A

This will display the SET SYSTEM sub-menu.



From the SET SYSTEM sub-menu

To enter "Set Interface Hardware":

PRESS KEY 1

To enter "Set I/O Software":

PRESS KEY 2

Set Interface Hardware

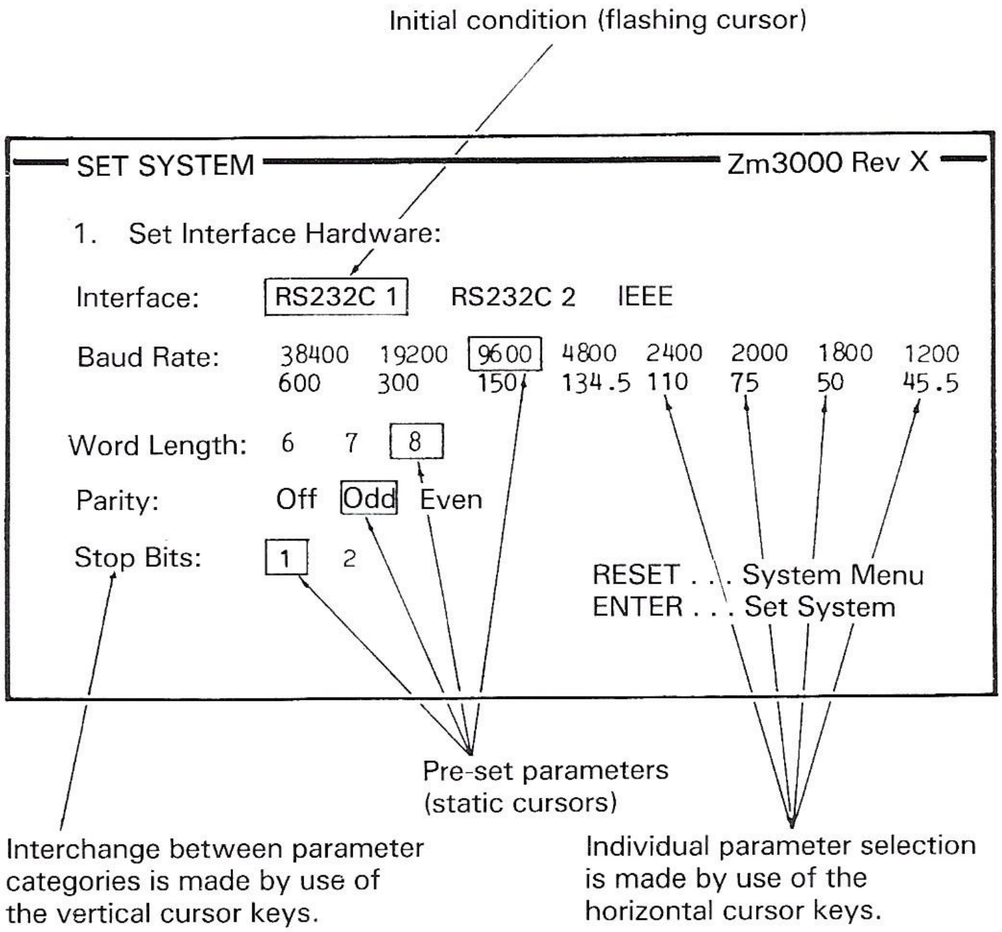
Hardware parameters for each of the three interface ports can be specified independently.

Selecting Hardware parameters for the RS232C 1 (female connector)

From the SYSTEM MENU:

PRESS KEY A followed by KEY 1

This will display the following:



The flashing screen cursor will be initially on the heading RS232C 1, and four static cursors will indicate the previously entered I/O parameters for this particular port.

Entry into a particular parameter category is made by use of the vertical cursor keys and is indicated by the applicable screen cursor flashing.

Individual parameter selection within each category is made by use of the horizontal cursor keys which direct the screen cursor from the latest entered selection to the required alternative.

For example: To change the Baud Rate from 9600 as displayed in the diagram to 2400.

PRESS KEY ↓ followed by:

PRESS KEY → twice

The screen cursor will now be flashing on the new Baud Rate selection of 2400.

After all new parameters have been specified,

PRESS ENTER

This will return the SET SYSTEM sub-menu. To return to the SYSTEM MENU:

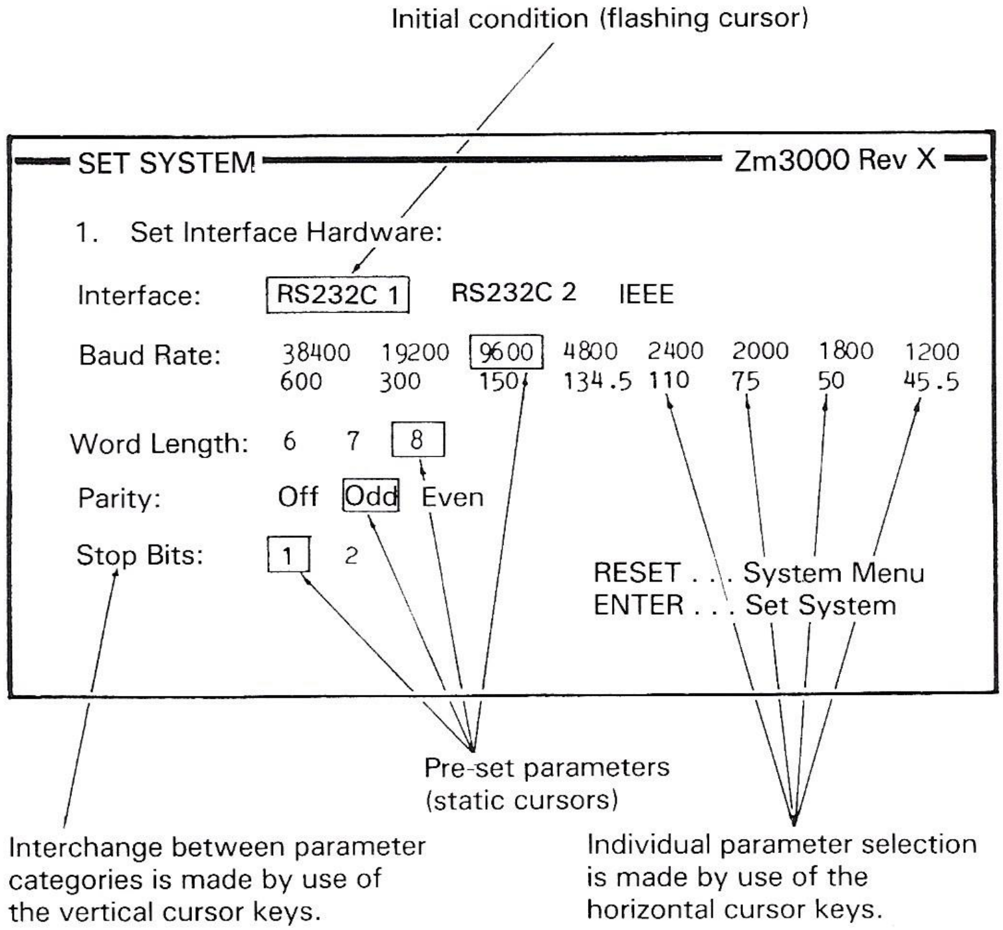
PRESS RESET

Selecting Hardware parameters for the RS232C 2 (male connector)

From the SYSTEM MENU:

PRESS KEY A followed by KEY 1

This will display the following:

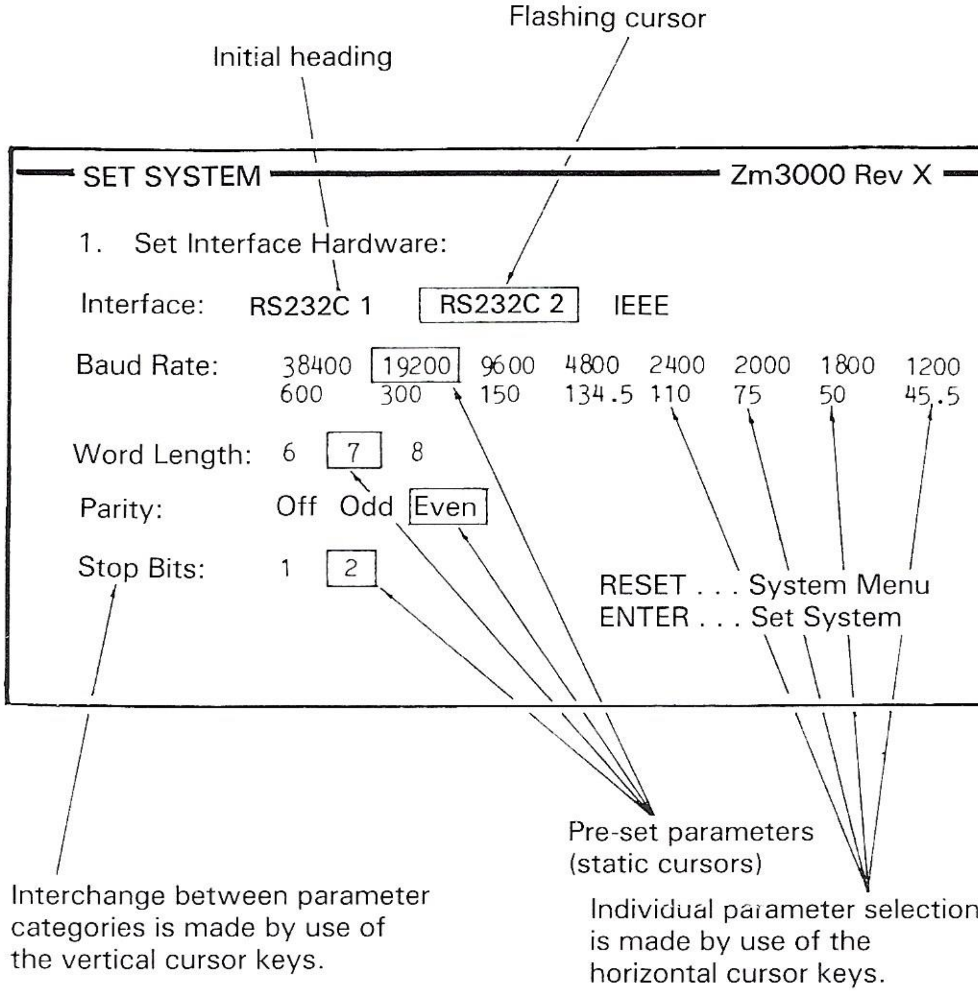


The flashing screen cursor will be in its default state on the RS232C 1 heading. To select the RS232C 2 category.

PRESS KEY →

With the selection of the RS232C 2 port, all the parameters indicated will be those previously selected for this configuration.

After RS232C 2 has been selected, the screen will display:



Entry into a particular parameter category is made by use of the vertical cursor keys and is indicated by the applicable screen cursor flashing.

Individual parameter selection within each category is made by use of the horizontal cursor keys which direct the screen cursor from the last entered selection to the required alternative.

To confirm the selections:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

Selecting the address parameter for the IEEE-488 Talker/Listener Interface Port

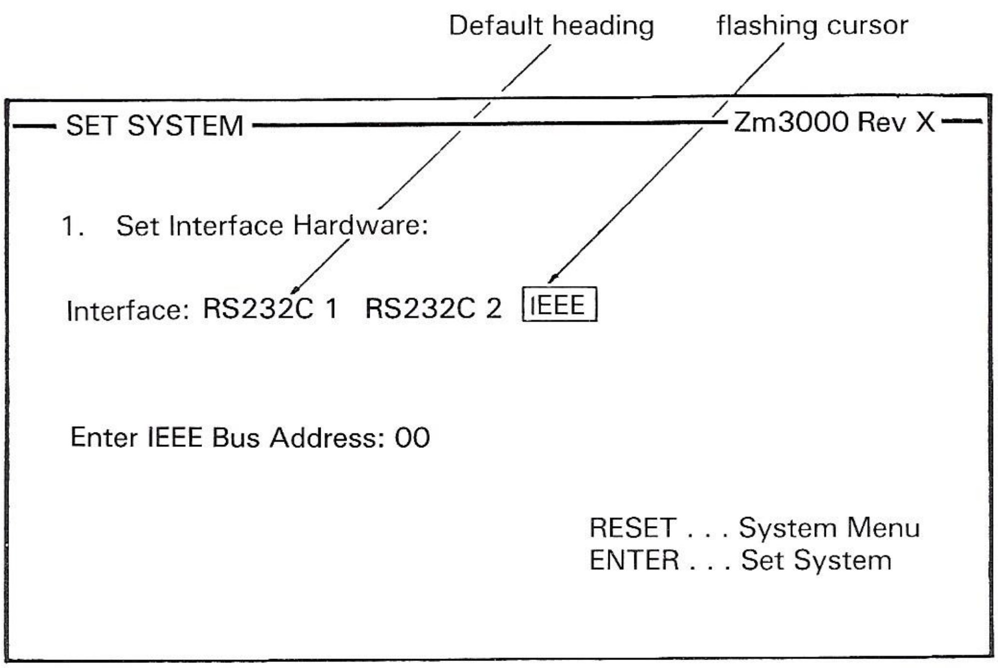
A two digit coded address can be entered that will allow a remote-controller connected via the IEEE interface port to identify the System 3000.

To select the address parameter from the SYSTEM MENU:

PRESS KEY A followed by KEY 1

This will display the screen cursor defaulted to the RS232C 1 heading. To select the IEEE:

PRESS KEY → twice to display:



Pressing the down cursor key (↓) will make the address available for selection, indicated by the pre-set parameter flashing. The new address can now be entered from the keyboard. To confirm this selection:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

SET I/O SOFTWARE (Format Parameters)

Note: Format parameter selection for the System 3000 does not apply to the IEEE interface. The IEEE may only be used under remote control, therefore the format selection must be made through the remote controller.

Selecting the format parameter.

From the SYSTEM MENU:

PRESS KEY A followed by KEY 2 to display:

Flashing screen cursor defaulted to the last used format parameter

```
—— SET SYSTEM ——————————————————————— Zm3000 Rev X ——

   2.  Set I/O Software:

 Format:  INTELLEC 86  [MOTOROLA S-Record]  Extended TEK-HEX

 Standard TEK-HEX   HEX-ASCII   STAG-HEX   DEC-Binary   Binary

  MOS Technology     Fairchild FAIRBUG     HEX-ASCII (SPACE)

  Cntrl Z:  [Off]  On
                                  RESET . . . System Menu
  Number of NULLS: 004            ENTER . . . Set System
```

Control Z option

Nine format parameters are available. Selection can be made by use of the horizontal cursor keys which direct the screen cursor from the last entered selection to the required alternative.

The Control Z function is a development aid used mostly on Intel development systems. It is required to terminate the transmission of INPUT to an outside system.

The Control Z option can be selected by use of the down cursor and specified as either On or Off by use of the horizontal cursor keys.

To confirm the selection:

PRESS ENTER

The 'Number of Nulls' option can be selected using the vertical cursor keys and the value can be entered from the hex. keyboard. Only values from 0 to 72 can be used. If a larger value is entered, the display will reset to 72 when Enter is pressed.

Press Enter to confirm the selection.

To return to the SYSTEM MENU:

PRESS RESET

6.2 I/O Mode

Only two of the available interface ports can be used for INPUT and OUTPUT when the System 3000 is in the LOCAL MODE of operation: these are RS232C 1 and RS232C 2.

The IEEE-488 interface is a talker/listener port and may only be used under REMOTE CONTROL.

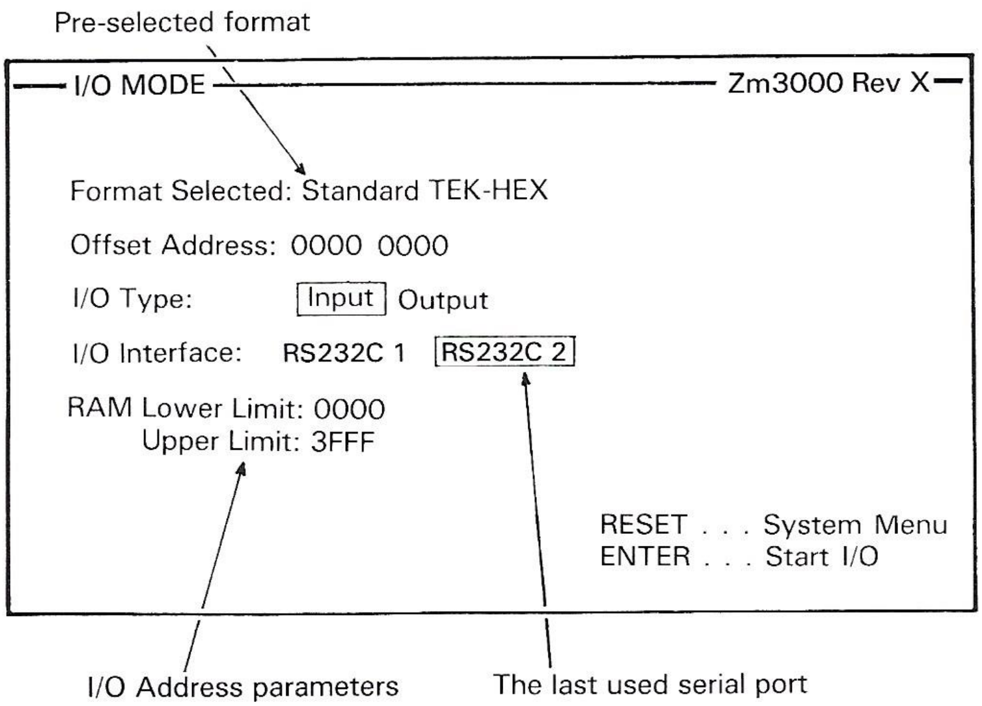
The RS232C ports can be individually selected to operate in either INPUT or OUTPUT modes.

Selecting the I/O mode.

To enter the I/O mode from the SYSTEM MENU:

PRESS KEY I/O

This will display the associated parameters in their default states. For example:



The display shows:

A. a flashing screen cursor indicating INPUT the I/O default state.

B. a static screen cursor indicating the last serial port used in conjunction with either INPUT or OUTPUT (in the example RS232C 2).

C. three pre-selectable address parameters for both INPUT and OUTPUT. These are:

INPUT OFFSET ADDRESS	1	OUTPUT OFFSET ADDRESS
INPUT RAM START ADDRESS	2	OUTPUT RAM START ADDRESS
INPUT RAM STOP ADDRESS	3	OUTPUT RAM STOP ADDRESS

The INPUT OFFSET ADDRESS is defaulted to zero.

The INPUT RAM START ADDRESS is defaulted to zero.

The INPUT RAM STOP ADDRESS is defaulted to the size of the RAM.

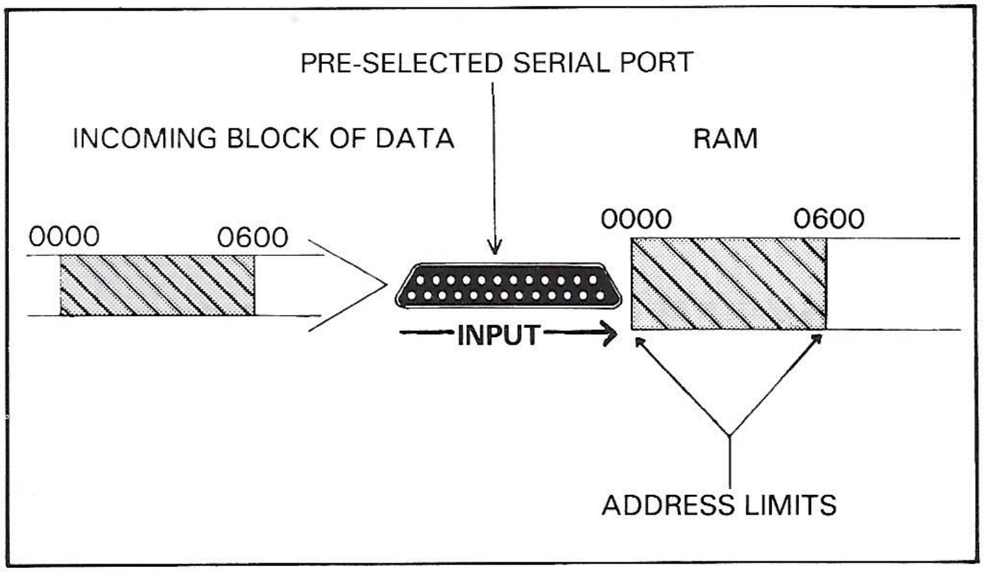
The OUTPUT OFFSET ADDRESS is defaulted to zero.

The OUTPUT RAM START ADDRESS is defaulted to zero.

The OUTPUT RAM STOP ADDRESS is defaulted to the size of the pre-selected device.

6.2.1 **Input**

Operation of Input



The I/O mode can be entered by pressing the I/O key.

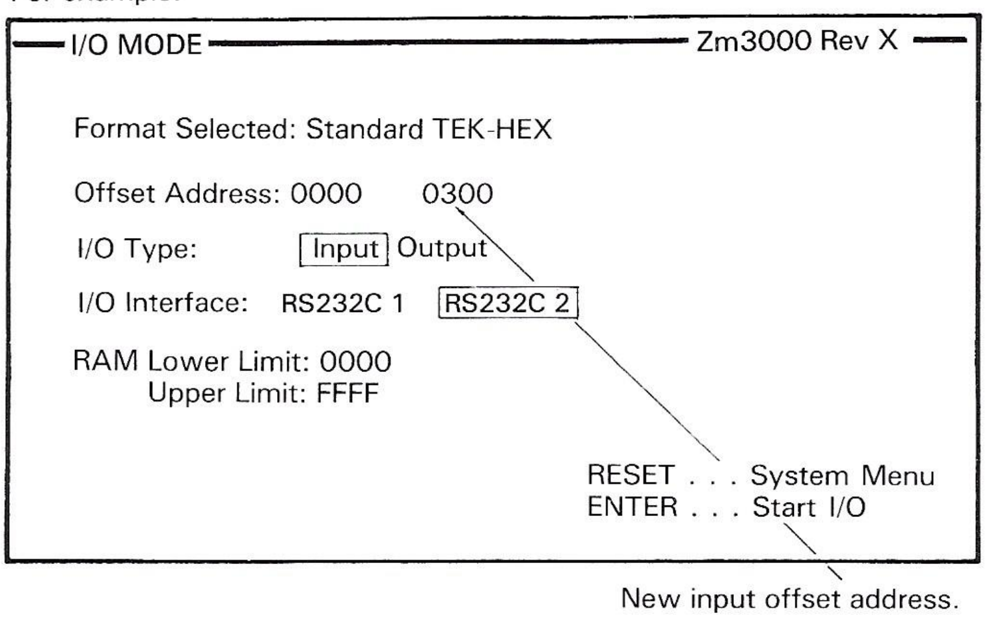
Entering the Input Parameters

Input Offset Address

The offset address will be shown flashing at the top of the screen in its default state 0000 0000.

A new selection can be made by use of the hex-keyboard whereby up to eight figures can be directly entered onto the display.

For example:

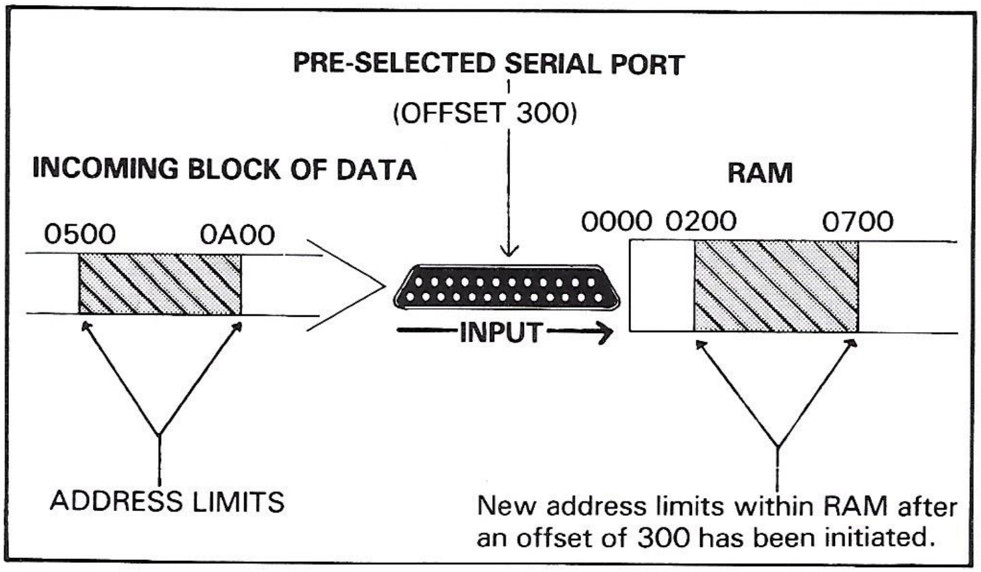


Eight figures are available in order to cater for the extended formats which contain a larger address field than the standard version. Therefore, to position data within the Programmer's RAM an eight figure offset may need to be specified.

Structure of Input Offset

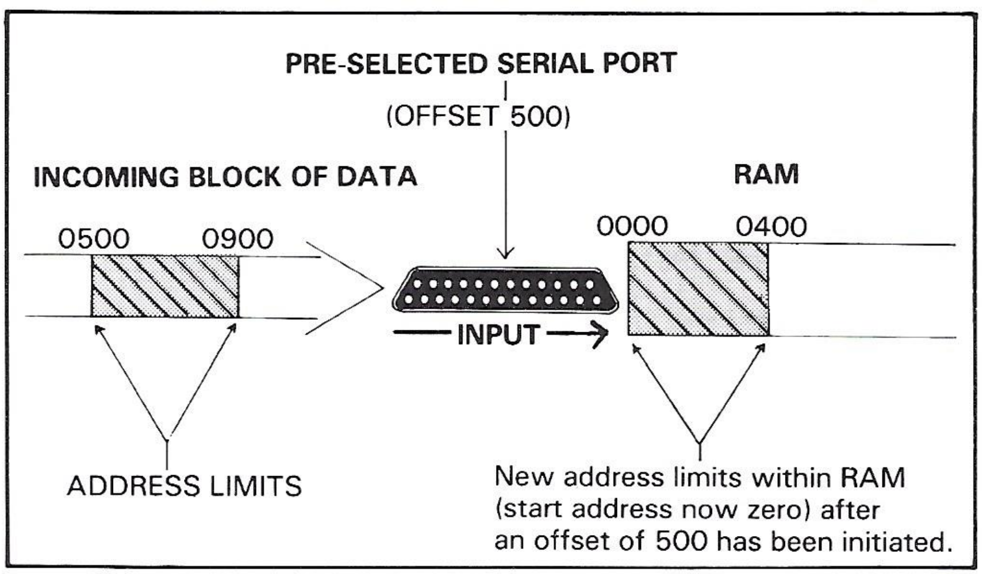
An incoming block of data originating from peripheral equipment can be re-located at a lower address within the RAM.

For example: OFFSET 0300



A more widely used example occurs when an offset is specified so that the start address of the incoming block of data becomes zero in the RAM.

For example:



Selecting Input and the Required Serial Port

If the offset address has been selected or is not required, to select the INPUT mode:

PRESS KEY ↓

The screen cursor will show the INPUT mode already entered. (INPUT is the default state of the I/O mode).

To select a specific serial port:

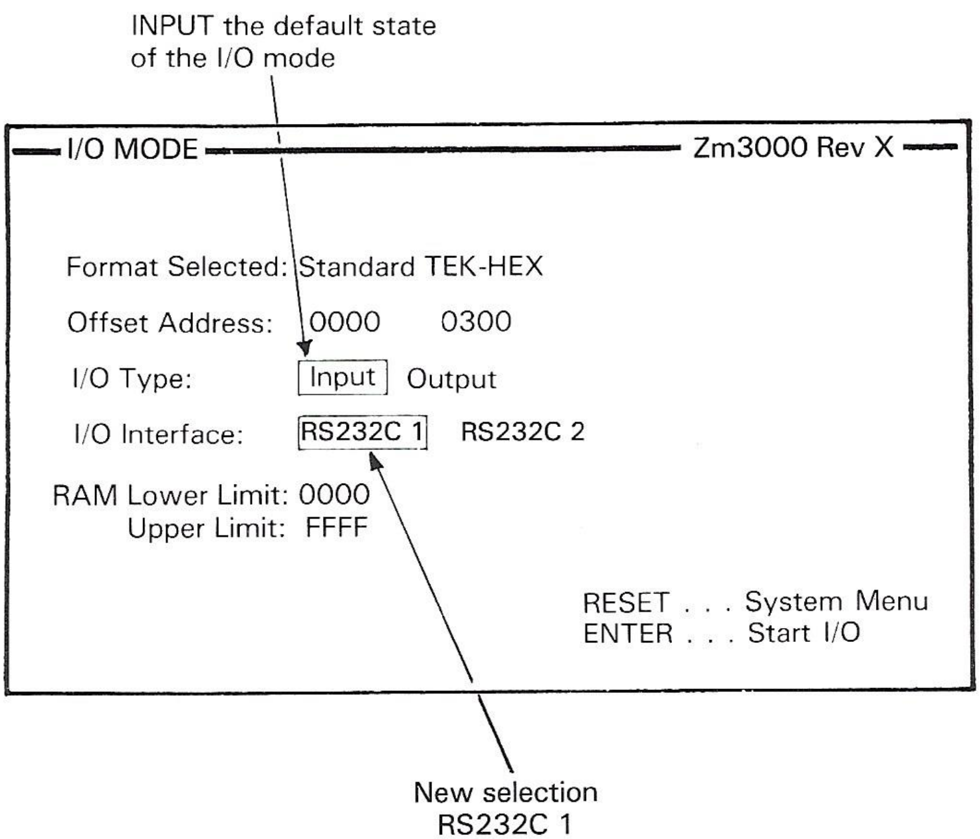
PRESS KEY ↓

This will show the screen cursor flashing on the last used selection (for instance RS232C 2).

To select RS232C 1:

PRESS KEY ←

The screen cursor will have now moved onto the RS232C 1 selection. Interchange between the serial interfaces is made by use of both horizontal cursor keys.



INPUT: RAM Start Address and RAM Stop Address

In addition to an offset a RAM start and stop address can be specified, whereby only a section of the in-coming data from peripheral equipment will be input into specified address limits within the RAM.

After the serial port has been specified, to select the RAM start and stop addresses:

PRESS KEY ↓

The lower limit will now be flashing. It will be in its default state of zero, and a new value can be entered from the hex-keyboard. To select the upper limit:

PRESS KEY ↓

This will show the upper limit flashing. It will be defaulted to the size of the RAM. A new value can be entered from the hex-keyboard.

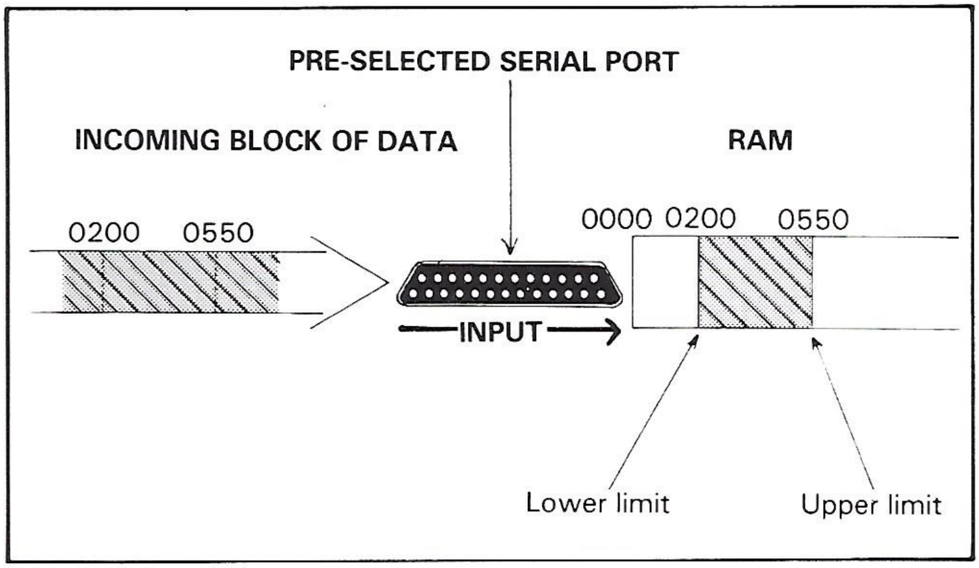
```
—— I/O MODE ——————————————————————————— Zm3000 Rev X ——

  Format Selected: Standard TEK-HEX

  Offset Address: 0000      0300

  I/O Type:         [Input]  Output

  I/O Interface:    [RS232C 1]  RS232C 2

  RAM Lower Limit: 0200
        Upper Limit:  0550

                                   RESET . . . System Menu
                                   ENTER . . . Start I/O
```

New RAM address limits

Structure of Input Start and Stop Addresses

A section of the incoming data from peripheral equipment will be put into specified address limits within the RAM.

For example: Lower limit 0200
Upper limit 0550

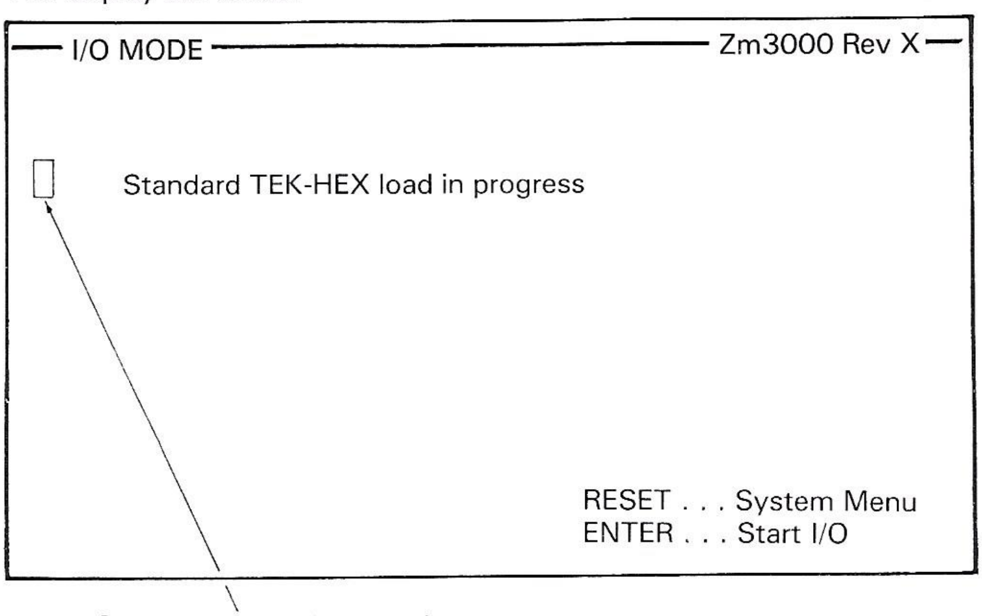


Implementing the Input

When the necessary parameters have been set, to implement INPUT:

PRESS ENTER

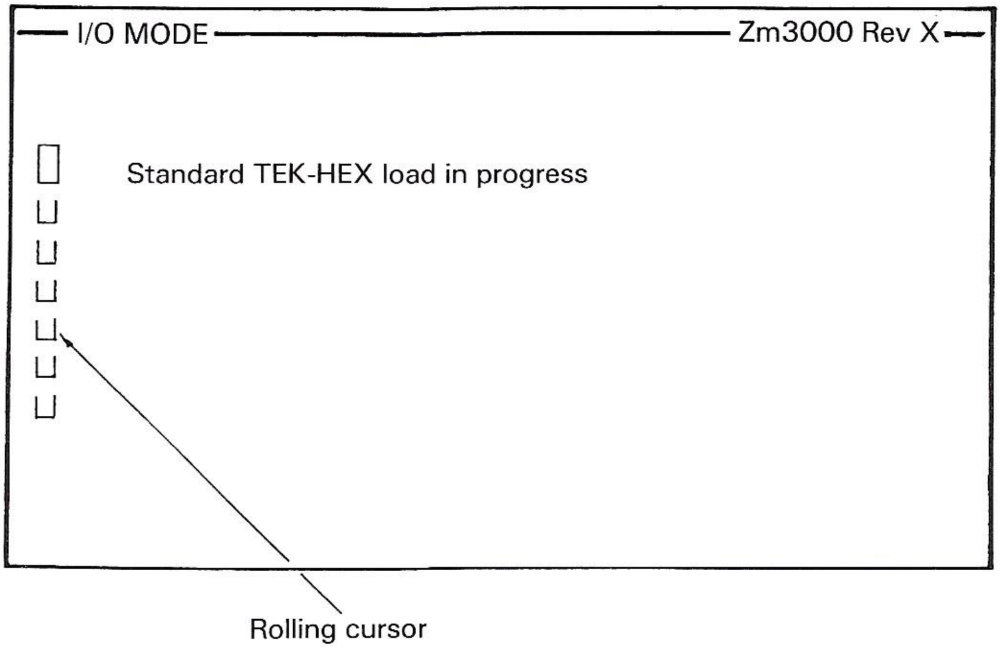
This will put the System 3000 into a state of readiness to receive incoming data.
The display will show:



Static cursor, indicating that
the System 3000 is ready to receive
incoming data.

During INPUT a cursor will roll in response to each ''carriage return'', ''line feed'' (CR,LF) to indicate that the LOAD is in progress.

For example:



NB: Transmitted data that falls outside the boundaries of any pre-selected RAM limits will still be indicated during the INPUT operation by the rolling cursor but will not affect the RAM.

When the INPUT is complete the display will show a checksum and a byte count of all data loaded into RAM.

For example:

```
— I/O MODE ———————————————————————— Zm3000 Rev X —

   Standard TEK-HEX load completed




   Bytes loaded 1000
   Checksum   : A633

                                RESET . . . System Menu
                                ENTER . . . Start I/O
```

To return to the I/O mode:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

Error Reporting on Input

Two possible error messages can be displayed by the System 3000 at the end of a data INPUT instead of the checksum.

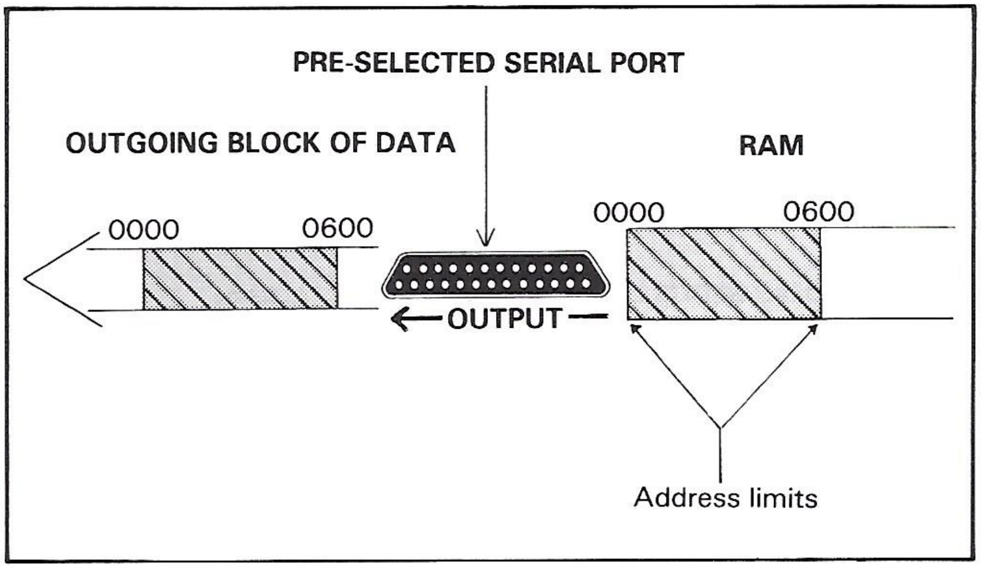
These are:

1. Non-hex character received
A non-hex character was received where a hex character was expected.

2. Checksum field error detected.
A checksum error in a record (a line) was detected.

6.2.2 **Output**

Operation of Output



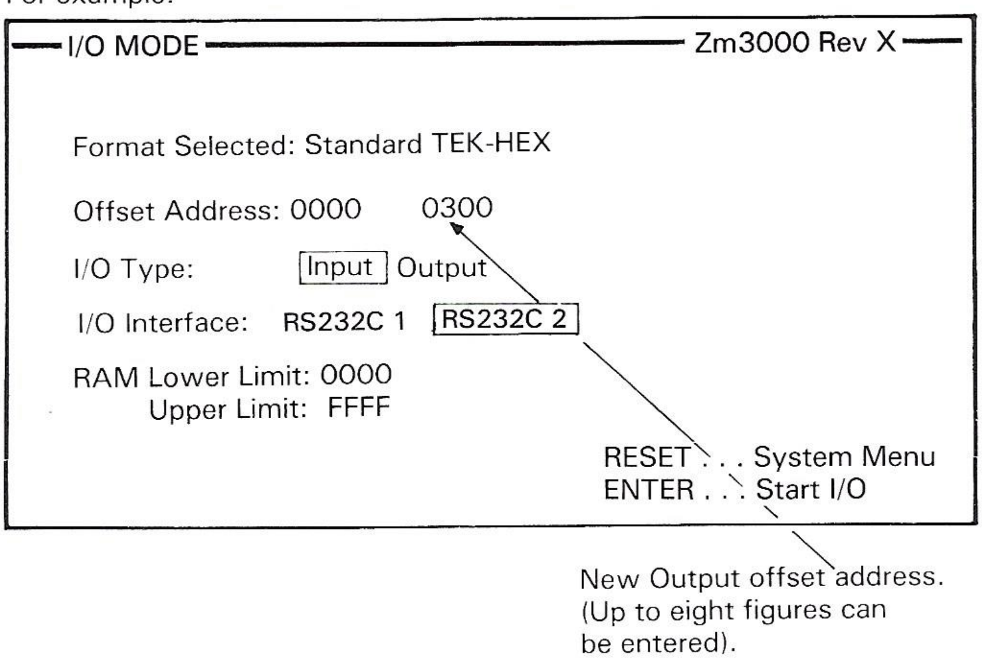
Press I/O

Output Offset Address

The offset address will be shown flashing at the top of the screen in its default state 0000 0000.

A new selection can be made by use of the hex-keyboard whereby up to eight figures can be directly entered onto the display.

For example:

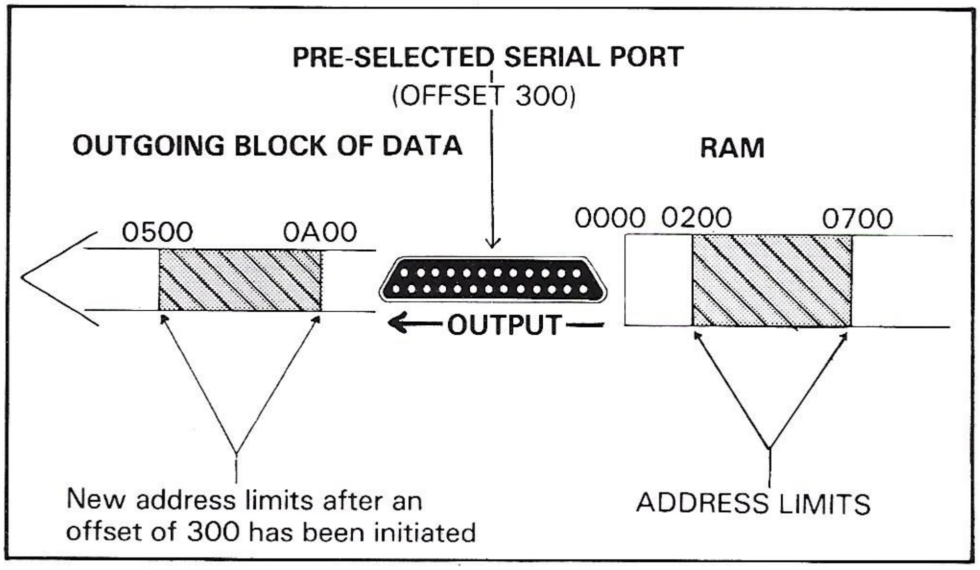


Eight figures are available in order to cater for the extended formats which contain a larger address field than the standard versions. Therefore, to position data within the RAM an eight figure offset may need to be specified.

Structure of Output Offset

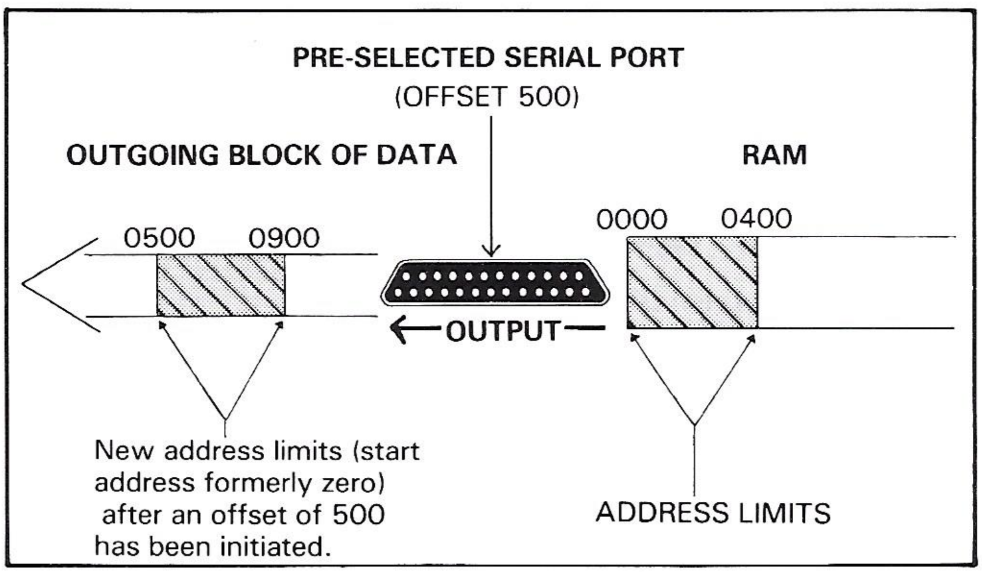
A block of data originating from the RAM can be re-located at a higher address.

For example: OFFSET 0300



A more widely used example is when re-location to a higher address occurs from a start address of 0000 in the RAM.

For example:



Selecting Output and the Required Serial Port

If the offset address has been selected or is not required, to select the OUTPUT mode:

PRESS KEY ↓

The screen cursor will show the INPUT mode which is the default state. To select OUTPUT:

PRESS KEY →

The screen cursor will now indicate that the System 3000 is configured for this mode of operation.

To select a specific serial port:

PRESS KEY ↓

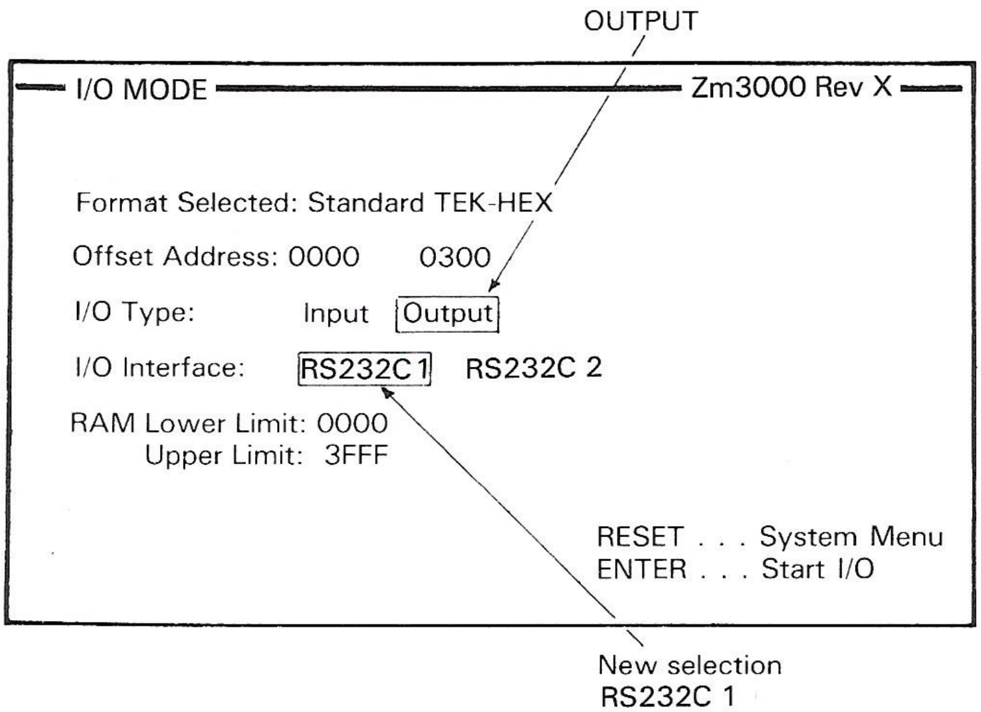
This will show the screen cursor flashing on the last used selection (for instance RS232C 2).

To select RS232C 1:

PRESS KEY ←

The screen cursor will have now moved onto the RS232C 1 selection. Interchange between the serial interfaces is made by use of both horizontal cursor keys.

After having selected OUTPUT, the screen display will show the following:



In some circumstances, the static screen cursor indicating the last used serial port will move onto the alternative serial option simultaneously with selection of either INPUT or OUTPUT.

This feature simplifies the operation of the System 3000 when it is linked (in LOCAL MODE) to two items of peripheral equipment via both serial interfaces.

For example: a "computer" (with the System 3000 configured for INPUT through the RS232C1 port) and a "tape reader" (with the System 3000 configured for OUTPUT through the RS232C2 port).

The configuration can be entered or reset by use of the cursor keys.

Output: RAM Start Address and RAM Stop Address

In addition to an offset, a RAM start and stop address can be specified on OUTPUT.

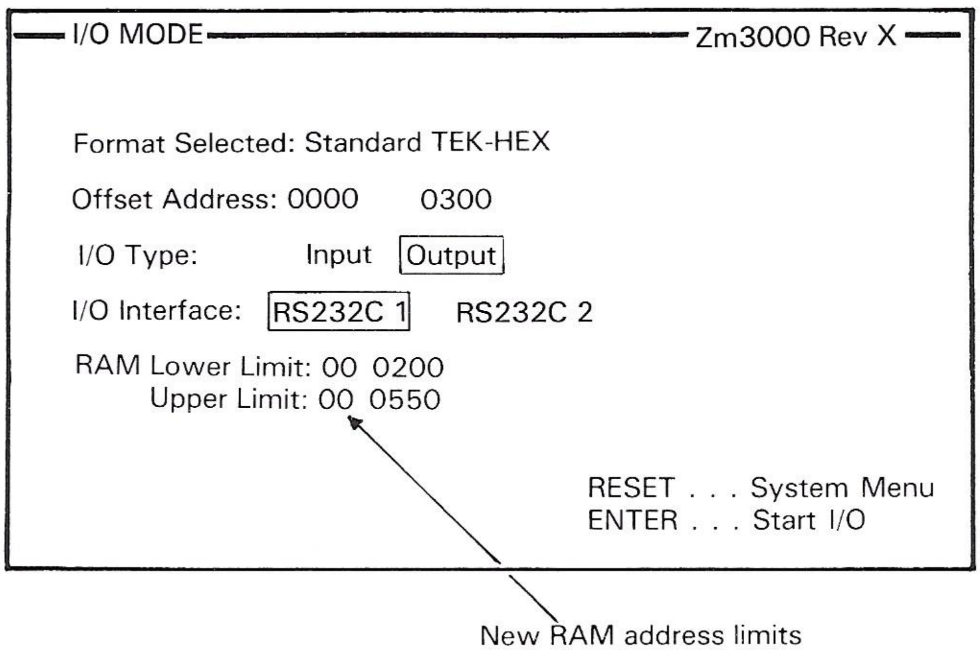
After the serial port has been specified, to select the RAM start and stop addresses:

PRESS KEY ↓

The lower limit will now be flashing. It will be in its default state of zero, and a new value can be entered from the hex-keyboard. To select the upper limit:

PRESS KEY ↓

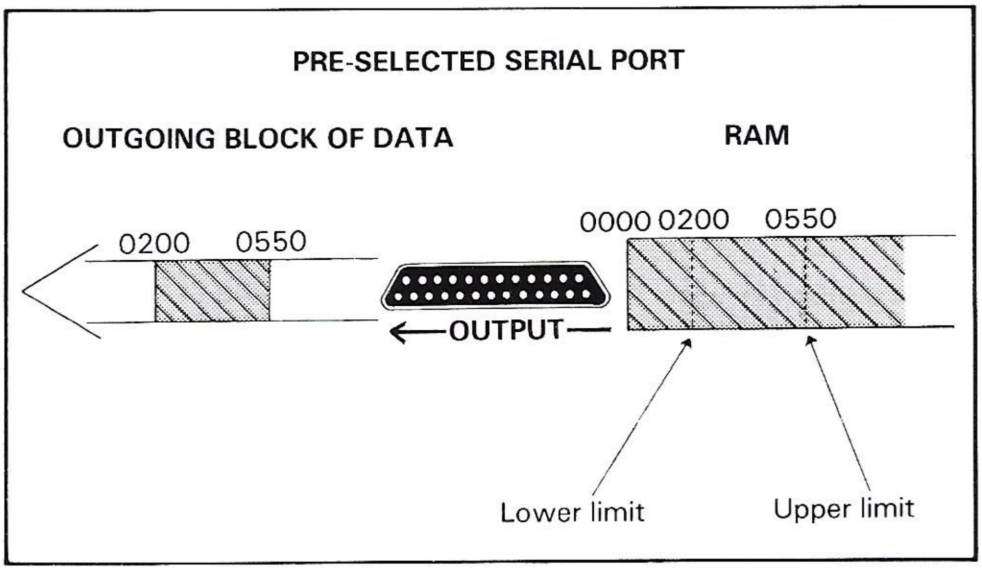
This will show the upper limit flashing. It will be defaulted to the size of the pre-selected device. A new value can be entered from the hex-keyboard.



Structure of Output Start and Stop Addresses

The block of data to be output within the RAM can be specified by using start and stop addresses.

For example: Lower Limit 0200
Upper Limit 0550



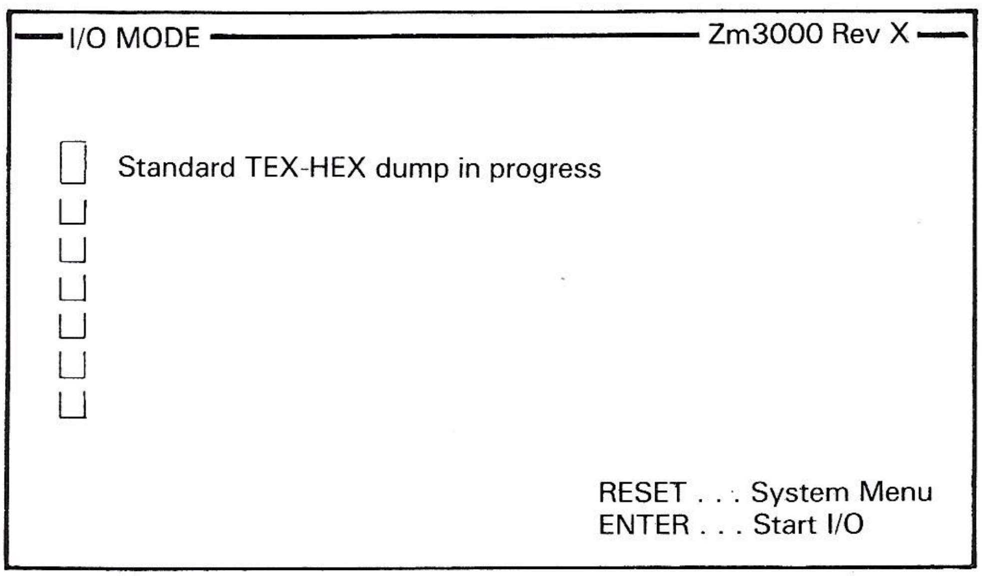
Implementing the OUTPUT

When the necessary parameters have been set, to implement the OUTPUT of data to peripheral equipment:

PRESS ENTER

During OUTPUT a screen cursor will roll in response to each ''carriage return, line feed'' (CR, LF), ie. after each data line is sent from the RAM.

For example:



SECTION 7

7

stag

Sophisticated systems for the discerning engineer.

7 Format Descriptions—Memory Devices Only

7.1 Interface Formats

The following formats are available on the System 3000 for memory devices.

FORMAT	SECTION
STANDARD TEK-HEX	7.2
EXTENDED TEK-HEX	7.3
INTELLEC 86	7.4
MOTOROLA S-RECORD	7.5
HEX-ASCII	7.6
STAG-HEX*	7.7
BINARY AND DEC-BINARY	7.8
MOS-TECHNOLOGY	7.9
FAIRCHILD FAIRBUG	7.10
HEX ASCII (SPACE)	7.11

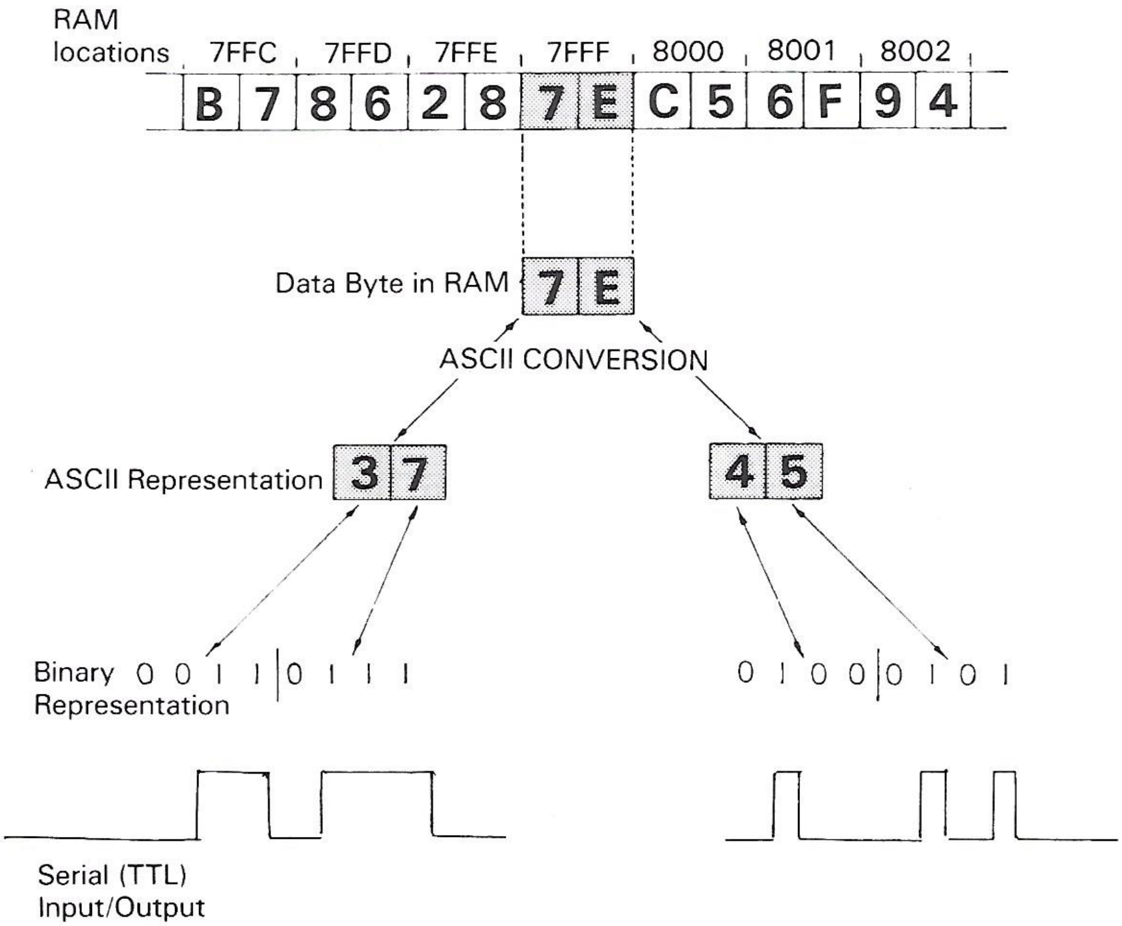
7 Format Descriptions—Memory Devices Only

7.1 Interface Formats

The following formats are available on the System 3000 for memory devices.

FORMAT	SECTION
STANDARD TEK-HEX	7.2
EXTENDED TEK-HEX	7.3
INTELLEC 86	7.4
MOTOROLA S-RECORD	7.5
HEX-ASCII	7.6
STAG-HEX*	7.7
BINARY AND DEC-BINARY	7.8
MOS-TECHNOLOGY	7.9
FAIRCHILD FAIRBUG	7.10
HEX ASCII (SPACE)	7.11

Structure and Conversion of Data between the Serial Signal via the RS232C Interface Ports of the System 3000 Ram.



7.2 **Standard Tek-Hex**

The TEK HEX format consists of:

(a) a start code, i.e. /

(b) the address of the first byte of data in an individual record, e.g.: 0000

(c) the sum of the number of data bytes in an individual record, e.g.: 1A

(d) checksum 1 which is a nibble addition of the address (4 characters) and the byte count (2 characters), e.g.: 0B

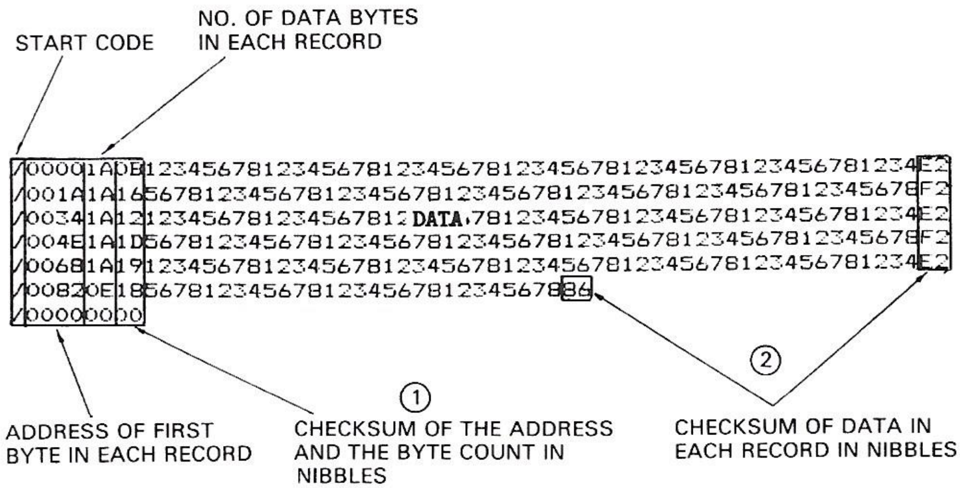
(e) data in bytes, e.g.: 12 34 56 78

(f) checksum 2 which is a nibble addition of all data.

(g) an end record which automatically stops the operation when 00 is specified in the byte count (c).

For example:

START ADDRESS:	0000
STOP ADDRESS:	008F
OFFSET:	0000



Calculation of TEK HEX Checksums

Unlike the other formats, the TEK HEX has two checksums which are both the result of nibble additions, as opposed to byte additions.

Checksum 1 is a nibble addition of the 'address' and the 'byte count' which make 6 characters in total.

Checksum 2 is a nibble addition of data alone.

```
/00001A0B12345678123456781234567812345678123456781234E2
CHECKSUM 1 CHECKSUM 2 i6781234567812345678123456781234567812345678F2
/00034030A12345615
/00000000
```

EXAMPLE: THE THIRD "DATA RECORD" OF THE ABOVE FORMAT

CHECKSUM 1

(i)	this is:	/10034030A
(ii)	the start code and the checksum are removed:	/0A
(iii)	six nibbles remain:	003403
(iv)	they are added together:	0+0+3+4+0+3 = A
(v)	A is the checksum which is displayed in byte form as above:	/1003403 (0A)

CHECKSUM 2

(i)	this is	123456 15
(ii)	the checksum is removed:	15
(iii)	six nibbles remain:	123456
(iv)	these are added together:	1+2+3+4+5+6 = 15
(v)	15 is the checksum as above:	123456 (15)

When addition of nibble information occurs in longer records the checksum may consist of more than one byte. When this occurs the least significant byte is always selected to undergo the above calculation.

7.3 **Extended Tek-Hex**

The EXTENDED TEK HEX consists of:

(a) a start code: % (percentage)

(b) a count of the nibbles in an individual record, e.g.: 3B

(c) the record types, i.e.: 6—Data Record;
8—End Record

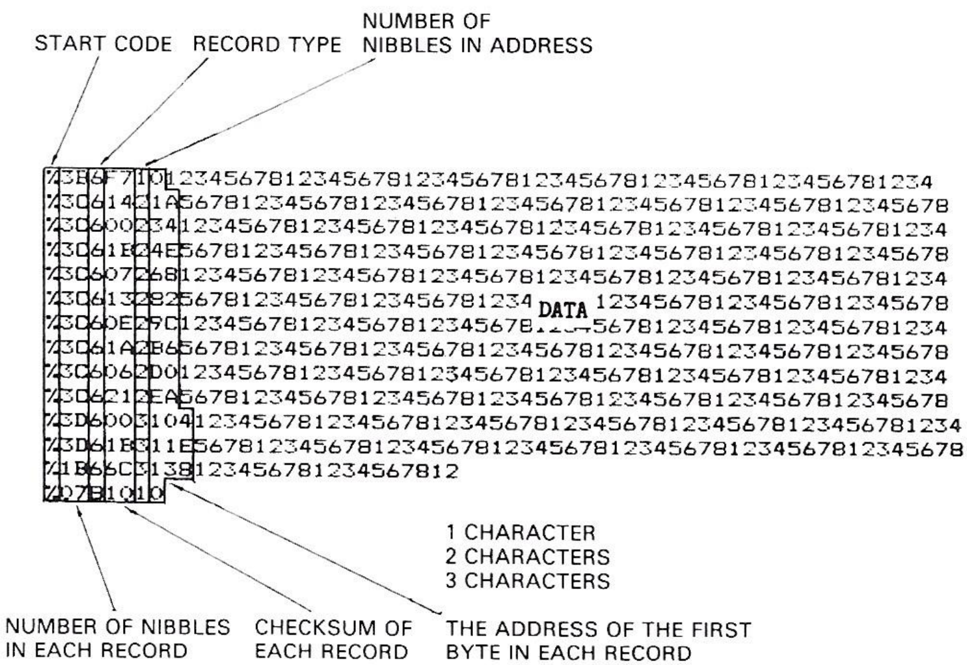
(d) a checksum of the whole of an individual record excluding the %
e.g.: F7

(e) *the number of nibbles comprising—"the address of the first byte in each record", e.g.: 1, 2, 3 etc.

(f) the address of the first byte of data in an individual record, e.g.: 0, 1A, 104

For example:

START ADDRESS:	0000
STOP ADDRESS:	0140
OFFSET:	0000 0000



*Sections (e) and (f) are integrated:
As the operation progresses the address field lengthens. More characters are added to show this expansion. The nibble count of section (e) reflects this, e.g.:

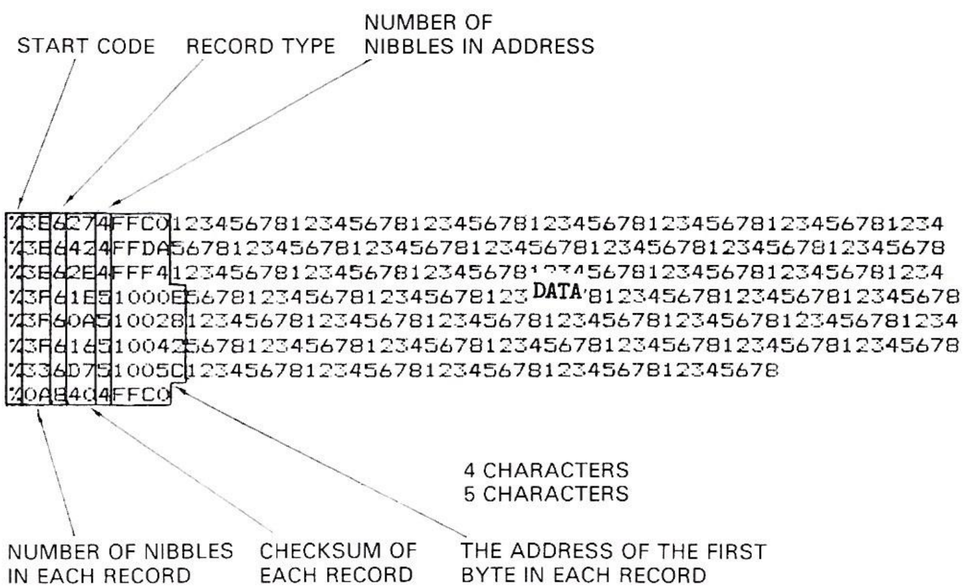
2/1A	6/100000	A/1B4625DC95
2 Characters	6 Characters	A Characters (10 in Decimal)

The nibble count takes values from '0' to rise to 'F' making a fifteen character address field possible.

EXTENDED TEK HEX with an Offset, Displaying Transition from 4 Character Address Field to 5 Character Address Field.

For example:

START ADDRESS:	0000
STOP ADDRESS:	00AF
OFFSET:	0000 FFC0



Calculation of the Extended Tek Hex Checksum

Unlike the standard version the EXTENDED TEK HEX has only one checksum.

```
%3B6F7101234567812345678123456781234567812345678123456781234
%3C61421A5678123456781234567812345678123456781234567812345678
%0A61C23412
%0781010
```

EXAMPLE: THE THIRD LINE OF THE ABOVE FORMAT.

(i) this is:	% 0A61C23412
(ii) the start code and the checksum are removed:	% IC
(iii) eight nibbles remain:	0A623412
(iv) these are added together:	0+A+6+2+3+4+1+2=1C
(v) 1C is the checksum as above:	% 0A6 (1C) 23412

When addition of nibble information occurs in longer records, the checksum may consist of more than one byte. When this occurs the least significant byte is always selected to undergo the above calculation.

7.4 Intellec 86

The extended INTELLEC 86 format when displayed consists of:

(a) a start code, i.e.: (colon)

(b) the sum of the number of bytes in a particular record, e.g.: 1A

(c) the address of the first byte of data in an individual record, e.g.: 0000

(d) the record types, i.e.: 00—Data Record
01—End Record
02—Segment Base Address record (SBA)*

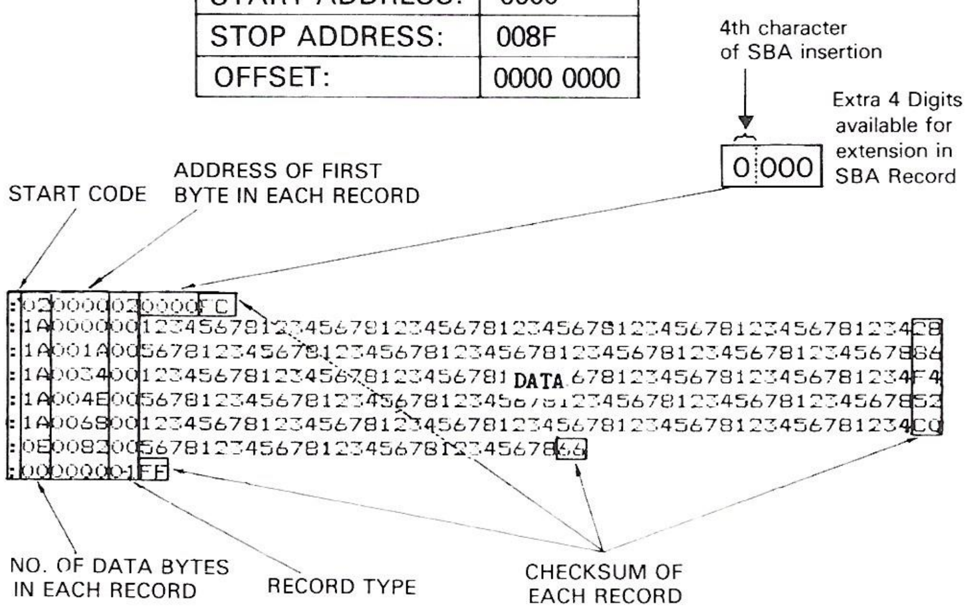
*The SBA is the record that displays the INTELLEC 86 extension. This is achieved by the provision of an extra digit which corresponds to the 4th character of the SBA insertion. This 4th character is effectively the extension which lengthens the standard (FFFF) limitation, into the EXTENDED INTELLEC (FFFFF).

(e) data (in bytes), e.g.: 12 34 56 78

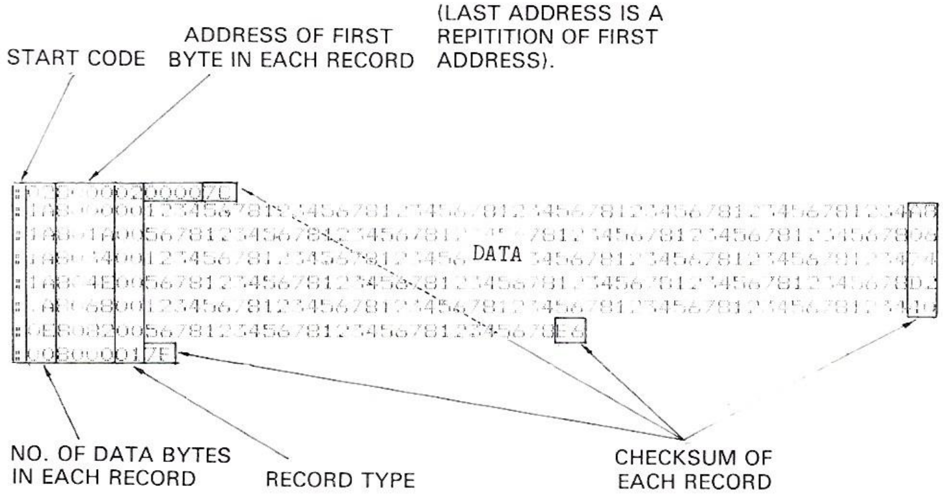
(f) a checksum of an individual record, e.g.: 28

For example:

START ADDRESS:	0000
STOP ADDRESS:	008F
OFFSET:	0000 0000



And with an Offset of: (0000 8000)



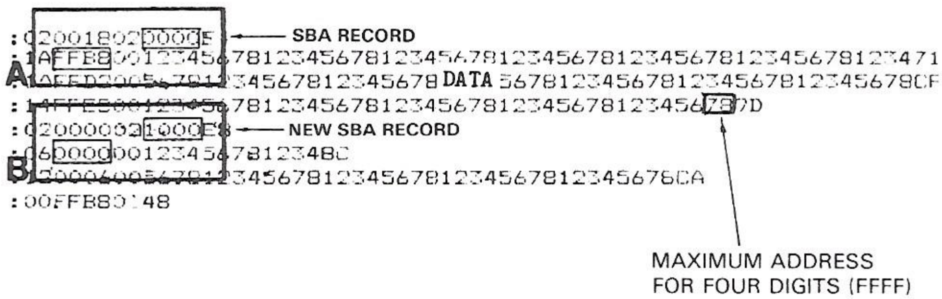
SBA Repetition

In some operations where an offset is in use the SBA can be displayed twice.

When the address field passes the maximum quantity for a four digit figure, i.e.: (FFFF) a second SBA record is specified.

For example:

START ADDRESS	FFAO
STOP ADDRESS	FFFF
OFFSET	0000 0018



The SBA is added to the address field in the following fashion:

B

EXTENSION
DIGIT

```
  1 0 0 0   SBA INSERTION
+   0 0 0 0   ADDRESS FIELD

= 1 0 0 0 0
```

A

EXTENSION
DIGIT

```
  0 0 0 0   SBA INSERTION
+   F F B 8   ADDRESS FIELD

= 0 F F B 8
```

If required by the user the remaining 3 digits of the SBA insertion can be non zero.

Calculation of the Intellec 86 Checksum

```
:1A00000012345678123456781234567812345678123456781234 28
:0100 1A00568F
:00000001FF
```

EXAMPLE: THE SECOND "DATA RECORD" OF THE ABOVE FORMAT.

(i)	this is:	: 01 00 1A 00 56 8F
(ii)	the start code and the checksum are removed:	: 8F
(iii)	five Bytes remain:	01 00 1A 00 56
(iv)	these are added together:	01 + 00 + 1A + 00 + 56 = 71
(v)	the total '71' is converted into Binary:	7 → 0111, 1 → 0001
(vi)	the Binary figure is reversed. This is known as a complement:	8 → 1000, E → 1110
(vii)	a one is added to this complement. This addition forms a two's complement	8 → 1000, F → 1111
(viii)	8F is the checksum as above:	: 01 00 1A 00 56 (8F)

When addition of information occurs in longer records the checksum may consist of more than one byte. When this occurs the least significant byte is always selected to undergo the above calculation.

7.5 **Motorola S-Record**

The Motorola S-Record is identical to its standard version when displayed, up to the point that the data's address goes beyond FFFF and thus requires a 5th digit, e.g.: 10000. To compensate for this addition an extra byte is added to the address giving 010000.

When this occurs the record type changes:

The data record changes from 1 to 2
and the end record changes from 9 to 8.

Similarly when the data address goes beyond FFFFFF a 7th digit is required and likewise a byte is added giving the address 8 characters: 01000000.

When this occurs:

The data record changes from 2 to 3
and the end record changes from 8 to 7.

The Motorola S—Record consists of:

(a) a start code, i.e.: S

(b) the record types, i.e.: 1—Data Record (Four Character address)
9—End Record (Four Character address)

2—Data Record (Six character address)
8—End Record (Six character address)

3—Data record (Eight character address)
7—End Record (Eight character address)

(c) The sum of the number of bytes in an individual record, e.g.: 1D

(d) the address of the first byte of data in an individual record, e.g.: 0000, 010000, 01000000

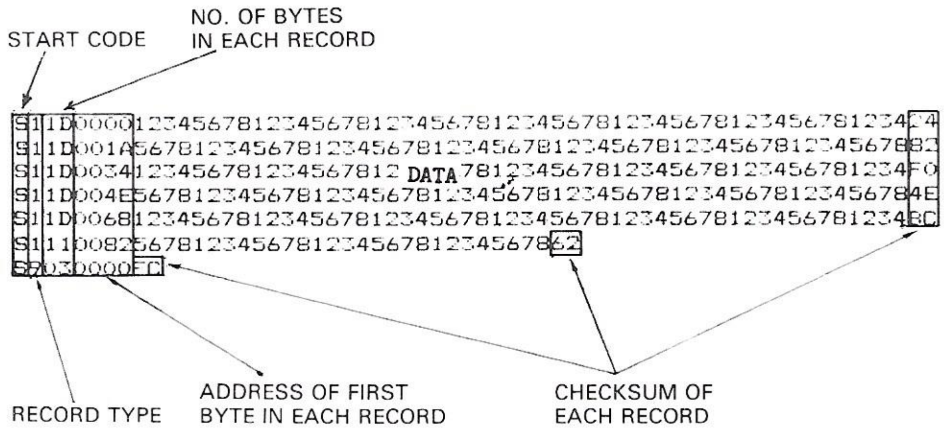
(e) data in bytes, e.g.: 12 34 56 78

(f) checksum of an individual record: 24

1-Data Record (Four Character Address)
9-End Record (Four Character Address) } **2 BYTES**

For example:

START ADDRESS:	0000
STOP ADDRESS:	008F
OFFSET:	0000 0000

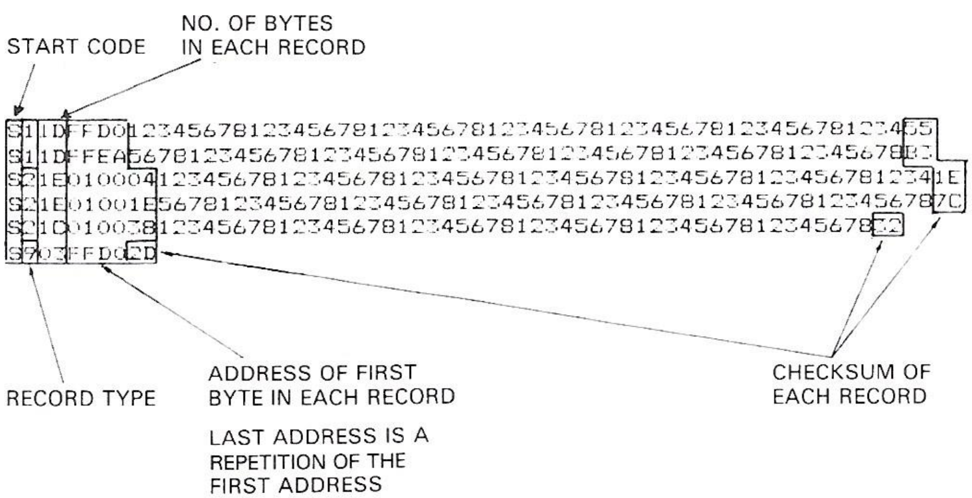


1 — Data Record (Four Character Address)
9 — End Record (Four Character Address) } 2 BYTES

The MOTOROLA S-RECORD format stays identical in layout to that of the standard when the address field stays below FFFF.

Transition from 2 Byte Address (4 CHARACTERS) Through to 3 Byte Address (6 CHARACTERS).

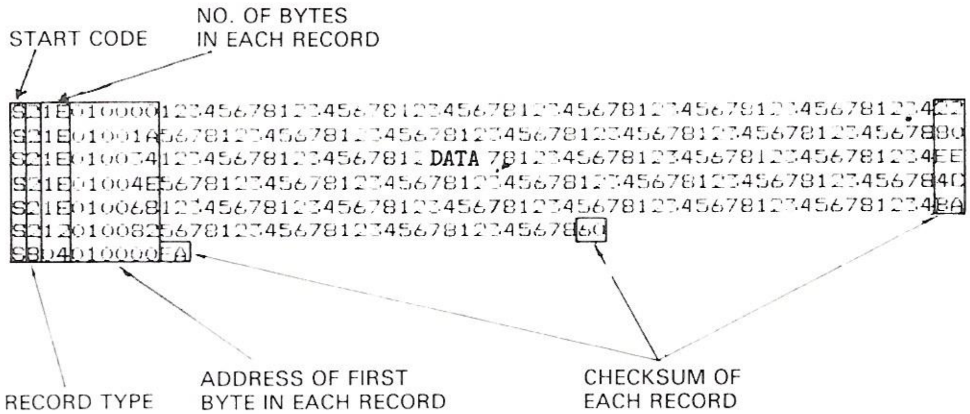
START ADDRESS:	FF80
STOP ADDRESS:	FFFF
OFFSET:	00000050



2-Data Record (Six Character Address)
8-End Record (Six Character Address) } **3 BYTES**

For example:

START ADDRESS:	0000
STOP ADDRESS:	008F
OFFSET:	00010000

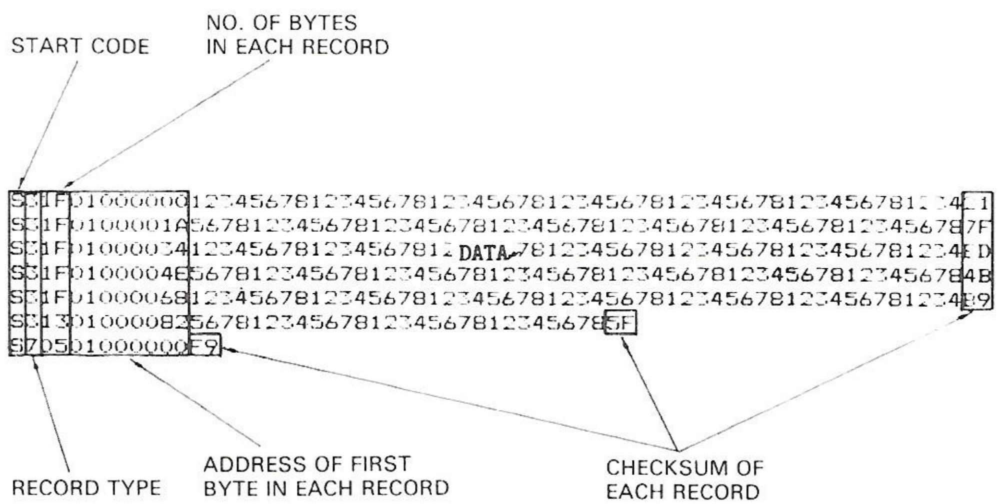


2-Data Record (Six Character Address)
8-End Record (Six Character Address) } 3 BYTES

3-Data Record (Eight Character Address)
7-End Record (Eight Character Address) } **4 BYTES**

For example:

START ADDRESS:	0000
STOP ADDRESS:	008F
OFFSET:	01000000



3-Data Record (Eight Character Address)
7-End Record (Eight Character Address) } 4 BYTES

Calculation of the Motorola S—Record Checksum

```
S11D0000123456781234567812345678123456781234567812345678123424
S104001A568B
S9030000FC
```

EXAMPLE: THE SECOND "DATA RECORD" OF THE ABOVE FORMAT.

(i)	this is:	S1 04 00 1A 56 8B	
(ii)	the start code, the record type and the checksum are removed:	S1 8B	
(iii)	four bytes remain:	04 00 1A 56	
(iv)	these are added together:	04 + 00 + 1A + 56 = 74	
(v)	the total '74' is converted into Binary:	7 0111	4 0100
(vi)	the Binary figure is reversed. This is known as a complement*:	8 1000	B 1011
(vii)	8B corresponds to the checksum as above:	S1 04 00 1A 56 (8B)	

When addition of information occurs in longer records the checksum may consist of more than one byte. When this occurs the least significant byte is always selected to undergo the above calculation.

7.6 **Hex ASCII**

The Hex ASCII format consists of:

DATA ALONE

However, invisible instructions are necessary for operation. These are:

a hidden start character known as Control B.
(02: ASCII Code, STX ASCII character).

a hidden stop character known as Control C.
(03: ASCII Code, ETX ASCII character).

a hidden 'space' character between data bytes
(20: ASCII Code, SP ASCII character)

For example:

START ADDRESS	0000
STOP ADDRESS:	008F

OFFSET: NONE REQUIRED AS
HEX ASCII ALWAYS LOADS AT ZERO

HIDDEN START CHARACTER
(Control B)

HIDDEN SPACE CHARACTERS

```
O12 34 56 78 12 34 56 78 12O34O56O78O12 34 56 78
 12 34 56 78 12 34 56 78 12 34 56 78 12 34 56 78
 12 34 56 78 12 34 56 78 12 34 56 78 12 34 56 78
 12 34 56 78 12 34 56 78 12 34 56 78 12 34 56 78
 12 34 56 78 12 34 56 78 12 34 56 78 12 34 56 78
 12 34 56 78 12 34 56 78 12 34 56 78 12 34 56 78
 12 34 56 78 12 34 56 78 12 34 56 78 12 34 56 78
 12 34 56 78 12 34 56 78 12 34 56 78 12 34 56 78
 12 34 56 78 12 34 56 78 12 34 56 78 12 34 56 78O
```

HIDDEN STOP CHARACTER
(Control C)

16 bytes per line
on output

7.7 **Stag Hex***

The STAG HEX* format consists of:

(a) a start code, i.e. * (an asterisk, 2A—ASCII Code)

(b) the address of the first byte of data in an individual record, e.g. 0000

(c) data in bytes, e.g. 12 34 56 78

(d) a stop code, i.e. $ (a dollar sign, 24—ASCII Code)

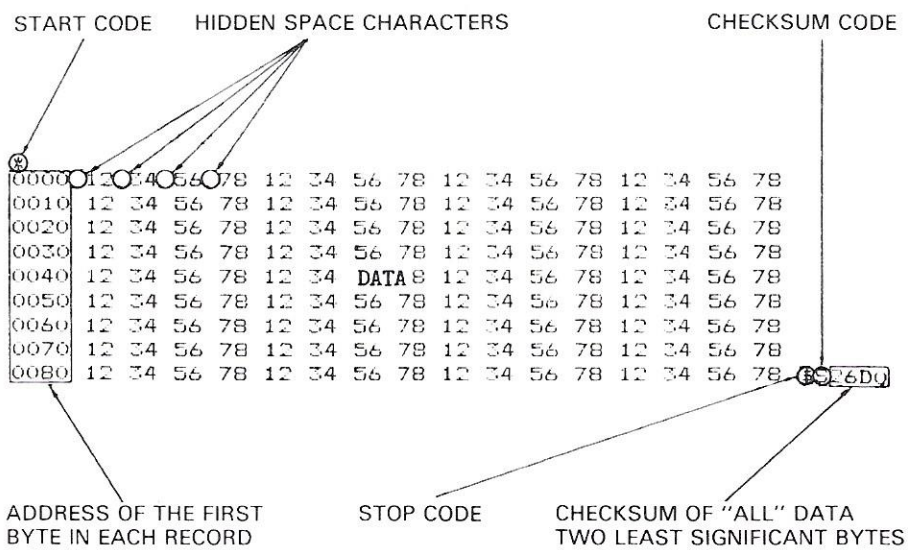
(e) a checksum of all data over the entire address range. (The checksum is the two least significant bytes.)

(f) a checksum start code: S

(g) an invisible space character between data bytes (20-ASCII Code)

For example:

START ADDRESS:	0000
STOP ADDRESS:	008F
OFFSET:	0000



Calculation of the STAG HEX* Checksum

''Data alone'', in bytes over the entire address range (as opposed to individual records) is added together to give the checksum. The address is not included in this calculation.

```
*
0000 12 34 56 78 $S0114
```

EXAMPLE: THE SEGMENT OF DATA ABOVE

(i)	this is:	*0000 12 34 56 78 $S0114
(ii)	the start code, the address, the stop code, the checksum code and the checksum are removed:	*0000 $S0114
(iii)	four Bytes remain:	12 34 56 78
(iv)	these are added together:	12 + 34 + 56 + 78 = 114
(v)	114 is the checksum which is displayed in two byte form as above:	*0000 12 34 56 78 $S0114

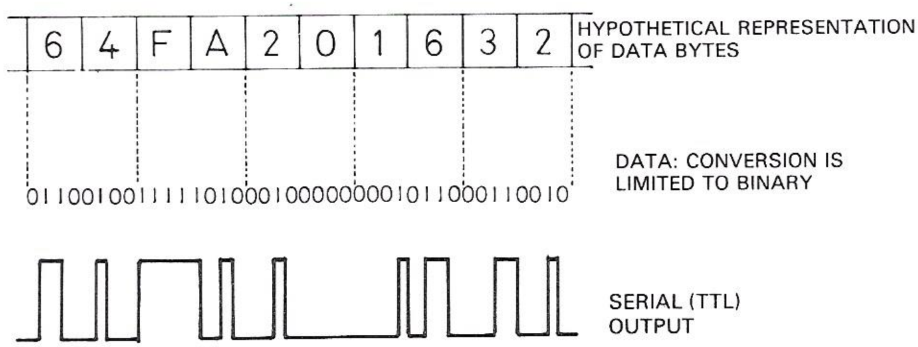
As the STAG HEX* checksum is an addition of all data the total will invariably constitute more than two bytes. When this occurs the least significant 'two' bytes are always selected to undergo the above calculation.

7.8 Binary and DEC Binary

Binary and DEC Binary are the most fundamental of all formats. ASCII code conversion never occurs. Information is therefore limited to the interpretation of pulses via the RS232C interface port into either ONES or ZEROs. Hence 'Binary'. A visual display is not possible.

Binary:

Binary is data only. It is devoid of a start code, address, stop code and checksum.

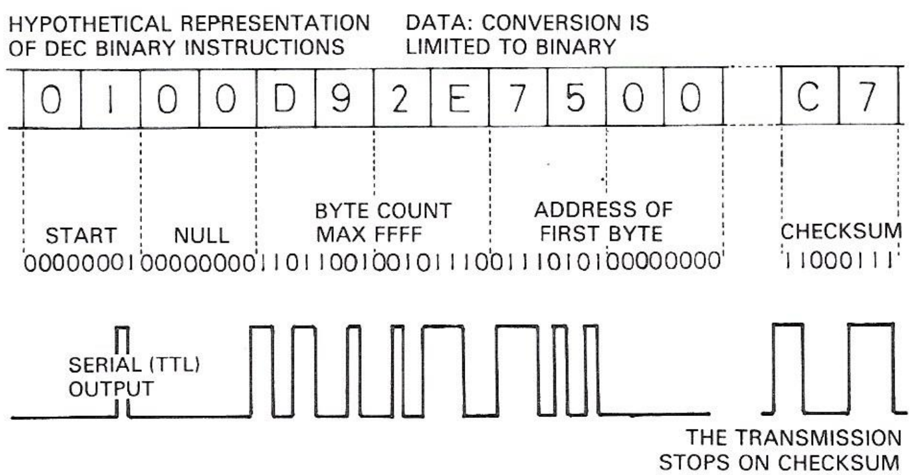


THE BINARY FORMAT OPERATION CAN ONLY BE STOPPED BY PRESSING EXIT.

Binary is used mainly for speed of transmission and RS232C communication problems, i.e. test.

DEC Binary

DEC Binary is an improvement of Binary. It has a start code, a null prior to transmission, a byte count, a single address and a single checksum of all data. It also has the facility for an offset to be set.



The DEC Binary checksum is an addition of all data (data only). The least significant byte is selected to represent the checksum.

DEC Binary is used for speed of transmission.

7.9 **MOS-Technology**

The MOS-Technology format consists of:

i. a start code, ie: ; (semi-colon)

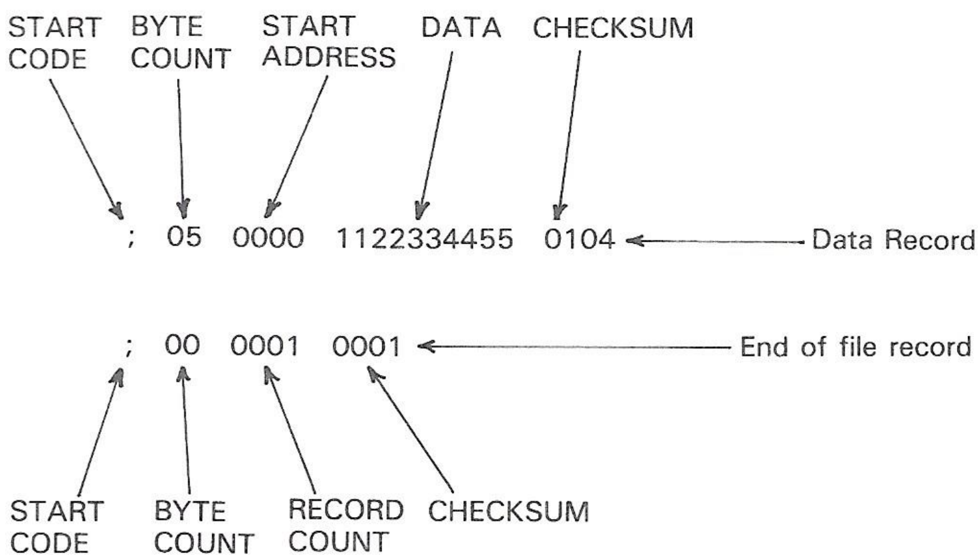
ii a byte count—that is the sum of the number of data bytes in an individual record, eg: 05

iii the address of the first byte of data in an individual record, eg: 0000

iv data in bytes, eg: 11 22 33 44 55. (The data bytes must consist of valid hexadecimal digits).

v. a checksum which is displayed as two hexadecimal bytes. It is the addition of the preceeding data bytes in the record including the address and byte count in hexadecimal form.

For example:



Calculation of MOS-Technology checksum

; 05 0000 11 22 33 44 55 0104
; 00 0001 0001

Example: the first line of the above format.

i. this is ; 05 0000 11 22 33 44 55 0104

ii. the start code and the checksum are removed: ; 0104

iii. this leaves:
 the byte count: 05
 the address of the first byte in the record: 0000
 and five data bytes: 11 22 33 44 55

iv. these are added together:
 05 + 0000 + 11 + 22 + 33 + 44 + 55 = 0104

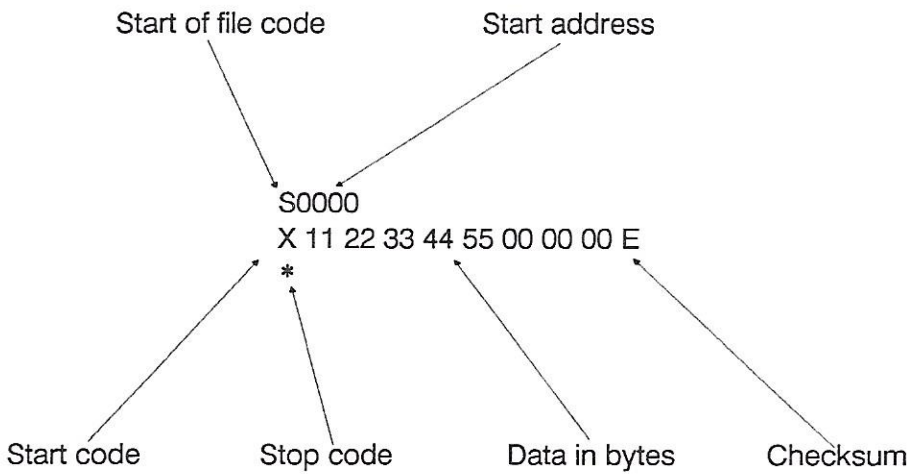
v. 0104 is the checksum as above: ; 0500001122334455(0104)

7.10 Fairchild Fairbug

The Fairchild Fairbug format consists of:

i) a start of file code, i.e. S
ii) the address of the first byte of data in an individual record, e.g. 0000
iii) a start code, i.e. X
iv) data in bytes, e.g. 11 22 33 44 55. (The data must consist of valid hexadecimal digits).
v) a checksum which is expressed as the least significant digit of the nibble addition of the data bytes.
vi) a stop code, i.e. *

For example:

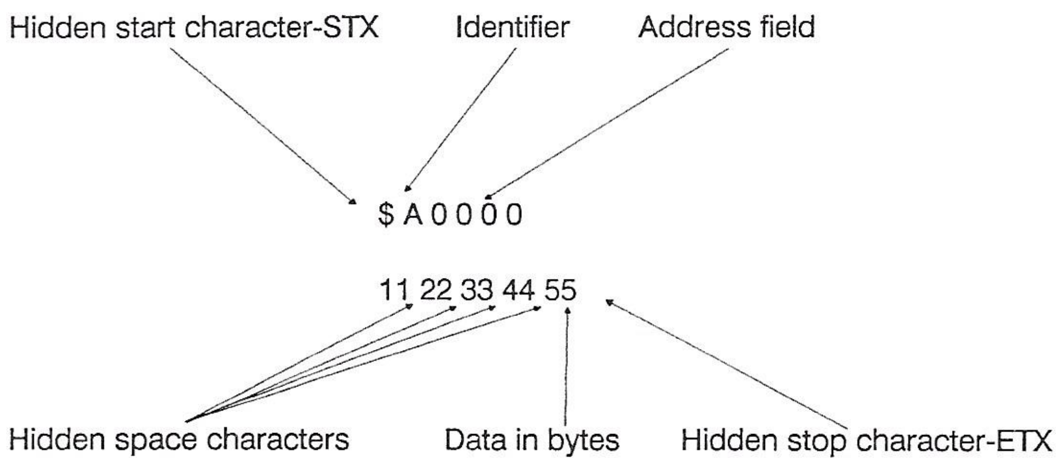


7.11 **ASCII-HEX. (Space)**

The ASCII-Hex (Space) format consists of:

i) a hidden start character STX (optionally SOH)
ii) an identifier $A denoting that an address field follows
iii) an address field
iv) data in bytes
v) a hidden space character between data bytes
vi) a hidden stop character ETX

For example:



SECTION 8

8

stag

Sophisticated systems for the discerning engineer.

8 Remote Control (Memory Devices Only)

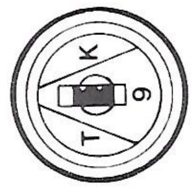
8.1 Selecting the Operating Modes

Selection of the operating mode is made by use of the three position key-switch located underneath the hinged flap. By inserting the supplied key one of these modes can be chosen.

Local Mode (arrow vertical)

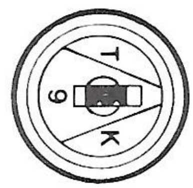
In LOCAL mode all functions of the programmer are controlled from its own keyboard. All the functions operate interactively with the CRT and Input/Output transmissions take place between the programmer and the peripheral equipment under 'Local Control'.

Remote Mode (arrow pointing to the left)



In REMOTE the programmer operates under remote control from a computer or a terminal. The keyboard of the programmer is inoperative at this time and the CRT will only display information as requested under remote command.

Edit Lock-Out Mode (arrow pointing to the right)



The Edit Lock-out mode ensures all editing key functions are inoperative but that all other functions operate normally under LOCAL MODE conditions.

In this mode data corruption or changes to the RAM data cannot take place and security of information is ensured.

Interchange between LOCAL and REMOTE MODE can be accomplished by turning the key-switch to the required direction and pressing RESET.

Interchange between LOCAL and EDIT LOCK-OUT can be made by simply turning the key to the required direction. (Should an edit key function be requested under Edit Lock-Out Mode the CRT will display: EDIT LOCKED OUT).

8.2 Setting Up Procedure for Remote Control

Before engaging REMOTE CONTROL, two selections have to be made:

1. the required interface.
2. the REMOTE CONTROL mode.

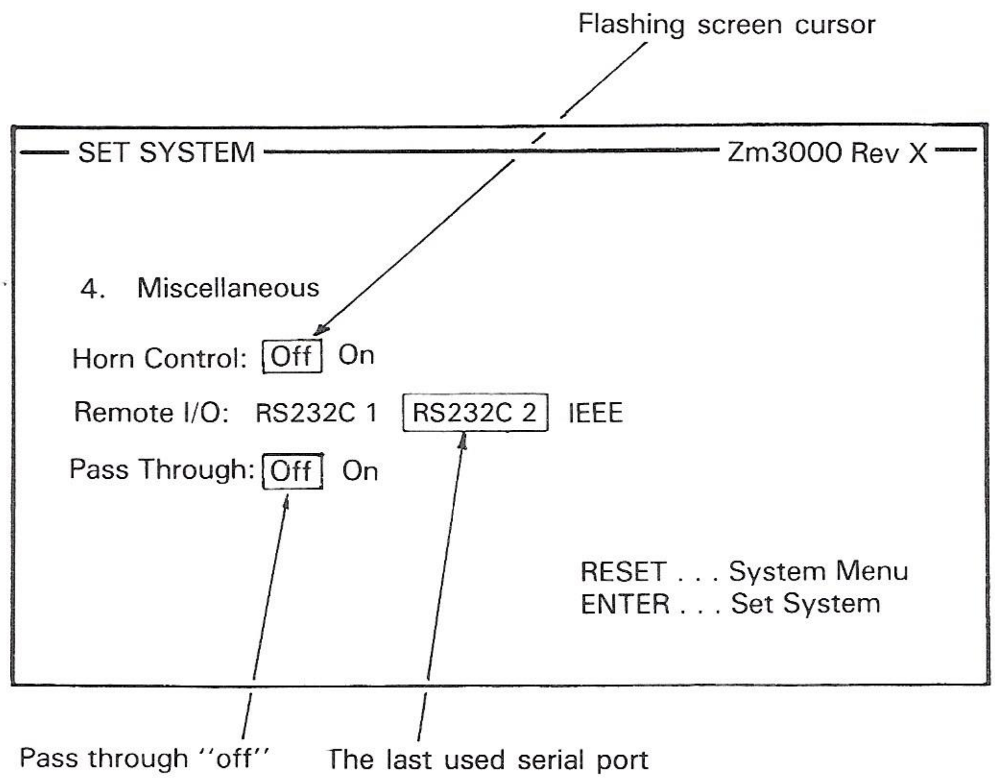
Selecting the required interface

From the SYSTEM MENU:

PRESS KEY A followed by KEY 4

This will display the 'Miscellaneous' page of the Set System option which will show a flashing screen cursor defaulted to HORN CONTROL and two static cursors. One of the static cursors will indicate the interface port last used in the REMOTE CONTROL mode and the other will indicate whether PASS-THROUGH IS 'on' or 'off'.

For example:

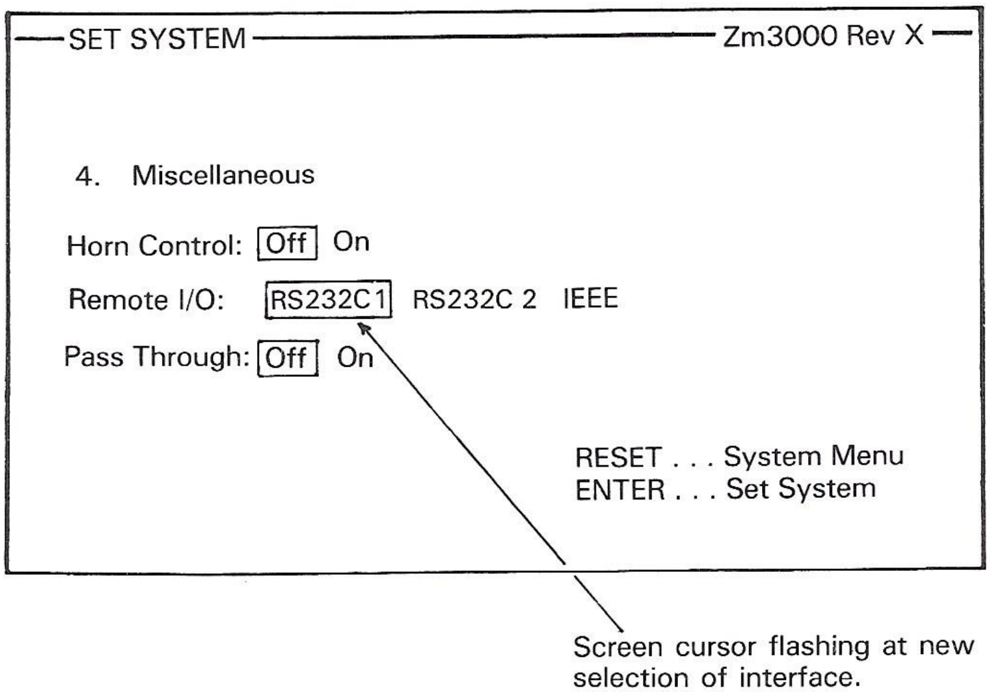


If, as in this example the RS232C 2 port was previously used in REMOTE MODE and RS232C 1 is the new requirement, then it can be selected as follows:

PRESS KEY ↓

This will move the flashing cursor onto RS232C 2. To select RS232C 1:

PRESS KEY ←



To confirm this selection:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

Selecting the address parameter for the IEEE

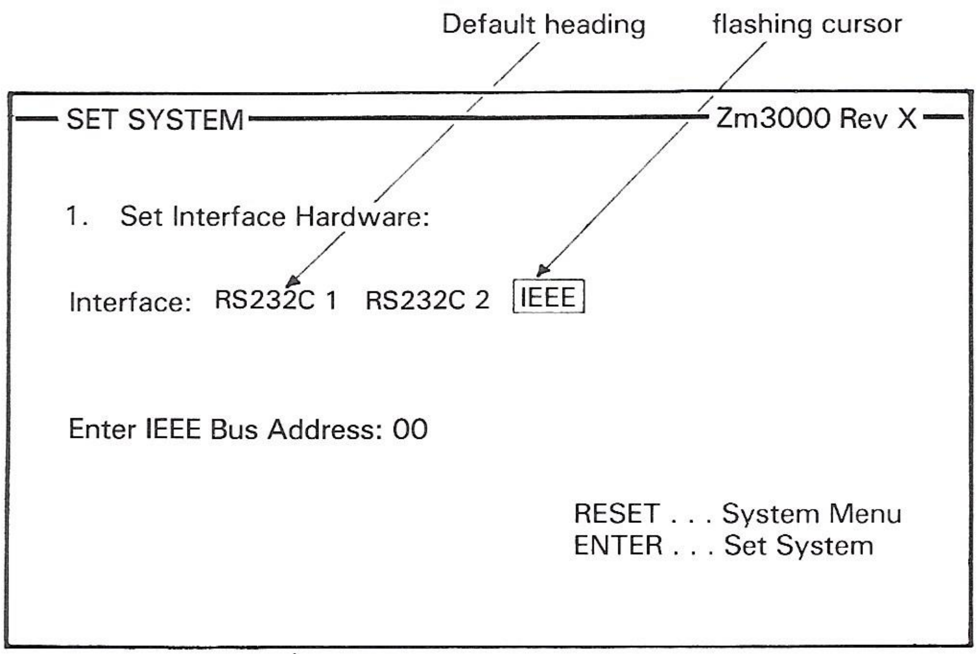
A two digit coded address can be entered that will allow a remote-controller connected via the IEEE interface port to identify the Programmer.

To select the address parameter from the SYSTEM MENU:

PRESS KEY A followed by KEY 1

This will display the screen cursor defaulted to the RS232C 1 heading. To select the IEEE:

PRESS KEY → twice



Pressing the down cursor key will make the address available for selection, indicated by the pre-set parameter flashing. The new address can now be entered from the keyboard. To confirm this selection:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

Selecting REMOTE CONTROL mode

Selecting the REMOTE MODE involves the 3-position key switch which is located beneath the hinged flap on the top panel of the System 3000.

To select REMOTE MODE from LOCAL MODE:

TURN THE KEY CLOCKWISE (arrow pointing left)

PRESS RESET

The screen will display:

```
── REMOTE CONTROL ──────────────── Zm3000 Rev X ──
```

The programmer is now configured for operation in REMOTE CONTROL mode via either of the serial interface ports and the keyboard of the Programmer will be inoperative.

To return to the LOCAL mode:

TURN THE KEY COUNTER CLOCKWISE ONE STOP (arrow pointing down)

PRESS RESET

The SYSTEM MENU will now be displayed.

Structure of the IEEE as a Talker/Listener Port

Certain communication procedures govern the set-up and operation of the Programmer in the REMOTE CONTROL mode with a controller via the IEEE interface port.

Some of the communication is inherent to the controller and some is Programmer based but both are interlinked and will respond to interpretations of instructions generated by the other.

Note: The communication procedure must be followed precisely, in order to prevent ''bus lock-up'' (a state whereby communication between the two machines breaks down).

This procedure can be shown in the form of two flow charts.

(i) The Programmer as a talker (TXD)

System 3000

An SRQ (Service Request) is automatically issued and the Programmer listens for an S.Poll to be transmitted from the controller.

CONTROLLER

In response to the SRQ the controller Serial Polls the Programmer.

The Programmer responds by sending decimal 73.

73 is interpreted by the controller as Programmer "request to talk". The controller addresses the Programmer as a talker and waits for S.Poll = 0

The Programmer in response to being addressed as a talker sends out S.Poll = 0. It then transmits data TXD

The controller receives the data RXD

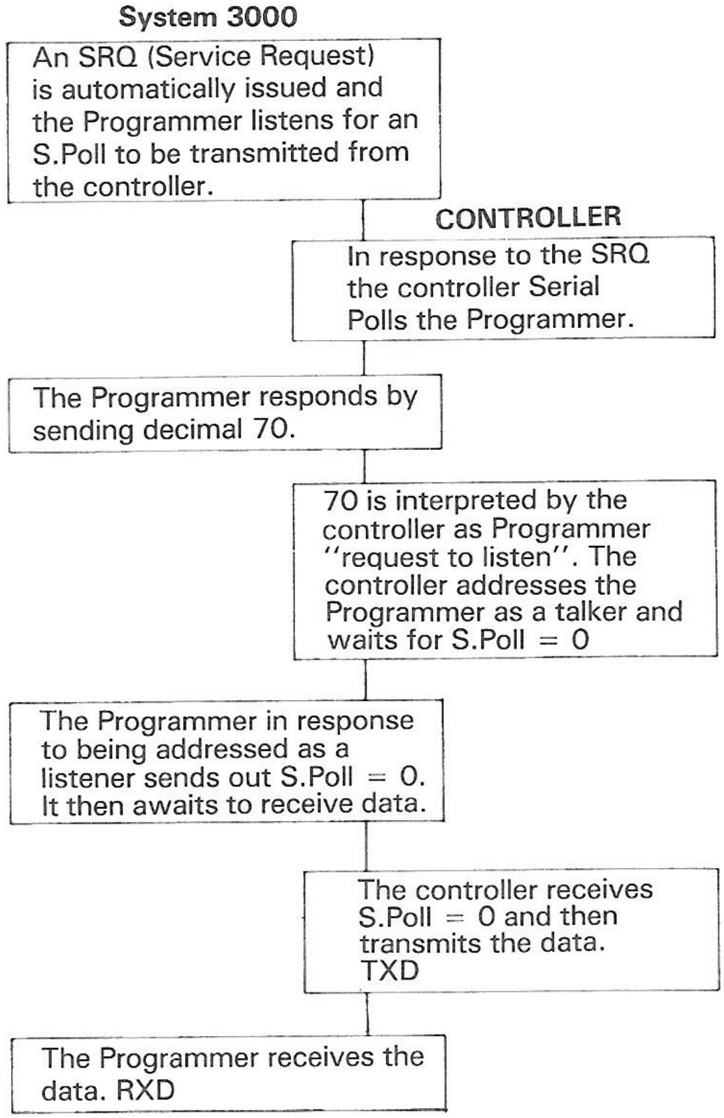
Notes:

(i) on power-up the Programmer transmits a pass response.

(ii) when the Programmer is not transmitting or receiving data, the S.Poll will equal decimal 128. This occurs during programming, load, etc.

(iii) a rolling cursor will indicate a dump is in progress.

(ii) The Programmer as a listener (RXD)



Notes:

(i) on power-up the Programmer transmits a pass response.

(ii) when the Programmer is not transmitting or receiving data, the S.Poll will equal decimal 128. This occurs during programming, load, etc.

(iii) a rolling cursor will indicate a load is in progress.

IEEE 488 Interface port capabilities

Note: An abbreviated list of the IEEE capabilities is printed on the back panel below the port itself.

SH1	Full source handshake capability.
AH1	Full acceptor handshake capability.
T2	Full talker function capability and full serial Poll capability.
L2	Full listener function capability.
SR1	Full service request function capability.
RL1	No remote/local function capability.
PP1	Parallel Poll capability without direct configuration from controller
DCO	No capability for device clear function.
DTO	No capability for device trigger.
CO	No controller function.
E1	Open collector drivers.

The following gives the signals on the IEEE 24 way connector:

Pin No	Name
1	DIO1
2	DIO2
3	DIO3
4	DIO4
5	DOI
6	DAV
7	NRFD
8	NDAC
9	IFC

Pin No	Name
13	DIO5
14	DIO6
15	DIO7
16	DIO8
17	REN
18	GND
19	GND
20	GND
21	GND

8.3 Remote Control Commands

To execute commands via a serial interface, send a CR,LF (Carriage Return, Line Feed).
To execute commands via the IEEE-488, send an EOI.

'Set' Commands

S0	MMDDD	Select the device—MM = manufacturer code, DDD = device code. (In the case of a 4 digit code, insert a zero between the first and second pair of digits).
S1	N	Set I/O format N. See table 1 for list of format codes.
S3	N	Set security fuse(s) status and encryption status. Micros only. Device dependent. See table 2.
S4		Set the RAM to the unprogrammed state.
SR	AAAAA	Set RAM start.*
SD	AAAAA	Set device start.
SE	AAAAA	Set RAM end.*
SO	AAAAA	Set I/O offset.

*The RAM start and RAM end are related to device, I/O and RAM functions and should be downloaded before any of these functions is executed.

Note: For the above commands, AAAAA is a variable length hex number which can have leading zeros suppressed. AAAAA represents a byte address.

Table 1—Format Codes

Code	Format	Code	
4	Intel hex	B	DEC Binary
5	Motorola S-record	C	Binary Rubout
6	Tek-Hex	D	Binary
7	Extended Te[illegible]	[illegible]	[illegible]

Table 2—Security Fuse and Encryption Status (See also Section 4.5)

Parameter 'N'	Function
0	Fuse(s) intact, No encryption.
1	Fuse 1 blown, Fuse 2 (if present) intact, No encryption.
3	Fuse 1 blown, Fuse 2 blown, No encryption.
4	Fuse 1 intact, Fuse 2 intact, Encryption set.
5	Fuse 1 blown, Fuse 2 intact, Encryption set.
7	Fuse 1 blown, Fuse 2 blown, Encryption set.

'Read' Commands

R0	Read device code.
R1	Read format currently selected.
R3	Read security fuse and encryption status. Micros only. Device dependent.
R4	Read Cyclic Redundancy Check (CRC).
R5	Read RAM size.
R6	Read software revision number.
R7	Read RAM checksum.
R9	Read device size (hex) /byte size (bits in hex) /unprogrammed state. Unprogrammed state FF = 0, Unprogrammed state 00 = 1.
RR	Read RAM start.*
RD	Read device start.*
RE	Read RAM end.*
RO	Read I/O offset.*

*These have variable leading zero suppressed.

Device Commands (Memory Devices Only)

P0	(also P4)	Program with full autosequence.
P1	(also P5)	Program without pre-program checks.
P2	(also P6)	Program without post-program verify.
P3	(also P7)	Program without pre- or post-programming checks.
I		Load device to RAM.
E		Empty check.
V		Verify.

RAM and I/O Commands

I	Input data to RAM from the interface.
O	Output data from RAM to the interface.
F hh	Fill RAM between previously set limits with hex string hh. Maximum string length of 16 characters - 8 bytes.

Miscellaneous

H nn	Sound horn n times. 'n' is in decimal.
D string	Display ASCII string.
K	Wait for any key to be pressed.
Z	Escape from remote control.

Responses to Commands

Some commands require the programmer to output data to the controlling equipment. In general, the data is of the same form as that for the corresponding 'Set' command.

For instance, the RS232C response to the RO (read device code) command will be:

MMNNN, CR, LF, SS, CR, LF, >

Where SS is the status response.
The IEEE-488 response will be:

MMNNN (EOI), status returned in status byte.

RS232C Status Responses

Hex Code	Status
00	Command executed O.K
01	No Blow
02	Verify Error
03	Illegal Bit
04	Not Empty
05	Connect Error
06	Reversed Device
08	Out of range address on download
09	Security fuse fail (Micros only)
0A	Illegal or Unrecognised command
0B	Load Error on Download
0D	RAM Fail

IEEE 488 Status Responses

The IEEE 488 responses are placed in the status byte. Bit 7 (MSB) of the byte is used to indicate that the programmer is busy and cannot accept remote control commands other than serial and parallel poll commands. If the programmer is addressed as a listener while bit 7 is set, it will assert NRFD (Not Ready For Data) and lock up the bus until the command is completed. If the programmer is addressed as a talker, it will not respond until bit 7 goes to zero.

Bit 6 of the status byte is the SRQ bit and indicates that the programmer is generating an SRQ. The programmer will generate an SRQ on completion of the E, V, P and K commands.

Bit 5 of the byte is used to indicate that the programmer has data to be transmitted in response to a command. A zero bit indicates no data and a one bit indicates data. If there is data to be transmitted, the IEEE 488 controller must read it before another command can be accepted.

Bits 4 to 0 provide a five bit code supplying error information. The codes are the same as for the RS232C responses.

Additional IEEE 488 Remote Control Commands

The IEEE 488 remote control commands are similar to those specified for the serial interface ports but an important addition should be noted.

EOI is used to terminate data strings, commands and error responses. The manner in which EOI is sent is dependent upon the way the 'controller' has been configured.

Logic Device Programming Sections 9 - 15

SECTION 9

9

stag

Sophisticated systems for the discerning engineer.

9 Device Selection (Logic Devices Only)

9.1 Selecting the Device to be Programmed

Device selection on the System 3000 is very straightforward.

Step 1—Set Family

Firstly, the correct 'family' of device has to be chosen, i.e. Prom/Eprom
Logic
Microcomputer

If the 'family' is already correct, proceed to Step 2—Set Device.

To enter the Set Family option:

Press Key 0

This will display the Set Family menu. Press Key 2 to select the logic device category. The logic device main system menu will be displayed as well as the last chosen device in that category.

Step 2—Set Device

Press Key 1

This will display the Select Manufacturer screen. The information is displayed on more than one page and is arranged alphabetically by manufacturer name. Interchange between the available pages is made with the horizontal cursor keys ← and →.

Selection of a particular Manufacturer/Device category is made with the vertical cursor keys ↑ and ↓ or by direct input of the manufacturer code, e.g. 10 for Signetics FPLA.

Press Enter to confirm the manufacturer selection.

This will also display the list of devices available for this manufacturer. If more than one page of devices is available, it can be selected by using either of the horizontal cursor keys ← or →. The device can be selected by scrolling through the list using the vertical cursor keys ↑ and ↓ or entering the three digit pinout code from the keyboard.

Press Enter to confirm the selection.

Pressing Reset at any time during the selection process will cause the previously selected device to be re-instated and the main system menu will be returned.

SECTION 10

10

stag

Sophisticated systems for the discerning engineer.

10 Device Functions

10.1 List Device

LIST DEVICE is a feature that enables the pattern of a device in the appropriate ZIF socket to be displayed on the screen.

Note:
Because of the continual demand for specialist features, the displayed fuse pattern of PLDs varies widely not only between classes of device, i.e. PALs and IFLs but also between individual devices within each group. As a consequence the LIST DEVICE mode is not a uniform feature. Special commands* have been added for ease of use on many devices, e.g.
A = Mode, E = PL, A = AR, B = SP, C = PL, D = RG, etc. This allows non-Product Terms to be directly displayed simply by pressing a single key: A, B, C, etc.

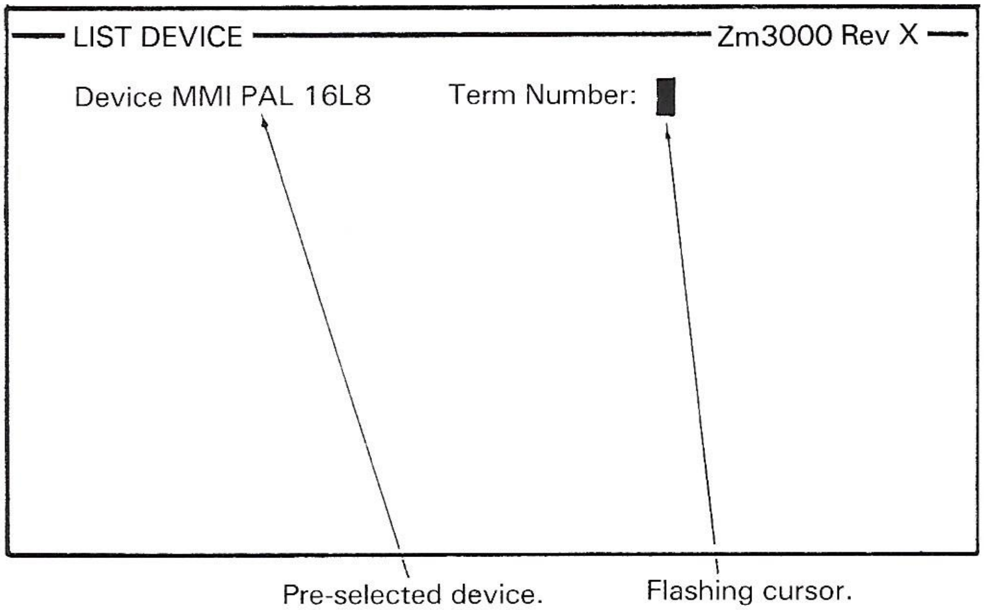
In order to cover this broad diversity as comprehensively as possible, several representative device examples will be shown in this section:

(a) An MMI 20 Pin PAL 16L8 (Code 20029)
(b) A National 24 Pin DMPAL 20R8 (Code 31057)
(c) An AMD 24 Pin AmPAL 22V10 (Code 91070)
(d) An Altera 20 Pin EPLD EP300 (Code 66049)
(e) A Signetics 20 Pin FPLS 82S159 (Code 14008)

*Identical commands are also used in other Zm3000 functions, ie EDIT RAM, LIST RAM and LIST COMPARE.

To enter into the LIST DEVICE function press key 6.

The display will show:



The display shows a flashing cursor indicating that a term number should be entered to display a section of fuse array.

10.1.1 **EXAMPLE (a): MMI 20 Pin PAL 16L8**

With 20 Pin PAL devices all Product Terms associated with one particular output (including the unused "phantom fuses") are displayed together. MMI's 20 Pin PAL 16L8, has 64 Product Terms divided equally between 8 outputs. This architecture is typical for all manufacturers with this type of device.

A section of this device's data can be displayed in two ways:

(i) by entering the required two digit Product Term number by use of the keyboard
(ii) by pressing the down cursor key.

A typical display for the device is shown below:

Defaults to the first Term line on each display

```
— LIST DEVICE ———————————————————————————————— Zm3000 Rev X —

   Device: MMI PAL 16L8          Term Number: 00

           Term                 Input Lines
            No  0123 4567 8901 2345 6789 0123 4567 8901
            00  ---- ---- ---- ---- ---- ---- ---- ----
            01  ---- ---- ---- -X-- ---- -X-- ---- ----
            02  ---- ---- ---- -X-- ---- ---- X--- ----
            03  ---- ---- ---- ---- ---- X--- X--- X---
            04  ---- ---- ---- X--- ---- X--- ---- X---
            05  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
            06  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
            07  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
```

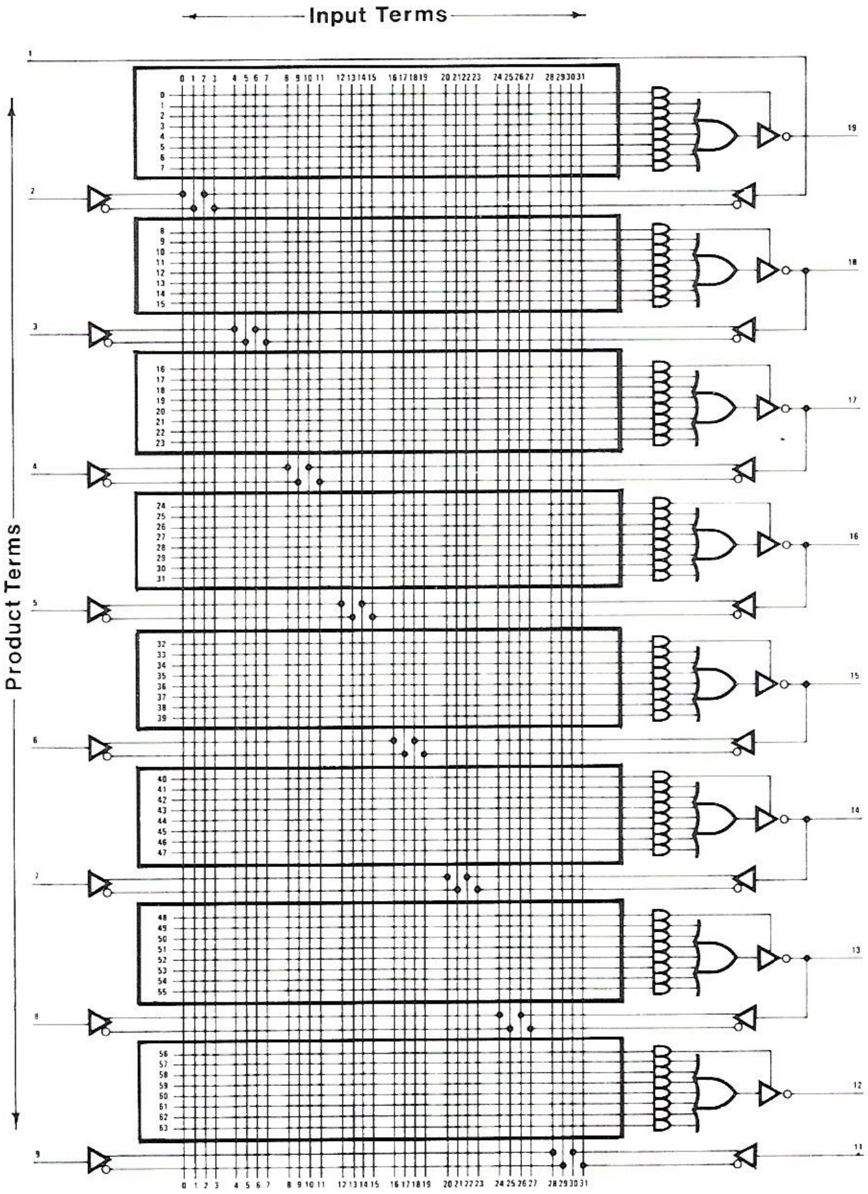
8 Product Terms are displayed simultaneously.

X = Connection
– = No Connection

Scanning of the device can continue by implementing instructions (i) or (ii) again.

An interpretation of the displayed abbreviations and acronyms* used for this device in List Device is shown below on a logic diagram:

16L8



*Identical abbreviations and acronyms are used in List RAM, List Compare and EDIT RAM

10.1.2 **EXAMPLE (b): National 24 Pin DMPAL 20R8**

With 24 Pin PAL devices, Product Terms associated with one particular output are displayed together*.

National's 24 Pin DMPAL 20R8, has 60 Product Terms divided equally between 8 outputs connected to fixed registers, plus 20 Phantom Terms (indicated by dots).

A section of this device can be displayed in 2 ways:

(i) by entering a two digit Product Term by use of the keyboard

(ii) by pressing the down cursor key

A typical Product Term display for the device is shown below:

Defaults to the first Term
line on each display.

```
── LIST DEVICE ──────────────────────────────── Zm3000 Rev X ──

   Device: National DMPAL 20R8      Term Number: 64

     Term                        Input Lines
      No  0123 4567 8901 2345 6789 0123 4567 8901 2345 6789
      64  XXX- XXX- XXX- XXX- XXX- XXXX XXXX XXXX XXXX XXXX
      65  X-XX XX-X XXX- XXX- XXX- XXXX XXXX XXXX XXXX XXXX
      66  XXXX X-X- XX-X XXX- XXX- XXXX XXXX XXXX XXXX XXXX
      67  XXXX XX-X XX-X XXXX XXX- XXXX XXXX XXXX XXXX XXXX
      68  XXXX XXX- XXX- X--X XXX- XXXX XXXX XXXX XXXX XXXX
      69  XXXX XX-X XXX- XX-X XXX- XXXX XXXX XXXX XXXX XXXX
      70  XXXX XXX- XX-X XX-X XXX- X-XX XXXX XXXX XXXX XXXX
      71  XXXX XX-X XX-X XX-X XXX- XXXX X-XX XXXX XXXX XXXX
```

8 Product Terms
are displayed

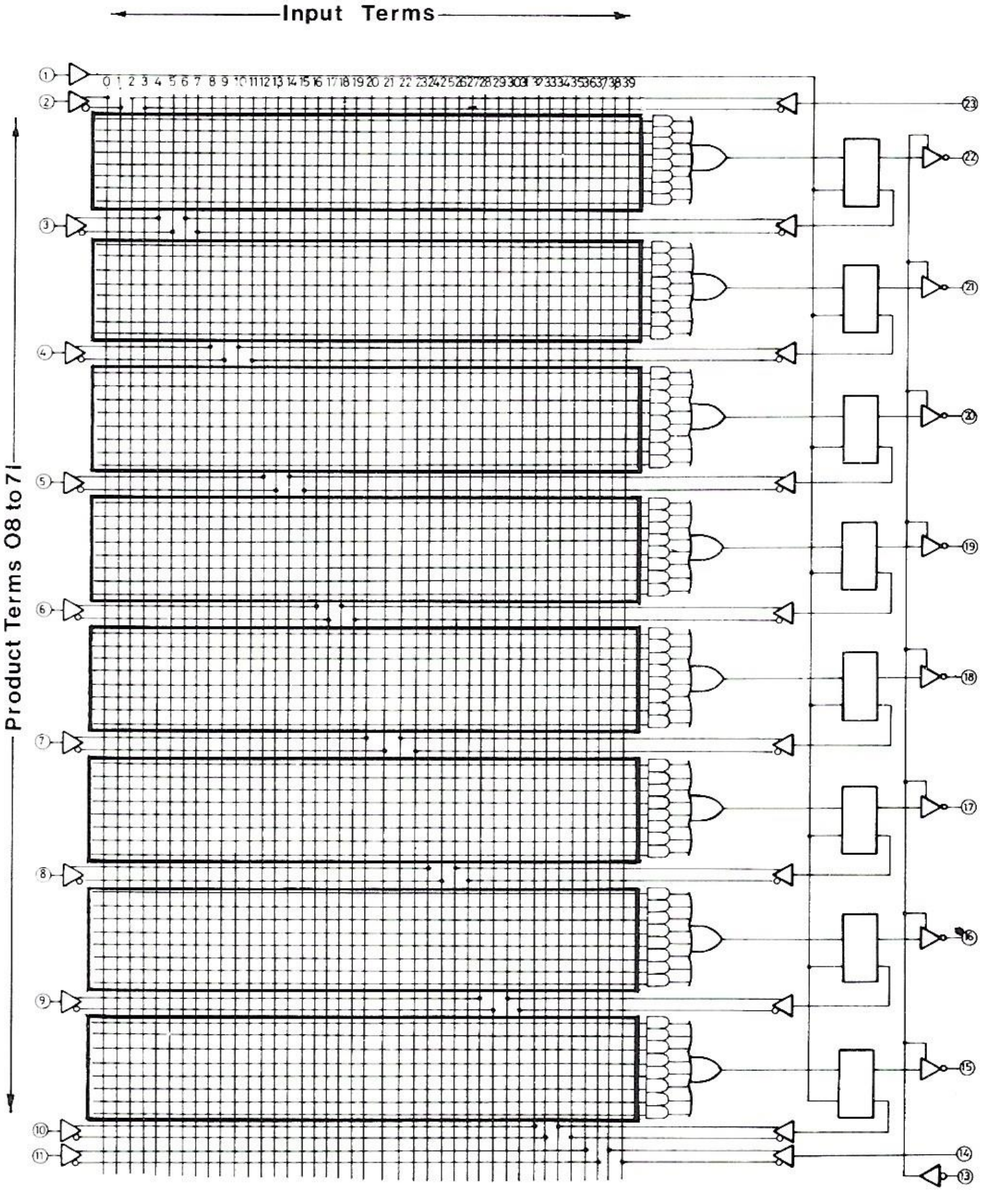
X = Connection
– = No Connection

Scanning of the device can continue by implementing instructions (i) or (ii) again.

*With the exception of AMD's 24 Pin AmPAL 22V10 (See next device example).

An interpretation of the displayed abbreviations and acronyms* used for this device in 'List Device' is shown below on a logic diagram:

20R8



*Identical abbreviations and acronyms are used in List RAM, List Compare and EDIT RAM.

10.1.3 EXAMPLE (c): AMD 24 Pin AmPAL 22V10

AMD's 24 Pin AmPAL 22V10, differs from other 24 Pin PAL devices in three ways:

The number of Product Terms associated with a particular output varies throughout the device from a minimum of 8 to a maximum of 16.

When Product Terms exceed a total of 10 for a particular output, a second screen page is utilized to list the remaining Terms.

Product Term numbers consist of 3 digits.

The programmable architecture of the device consists of:
120 Product Terms disproportionately divided between 10 Outputs thus: 2 x 8, 2 x 10, 2 x 12, 2 x 14, 2 x 16.

10 Output Enables: 00E, 10E, 20E, etc (displayed with the Product Terms).

1 Asynchronous Reset Term (AR) displayed alone

1 Synchronous Preset Term (SP) displayed alone

10 Macro Cells connected to each Output containing programmable Polarity (PL) and programmable Registers (RG).

A section of the device can be displayed and scanned in 3 ways:

(i) by entering a three digit Product Term number by use of the keyboard

(ii) by pressing the down cursor key

(iii) by pressing any of the keys: A, B, C, or D

AMD 24 Pin AmPAL 22V10: Product Terms

When Product Terms exceed a total of 10 for a particular output on this device, a second screen page is utilized to list the remaining Terms.

A typical example is shown below:

SCREEN PAGE ONE

Defaults to first term line on each display

```
— LIST DEVICE ————————————————————————————— Zm3000 Rev X —

  Device: AMD AmPAL 22V10      Term Number: 40E

 Term                           Input Lines
  No   0123 4567 8901 2345 6789 0123 4567 8901 2345 6789 0123
  40E  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
  400  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX ----
  401  ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
  402  ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
  403  ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
  404  ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
  405  ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
  406  ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
  407  ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
  408  ---- ---- XXXX XXXX XXXX XXXX XXXX ---- XXXX XXXX XXXX

            A = AR   B = SP   C = PL   D = RG
```

SCREEN PAGE TWO

```
— LIST DEVICE ————————————————————————————— Zm3000 Rev X —

  Device: AMD AmPAL 22V10      Term Number: 409

 Term                           Input Lines
  No   0123 4567 8901 2345 6789 0123 4567 8901 2345 6789 0123
  409  ---- XXXX XXXX XXXX XXXX XXXX ---- XXXX XXXX XXXX ----
  410  XXXX XXXX XXXX XXXX XXXX ---- XXXX XXXX XXXX ---- XXXX
  411  XXXX XXXX XXXX XXXX ---- XXXX XXXX XXXX ---- XXXX XXXX
  412  XXXX XXXX XXXX ---- XXXX XXXX XXXX ---- XXXX XXXX XXXX
  413  XXXX XXXX ---- XXXX XXXX XXXX ---- XXXX XXXX XXXX XXXX
  414  XXXX ---- XXXX XXXX XXXX ---- XXXX XXXX XXXX XXXX XXXX
  415  ---- XXXX XXXX XXXX ---- ---- ---- ---- ---- ---- ----

            A = AR   B = SP   C = PL   D = RG
```

16 Product Terms are displayed
(plus 1 Output Enable: 40E)

commands for
non-Product Terms

X = Connection
\- = No Connection

AMD 24 Pin AmPAL 22V10: Programmable Polarity (PL) and Registers (RG)

PRESS KEY C or D

A typical display is shown below:

Macro Cell Numbers

```
— LIST DEVICE ——————————————————————————————— Zm3000 Rev X —

Device: AMD AmPAL 22V10          Term Number: PL

                  Term
                   No    0 1 2 3 4 5 6 7 8 9
                   PL    X - - X - X X - - X
                   RG    X - - X X - X X - -


              A = AR   B = SP   C = PL   D = RG
```

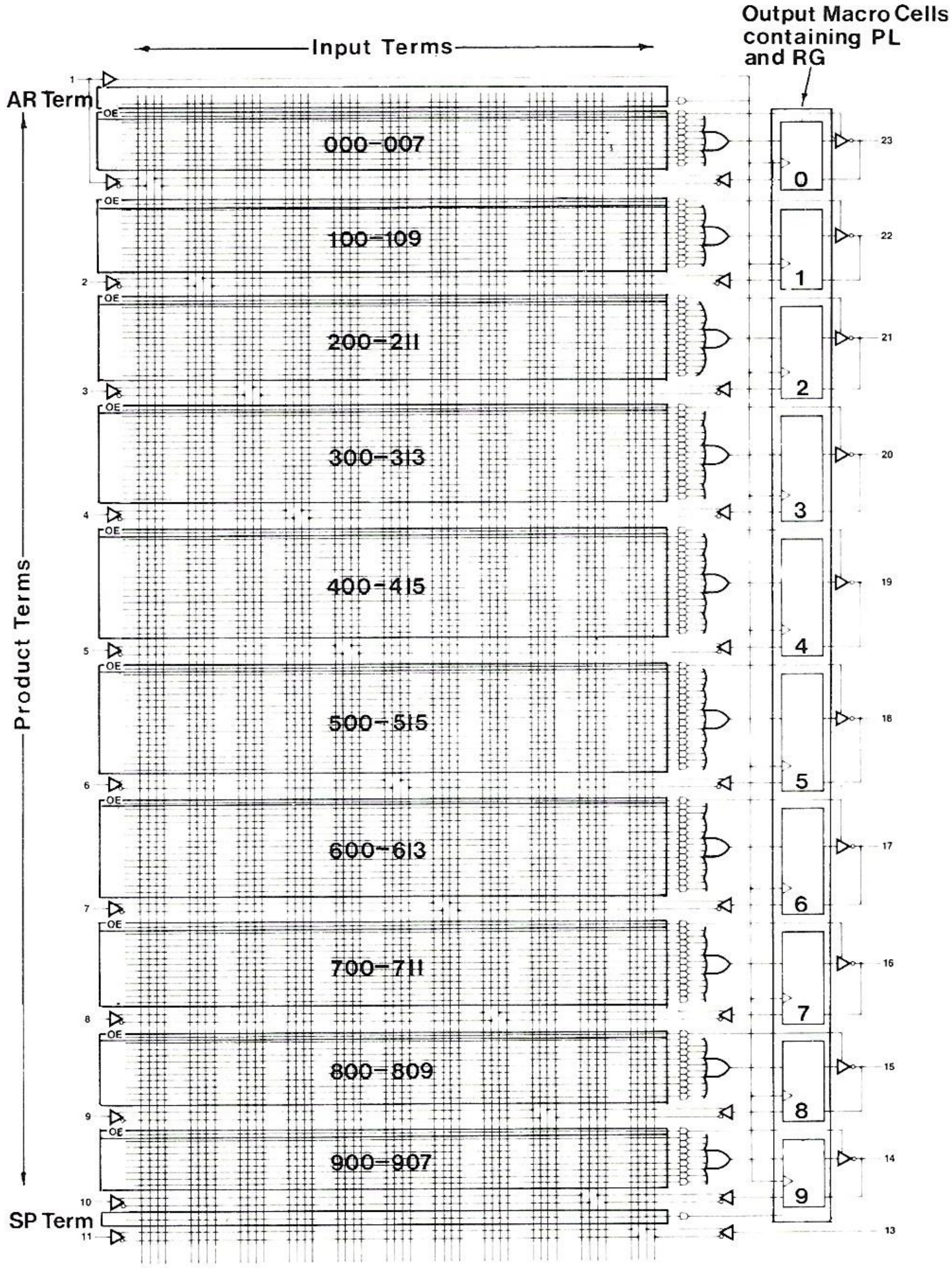
The programmable Polarity and Register operate interactively within the Output Macro Cell to produce one of four modes, as listed below:

Register	Polarity	Output Configuration
X	X	Registered/Active Low
X	—	Registered/Active High
—	X	Combinatorial/Active Low
—	—	Combinatorial/Active High

X = Connection
- = No Connection

An interpretation of the displayed abbreviations and acronyms* used for this device in 'List Device' is shown below on a logic diagram:

AMD AmPAL 22V10



*Identical abbreviations and acronyms are used in List RAM, List Compare and EDIT RAM.

10.1.4 **EXAMPLE (d): Altera 20 pin EPAL: EP300**

The programmable architecture of Altera's EP300 Erasable PAL consists of:

64 Product Terms equally divided between 8 outputs thus: 8 x 8.

8 Output Enables: 0E, IE, 2E, etc (displayed with the product Terms).

1 Asynchronous Reset (AR) Term displayed alone.

1 Synchronous Preset (SP) Term displayed alone.

8 I/O Architecture Control Cells each one consisting of 4 types of programmable outputs and 3 types of programmable feedback. The flexibility within these cells allows registered or combinational output in either active high or active low mode to each output pin. In addition, the feedback path can be programmed independently of the output to be either combinational, registered or I/O.

A section of this device can be displayed in 4 ways:

(i) by entering a two digit Product Term number by use of the keyboard.
(ii) by pressing the down cursor key.
(iii) by pressing one of the keys: E . . . for Asynchronous Reset or F . . . Synchronous Preset.
(iv) by pressing key . . . A followed by key . . . 1 to display the I/O architecture array.

A typical Product Term display for the device is shown below:

Defaults to the first term line on each display

```
— LIST DEVICE ——————————————————————————— Zm3000 Rev X —

  Device Altera EP300               Term Number: 2E

   Term                  Input Lines
    No  0123 4567 8901 2345 6789 0123 4567 8901 2345
    2E  ---- ---- ---- ---- ---- ---- ---- ---- ----
    20  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
    21  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
    22  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
    23  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
    24  ---- ---- ---- ---- ---- ---- ---- ---- ----
    25  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
    26  XX-- XX-- XX-- XX-- XX-- XX-- XX-- XX-- XX--
    27  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

                      E = AR  F = SP
```

commands for non-product Terms

8 Product Terms are displayed
(plus 1 output enable: 2E)

X = Connection
– = No Connection

A typical display for the devices I/O architecture is shown below:

Press key . . . A followed by key . . . 1

LIST DEVICE — Zm3000 Rev X

Device: Altera EP300 Term Number: A1

Cell No	Output 1	2	3	4	Feedback 1	2	3
A1	X	X	-	X	X	X	-
A2	X	X	-	X	X	-	X
A3	-	-	-	-	-	-	-
A4	-	-	-	-	X	X	X
A5	-	-	-	-	X	X	X
A6	-	-	-	-	-	-	-
A7	-	-	-	-	-	-	-
A8	-	-	-	-	-	-	-

E = AR F = SP

8 cell numbers are displayed

Programmable combinations of fuses on the I/O Architecture Array Cells

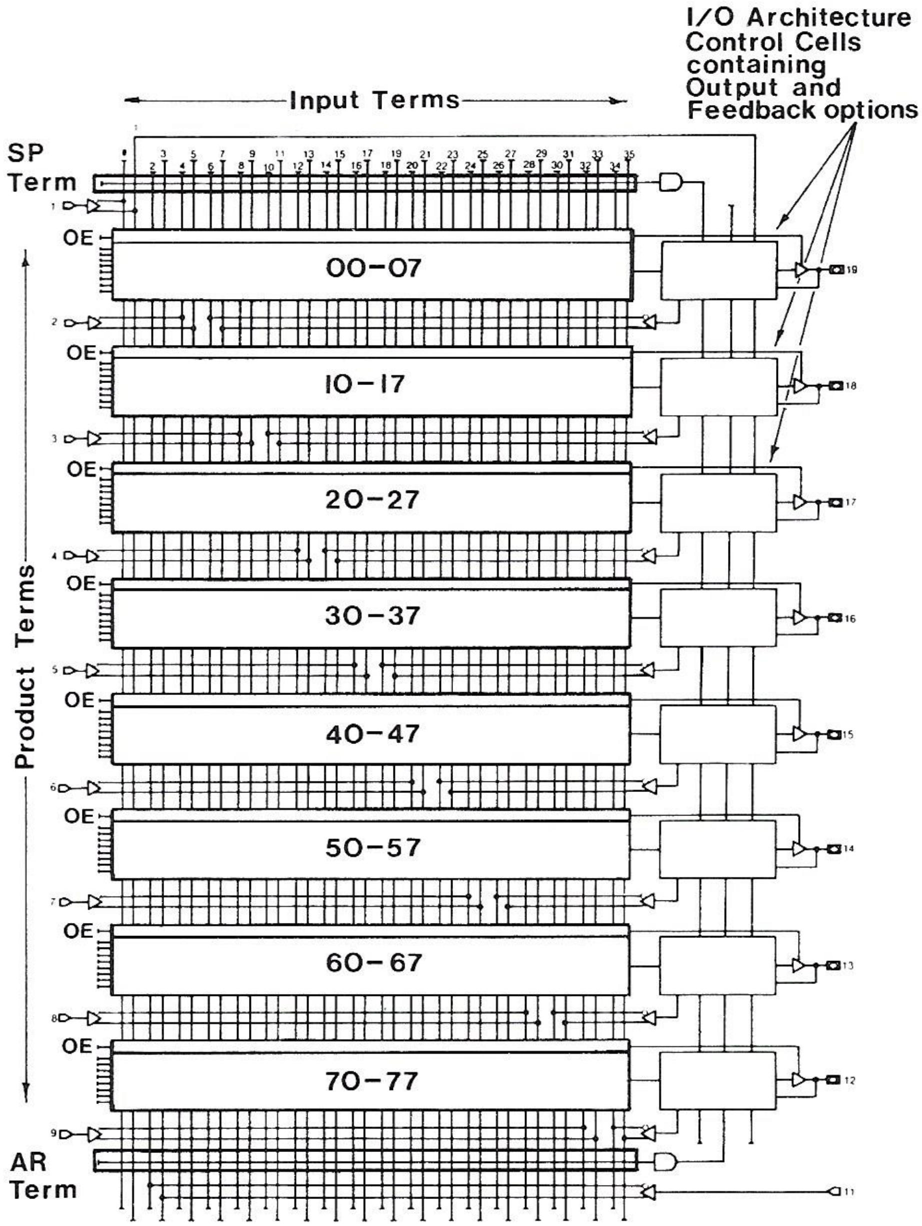
The programmable Output and Feedback operate interactively within the individual Architecture Array Cells to produce one of many possible modes.

O1	O2	O3	O4	F1	F2	F3	OUTPUT	FEEDBACK
X	-	-	-	X	-	-	Combinatorial/High	Combinatorial
X	-	-	-	-	X	-	Combinatorial/High	Registered
X	-	-	-	-	-	X	Combinatorial/High	I/O
-	X	-	-	X	-	-	Combinatorial/Low	Combinatorial
-	X	-	-	-	X	-	Combinatorial/Low	Registered
-	X	-	-	-	-	X	Combinatorial/Low	I/O
-	-	X	-	X	-	-	Registered/High	Combinatorial
-	-	X	-	-	X	-	Registered/High	Registered
-	-	X	-	-	-	X	Registered/High	I/O
-	-	-	X	X	-	-	Registered/Low	Combinatorial
-	-	-	X	-	X	-	Registered/Low	Registered
-	-	-	X	-	-	X	Registered/Low	I/O
n	n	n	n	-	-	-	any of above	none
-	-	-	-	m	m	m	none	any of above

X = Connection Closed
- = Connection Open

An interpretation of the displayed abbreviations and acronyms* used for this device in 'List Device' is shown below on a logic diagram:

Altera EP300



*Identical abbreviations and acronyms are used in List RAM, List Compare and EDIT RAM.

Scanning the IFL devices on the display

The IFL displays are incremented or decremented one product Term at a time. The entire programmable content can be taken as a single unit with every input potentially connected to every output. This is different from the PAL devices where groups of product terms associated with individual outputs are displayed on separate screen pages.

10.1.5 EXAMPLE (e): Signetics 20 pin FPLS 82S159

The 82S159 is one of the most sophisticated devices in PLD technology.

Its programmable architecture consists of:

an AND/OR/Complement array x 32 Product Terms

1 Flip-Flop control term (FC)

10 Control Terms which drive bi-directional I/O lines (D0 to D3), the J-K inputs of all flip-flops (LB and LA) and asynchronous preset and reset lines (PB, RB, PA and RA).

2 Output Enables (EE), 1 polarity term and 8 F/F Mode fuses which select the flip-flop type ''J-K'' or ''D''.

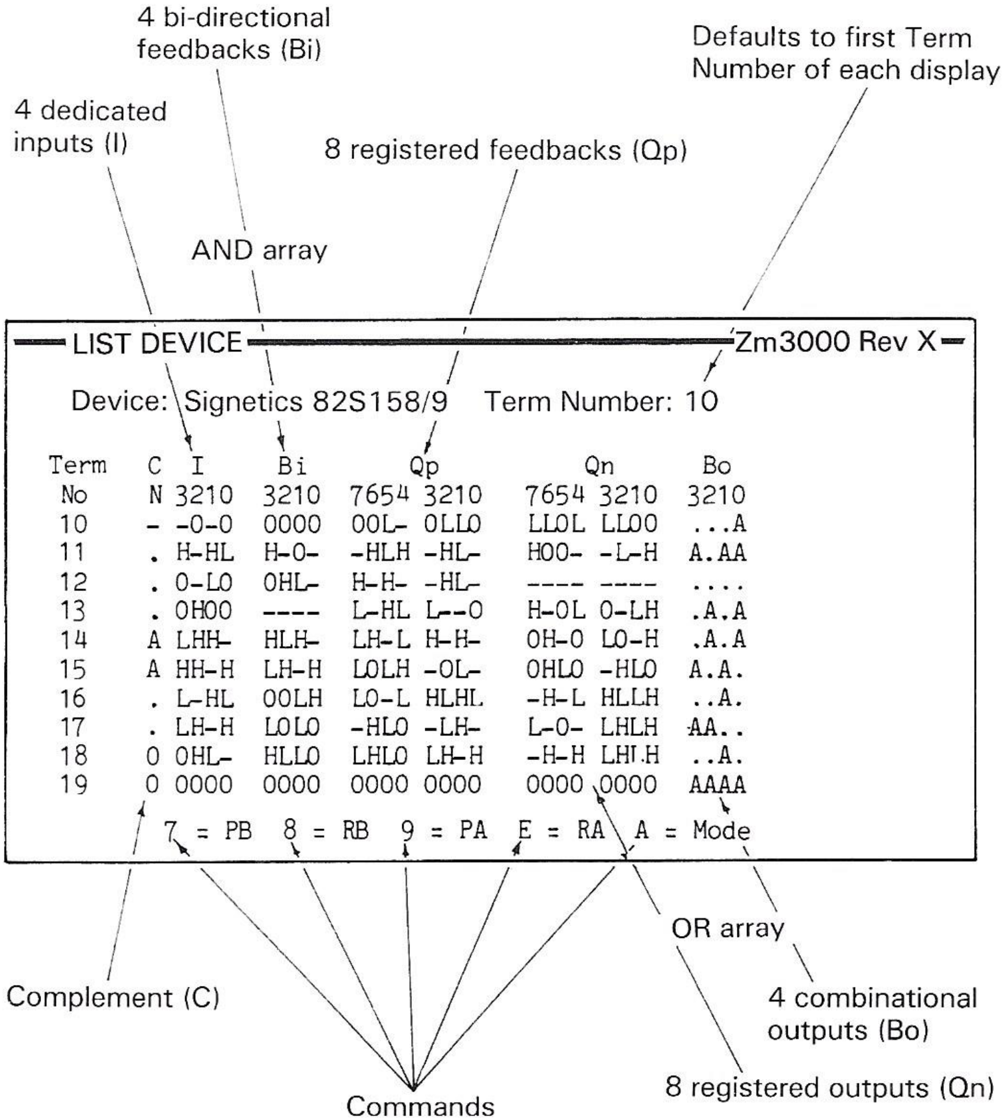
A section of the device can be displayed in 3 ways

(i) by entering a two digit product Term

(ii) by pressing the down cursor key

(iii) by pressing any of the keys: 7, 8, 9, E or A.

A typical AND/OR/Complement array display is shown below:



Complement

O = Inactive
A = Generate
. = Propagate
– = Transparent

''OR'' Array (Qn) D-Type

H = Active (Set)
– = Inactive (Reset)

''AND'' Array (I) (B) (Qp)

O = Inactive
H = True Input (High)
L = Complement Input (Low)
– = Don't Care

''OR'' Array (Qn) J-K Type

O = Toggle
H = Set
L = Reset
– = Hold

A typical display for this devices Flip-Flop Control (FC) is shown below:

```
— LIST DEVICE ——————————————————————————————— Zm3000 Rev X —

  Device: Signetics 82S158/59        Term Number: FC

Term    C  I      Bi       Qp
 No     N 3210   3210   7654 3210
 FC     -  OL-L  -OL-   HOOO HO-H



        7 = PB   8 = RB   9 = PA   E = RA   A = Mode
```

"AND" Array (I), (Bi), (Qp)

O = Inactive
H = True Input (High)
L = Complement Input (Low)
– = Don't Care

Complement (C)

O = Inactive
A = Generate
. = Propagate
– = Transparent

A typical display for this devices Control Terms is shown Below:

```
—— LIST DEVICE ——————————————————————— Zm3000 Rev X ——

    Device: Signetics 82S158/9      Term Number: PB

 Term   C  I      Bi       Qp
  No    N 3210   3210   7654 3210
  PB    - LHLH   HHOO   LHHL LOOH
  RB    - H-OL   HLHL   L--- HLLL
  LB    - OHHL   LLHO   LLLL LLHH
  PA    . LLL-   OO--   LLHL LLLL
  RA    - HLHL   LO--   -HHH HLLL
  LA    - H-LL   LLLL   H-LL LLLL
  D3    - LLLL   HLHL   HH-L LLL-
  D2    . -HLH   LLLH   ---L LLLL
  D1    - LHL-   L--O   -L-L L-LL
  DO    - -OOL   LLHL   OOOL -HHH

         7 = PB   8 = RB   9 = PA   E = RA   A = Mode
```

"AND" array

O = Inactive
H = True Input (High)
L = Complement Input (Low)
- = Don't Care

Complement Array

. = Propagate
- = Transparent

A typical display for this devices Output Enables, Polarity and Flip-Flop Modes is shown below:

```
── LIST DEVICE ──────────────────────────────── Zm3000 Rev X ──

   Device: Signetics 82S158/59        Term Number:

                     F/F Mode    E E    Pol
                     7654 3210   B A    3210
                     .AAA ..AA   - 0    LLLL




      7 = PB   8 = RB   9 = PA   E = RA   A = Mode
```

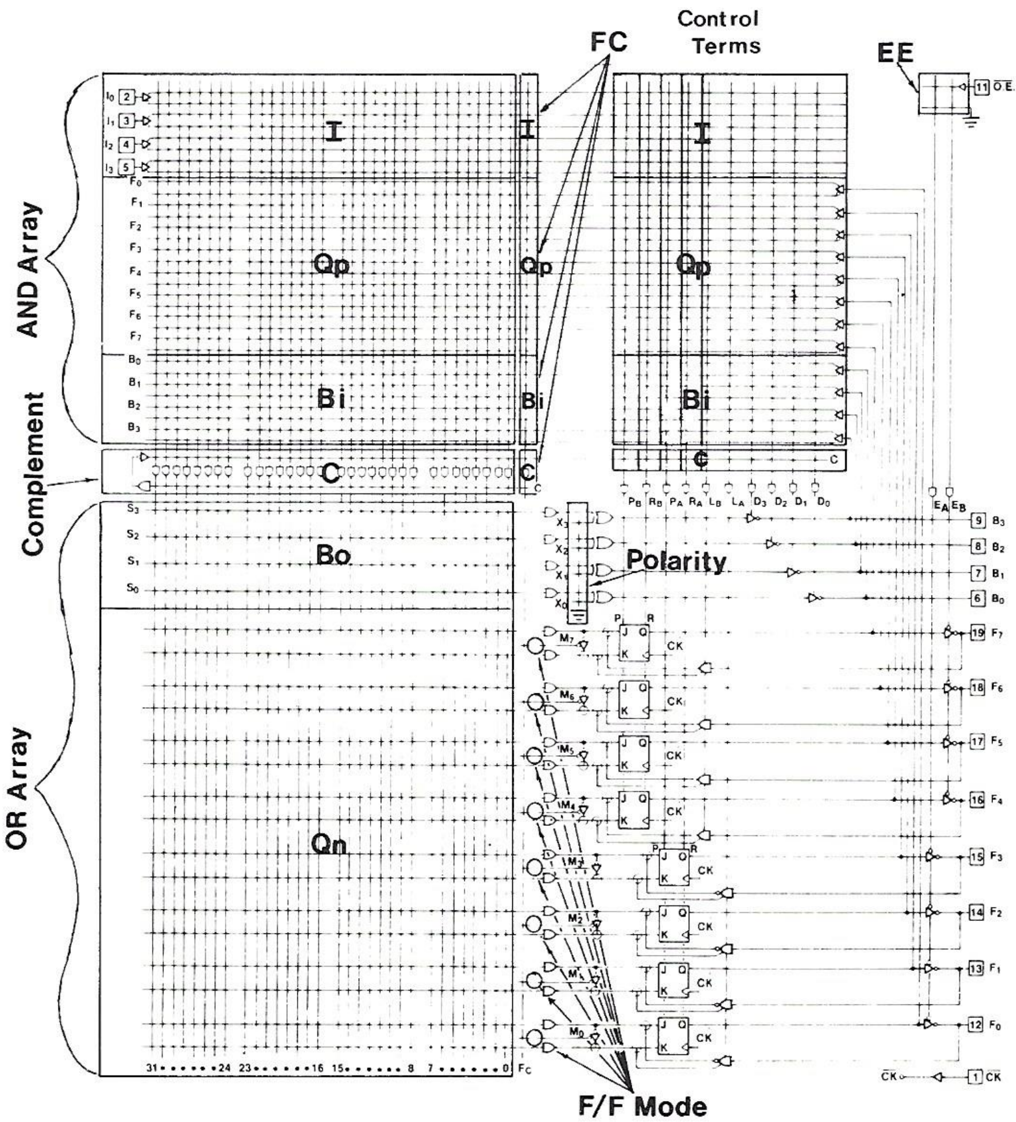
F/F Mode	
A	= Controlled
.	= J-K Type

Output Enable (E E)	
O	= Idle
A	= Control
.	= Enable
—	= Disable

Polarity	
L	= Low
H	= High

An interpretation of the displayed abbreviations and acronyms* used for this device in 'List Device' is shown below on a logic diagram:

82S159



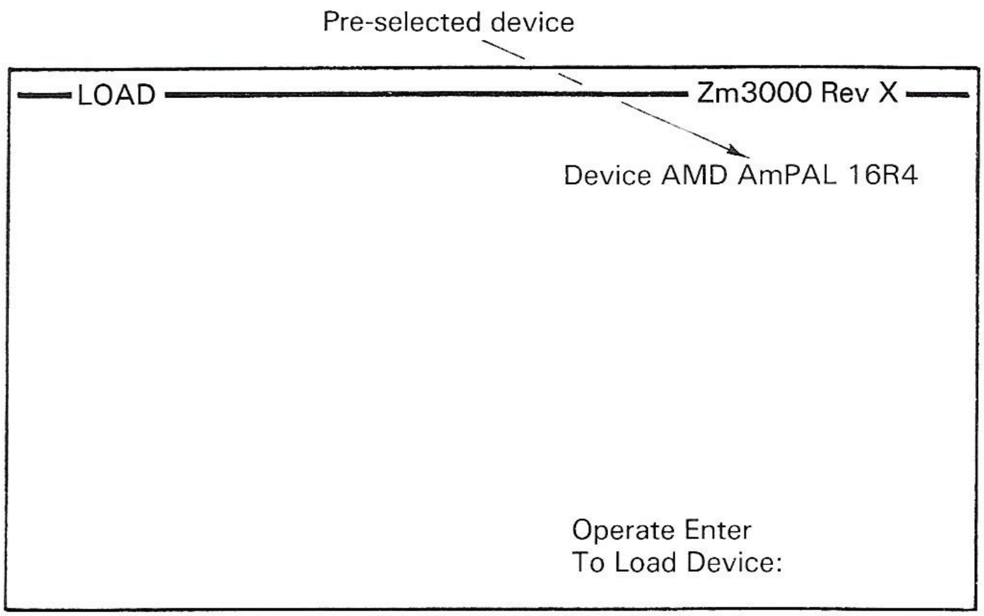
*Identical abbreviations and acronyms are used in List RAM, List Compare and EDIT RAM.

10.2 **Load**

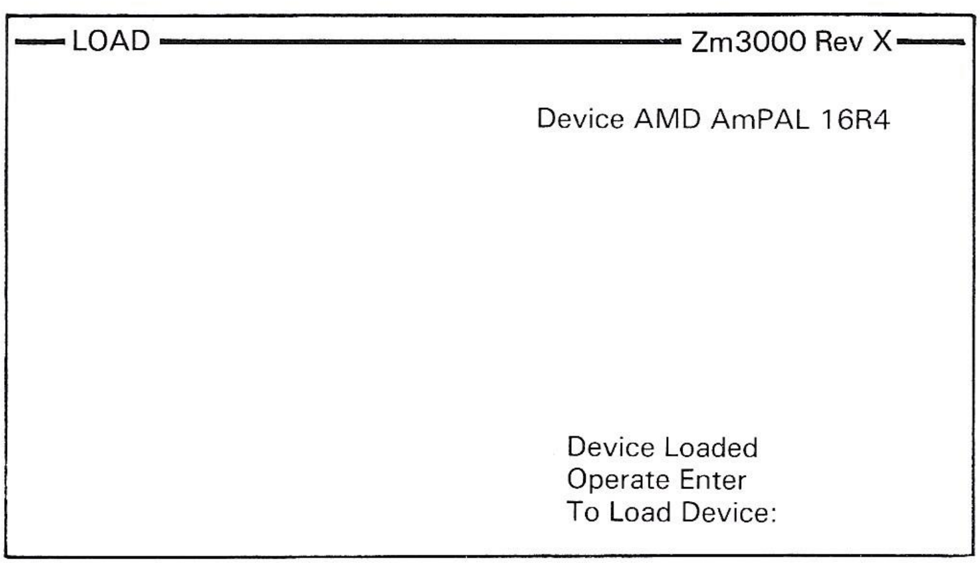
Loading the RAM from a master device

To load a device's fuse pattern into the programmer RAM, insert the device into the appropriate ZIF socket as indicated by the red LED. See Section 1.7—Correct Operation of ZIF Sockets.

Press the LOAD key and the display will show:



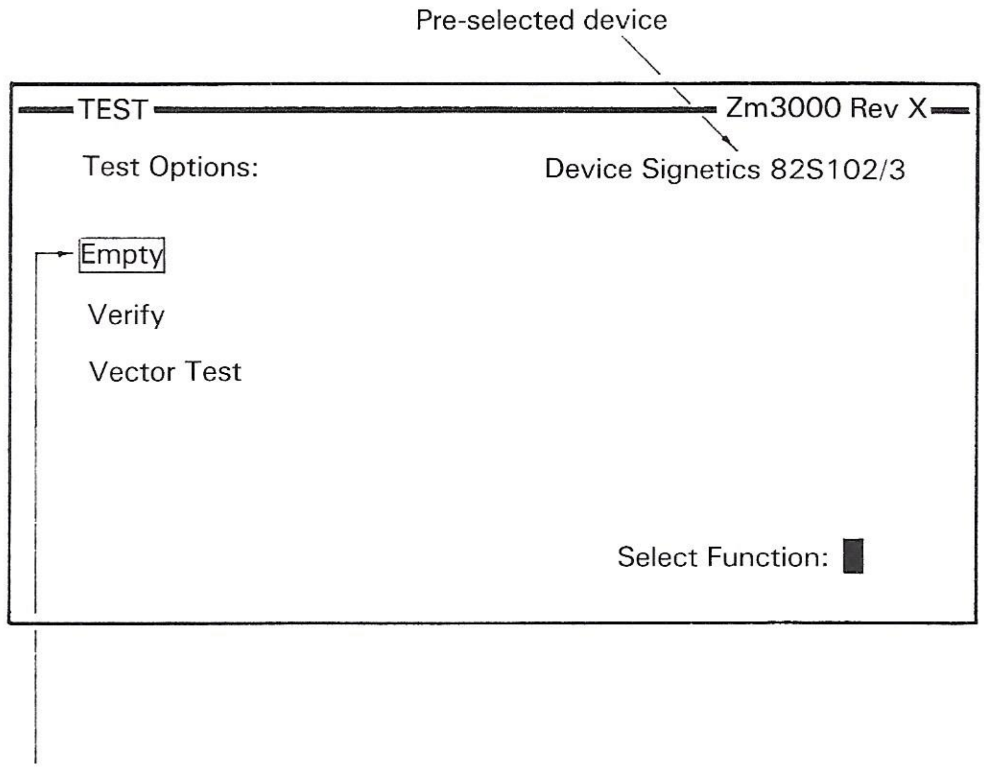
Press the ENTER key and the display will first show ''loading'' in progress and then ''Device Loaded'' indicating that the operation is complete.



Note: Should a PAL's security fuse be blown it will not be possible to load its fuse pattern into the programmer.

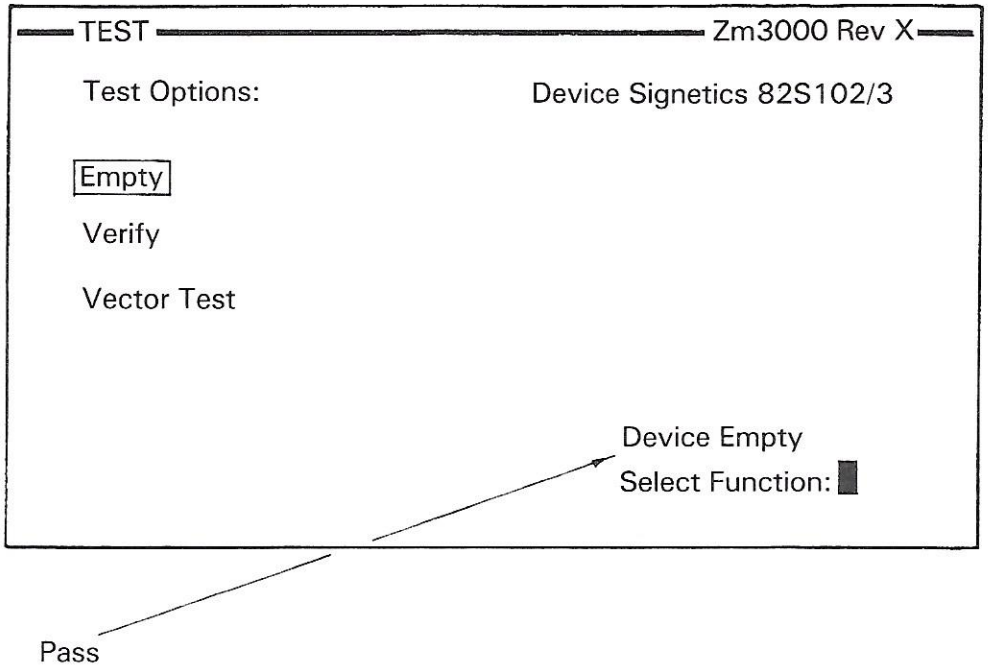
10.3 **Empty Test**

An EMPTY TEST can be applied to the device in the ZIF socket at any time. This can be done by pressing the TEST key to show the TEST menu:

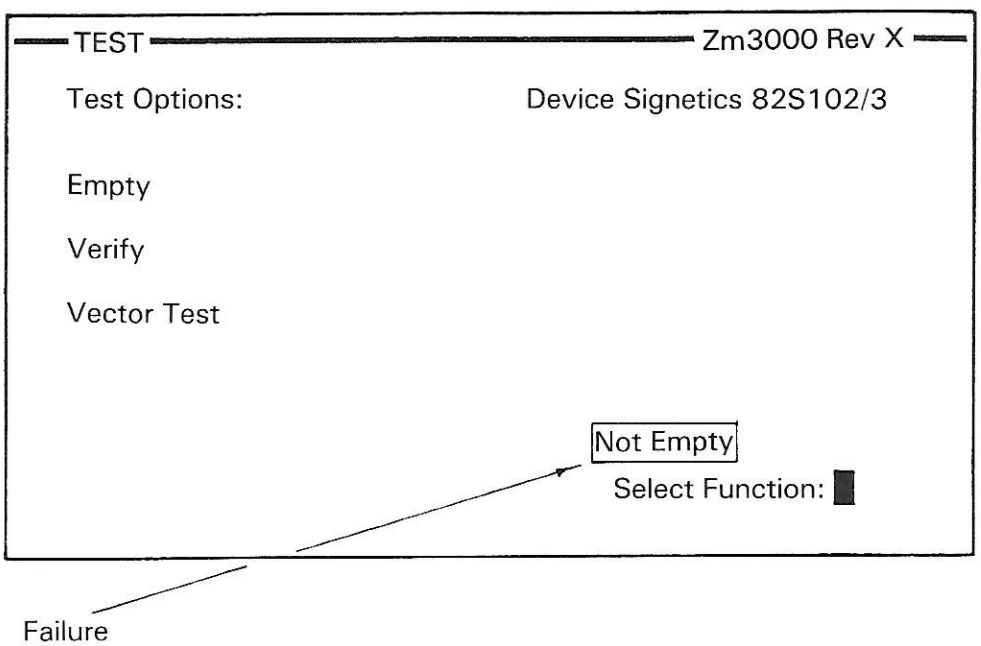


The EMPTY TEST is the default condition of TEST and is indicated by a screen cursor.

To implement the EMPTY TEST press the ENTER key. If the device passes the display will show:



If the device is not empty the display will show:

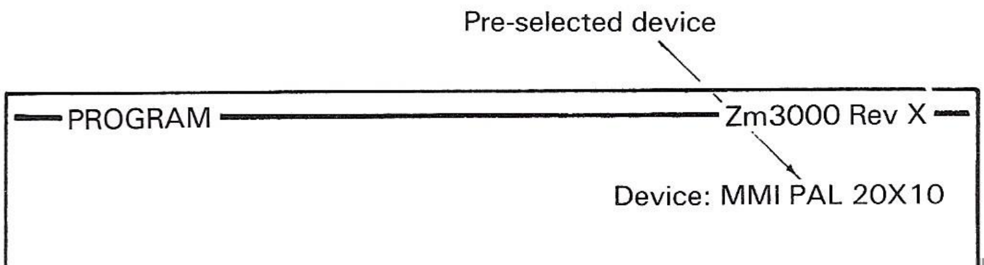


Pressing the RESET key returns to the system menu.

10.4 **Program**

To program a device, place it into the appropriate ZIF socket as indicated by the red LED.

Press the PROG key to display:



Pressing the ENTER key will implement the program sequence. This includes a series of automatic checks on the RAM and the device.

PROGRAM SEQUENCE

Checking: The programmer RAM is checked for valid data.

Illegal Fuse: The System 3000 automatically checks that the pattern already within the device is able to be programmed with the intended data from the RAM.

Programming: Data is copied from the RAM to the device.

Verifying: the device is checked to ensure that its pattern corresponds to the RAM contents. This test may include margin tests.

Security Fuse: After programming of a device the security fuses can be automatically blown causing the fuse pattern to become isolated and cutting off all external access.

If no Program errors occur then the display will show ''Device Programmed'':

```
— PROGRAM ——————————————————————————— Zm3000 Rev X —

                                    Device: MMI PAL 20X10




                                    Device Programmed
                                    Operate Enter
                                    To Program Device:
```

Pressing the RESET will return to the system menu.

List of possible error displays in the program mode (Logic Devices only)

No Blow: A link doesn't program within the device.

Illegal Fuse: A programmed link resident within the device prior to programming is in a location intended for the unprogrammed condition.

Verify Error: The pattern within the device is not identical to the pattern within the RAM after programming.

Connect Error: The device is not correctly inserted into the ZIF socket or the wrong device is being used.

Reverse Device: The device has been inserted into the ZIF socket the wrong way round.

Test Array Fail: Only occurs on IFLs. The manufacturers test pattern has not been completely removed from within the device.

Special cases for EEPALs

Illegal Fuse: Program counter has reached limit.

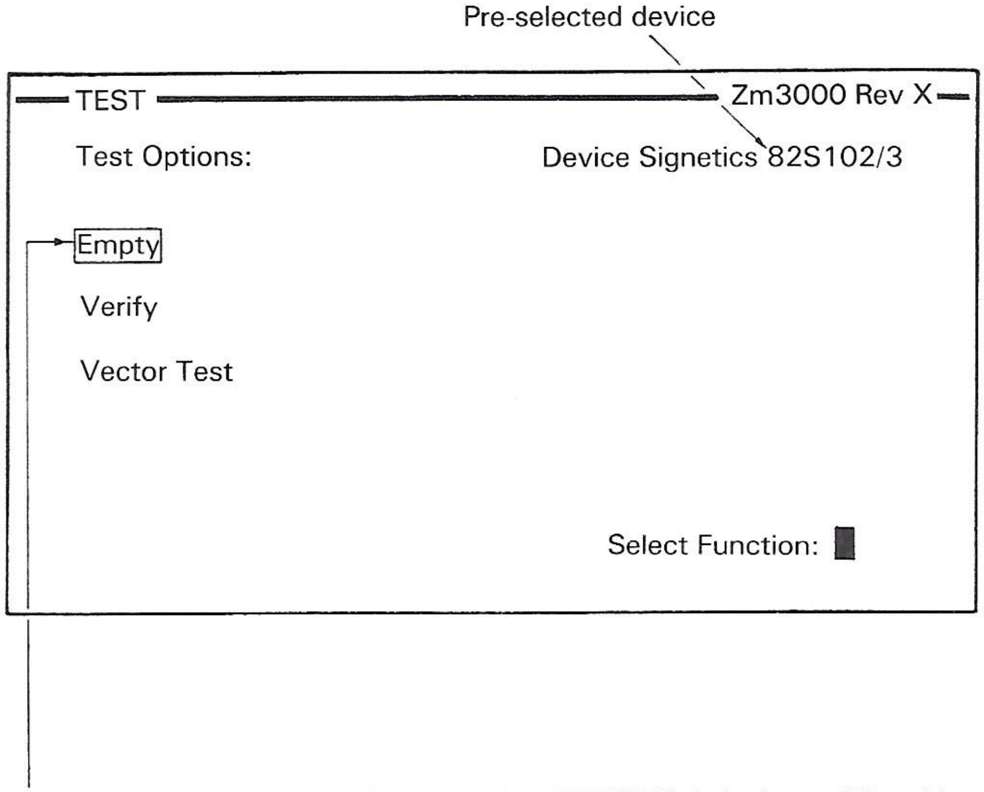
Not Empty: Unable to Erase.

4.5 Verify Test

Note: This is a manual VERIFY TEST and as such can be applied at any time (eg. after LOAD or PROGRAMMING) unlike the in-program VERIFY which is part of an automatic sequence.

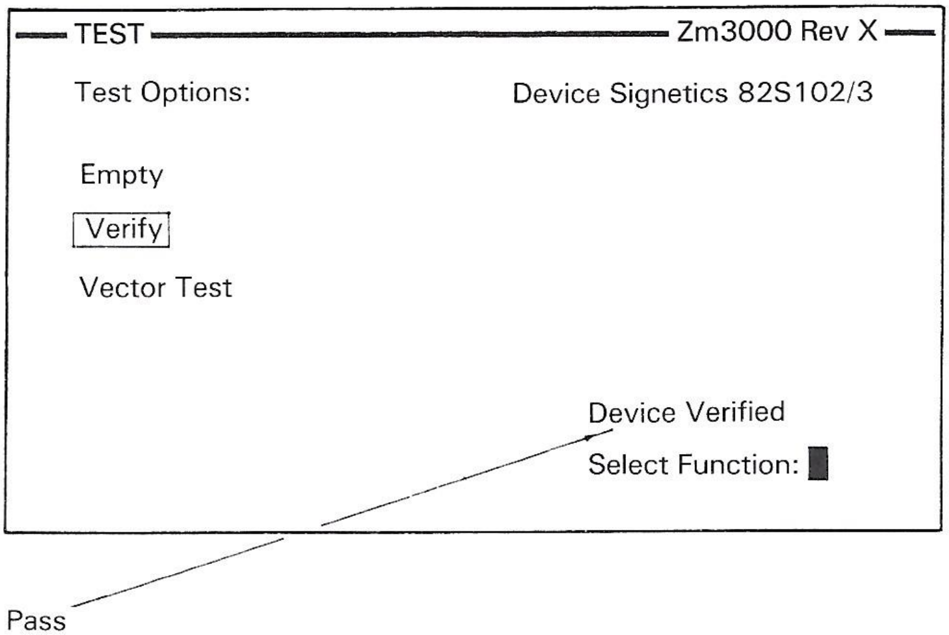
The function of the test is to check that the fuse pattern within a device inserted into its appropriate ZIF socket is identical to the fuse pattern within the programmer RAM. Margin tests may be performed depending on the device.

It can be entered by pressing the TEST key to show the TEST MENU.



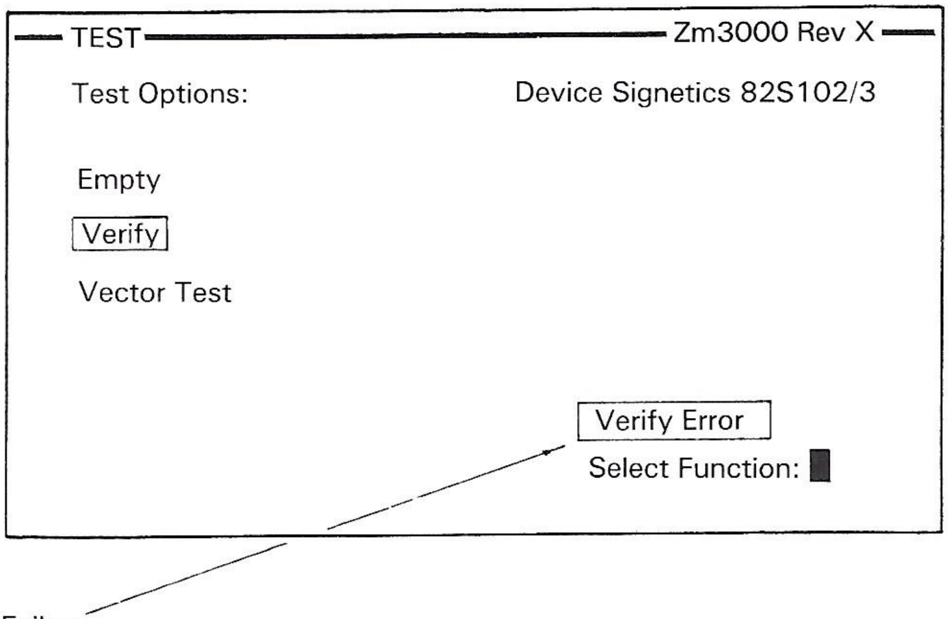
The screen cursor will indicate the "EMPTY TEST" default condition. The VERIFY TEST can be selected by the down cursor key.

To implement the VERIFY press the ENTER key. If the device and RAM pattern are identical the display will show:



Pass

If the device and RAM patterns do not match the display will show:



Failure

The mismatched locations can be found using the List Compare function.

Pressing the RESET key will return to the system menu.

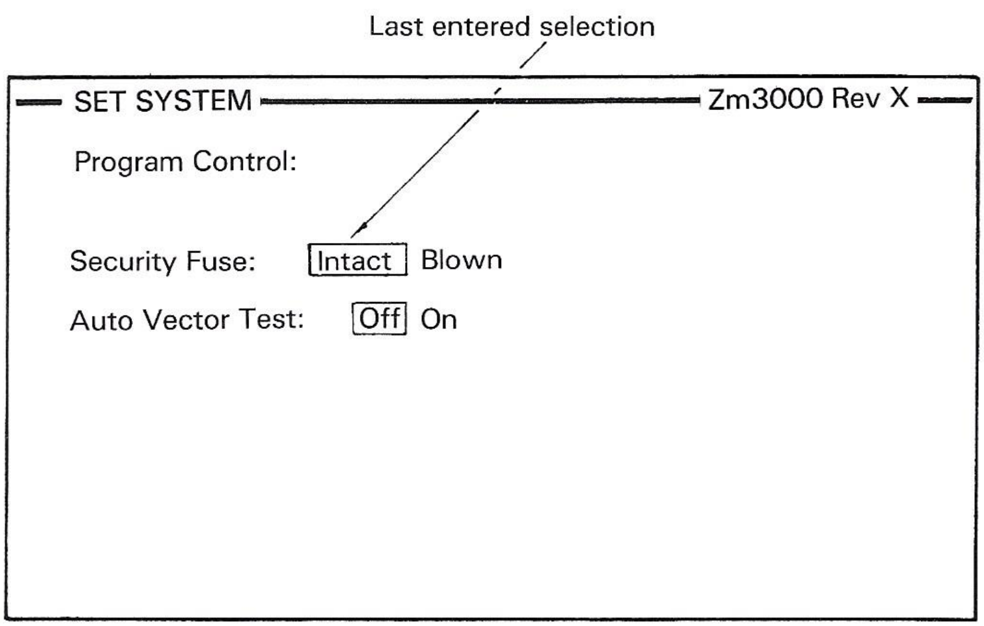
10.6 **Security Fuse State**

Security fuses are contained in some manufacturers devices. After programming of a device the security fuses can be automatically blown causing the fuse pattern to become isolated and cutting off all external access.

Selection

On power-up the Security Fuse feature will be either in its ''blown'' or ''intact'' state.

To check or change the status press key . . . A followed by key . . . 3 to display:



Selection can be made by use of the horizontal cursor keys.

Pressing the ENTER key will implement the selection and return to the SET SYSTEM MENU.

Pressing the RESET key will return to the main SYSTEM MENU.

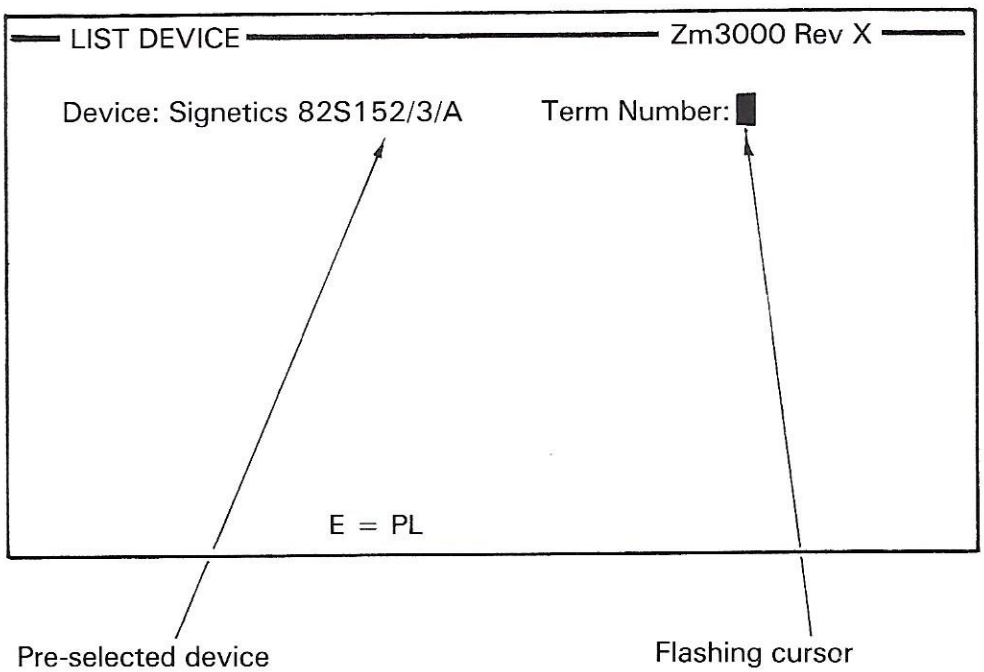
10.7 **List Compare**

List Compare is a visual Verify Test. It enables the pattern of a device in its appropriate ZIF socket to be displayed on the screen. Any disparity between the pattern of the device compared with data within the RAM is indicated in video reverse.

For uniformity all the display headings, acronyms and commands used in List Compare are duplicated in List Device, List RAM and Edit RAM

To initiate the List Compare press key . . . 7

The display will show:



The display shows a flashing cursor indicating that a Term Number should be entered to display a section of fuse array.

A section of this particular device's data can be displayed in three ways

(i) by entering a two digit product Term (00 to 31).
(ii) by pressing the down cursor key
(iii) by pressing the prompt command key . . . E to display the Output Active Levels.

A typical display for a section of this device in List Compare is shown below:

Defaults to first Term Number of each display.

```
── LIST DEVICE ──────────────────────────────────────── Zm3000 Rev X ──

   Device: Signetics 82S152/3A     Term Number: 02

 Term          Im                       Bi                      Bo
  No    7 6 5 4 3 2 1 0    9 8 7 6 5 4 3 2 1 0    9 8 7 6 5 4 3 2 1 0
  02    0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0 0 0    A A A A A A A A A A
  03    0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0 0 0    A A A A A A A A A A
  04    L L L L L - - -    - H - - - - - - - -    . . . A . . . . . .
  05    0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0 0 0    A A A A A A A A A A
  06    0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0 0 0    A A A A A A A A A A
  07    0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0 0 0    A A A A A A A A A A
  08    L L L L H - - -    - H - - - - - - - -    . . . . A . . . . .
  09    0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0 0 0    A A A A A A A A A A
  10    0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0 0 0    A A A A A A A A A A
  11    0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0 0 0    A A A A A A A A A A

                          E = PL
```

SECTION 11

11

stag

Sophisticated systems for the discerning engineer.

11 Ram Functions

11.1 List Ram

This is a feature enabling the data content of the RAM to be scanned on the display. The data is structured in such a way as to copy the pattern within any pre-selected PAL or IFL device. This duplicates display headings, acronyms and commands used in List Device, Edit RAM and List Compare.

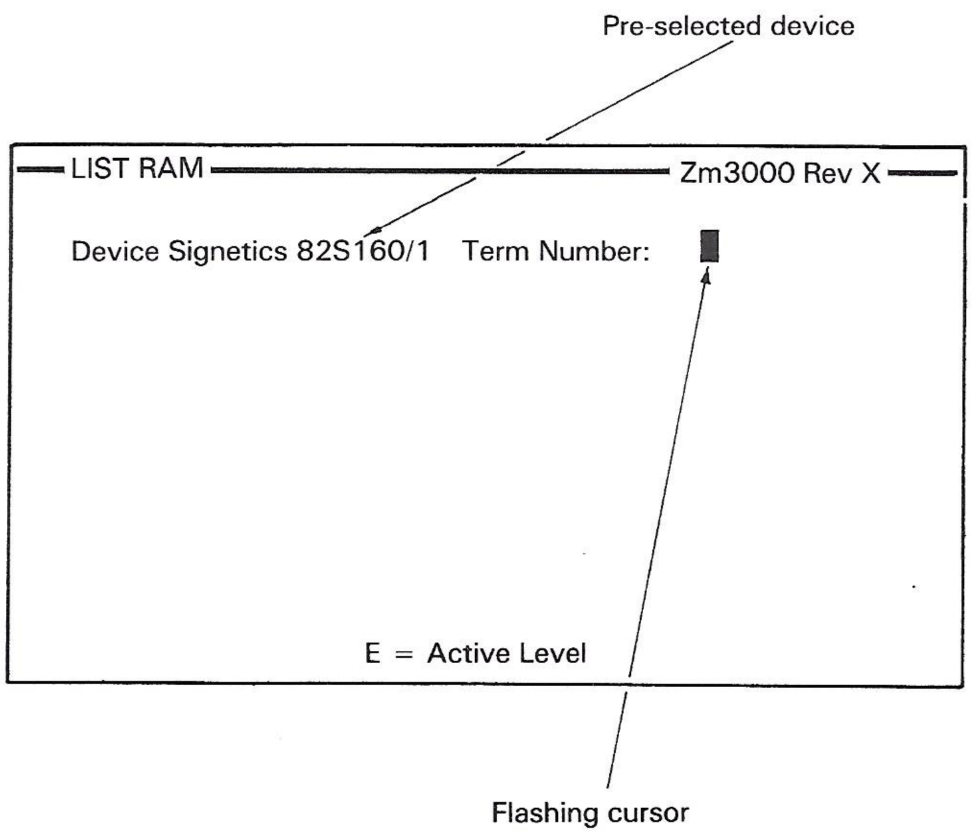
Two device examples will be shown in this section an IFL and a PAL. For a more comprehensive selection see 'List Device' which uses identically structured displays.

(a) an MMI 20 pin PAL 16P8 (Code 20038)

(b) a Signetics 24 pin FPLA 82S160 (Code 15090)

To enter into LIST RAM press key 5.

The display will show:



The display shows a flashing cursor indicating that a Term Number should be entered to display a section of array.

11.1.1 **EXAMPLE (a): MMI 20 pin PAL 16P8**

With 20 pin PAL devices, all Product Terms associated with one particular output (including unused ''phantom fuses'') are displayed together.

MMI's 20 pin PAL 16P8, has 64 Product Terms divided equally between 8 outputs, plus 8 Polarity fuses divided individually between the same 8 outputs.

A section of RAM data can be displayed in three ways:

(i) by entering the required 2 digit Product Term number by use of the keyboard.
(ii) by pressing the down cursor key.
(iii) by pressing the prompt command key . . . E to display the 8 fuse, Polarity Term.

A Product Term display for the RAM when configured for a 16P8 PAL is shown below:

Defaults to the first term line on each display

```
— LIST RAM ——————————————————————————————— Zm3000 Rev X —

  Device MMI PAL 16P8          Term Number: 48

    Term                  Input Lines
     No  0123 4567 8901 2345 6789 0123 4567 8901
     48  --XX XX-- XXXX XX-X --X- XX-X --X- XX-X
     49  -XXX X-XX -XXX X-XX ---X --X- XXX- ----
     50  XXXX --X- XXX- XX-X XX-- X-XX --X- XX--
     51  -XXX ---X XXX- XXXX -X-- XXXX ---- XXX-
     52  XX-- --XX --X- XXX- XXXX ---- X--- X-X-
     53  ---- -XX- ---- -X-- -X-- X--- X--X --XX
     54  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
     55  XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX


                        E = PL
```

8 Product Terms are displayed simultaneously

Prompt command for Polarity Term

X = Connection
– = No Connection

Scanning of the RAM can continue by implementing instructions (i) or (ii) again.

A polarity Term display for the RAM when configured for a 16P8 PAL is shown below:

Press key . . . E

```
—— LIST RAM ————————————————————————— Zm3000 Rev X ——

    Device MMI PAL 16P8              Term Number: PL

              Polarity  0 1 2 3   4 5 6 7

                 PL     - X - X   X - - X




                          E = PL
```

X = Connection
– = No Connection

11.1.2 EXAMPLE (b): Signetics 24 pin FPLA 82S160

The 82S160 utilizes the standard AND/OR/Invert architecture to directly implement custom sum of product logic equations.

The programmable architecture of this device consists of:

An AND array consisting of 12 dedicated Inputs and 48 Product Terms.

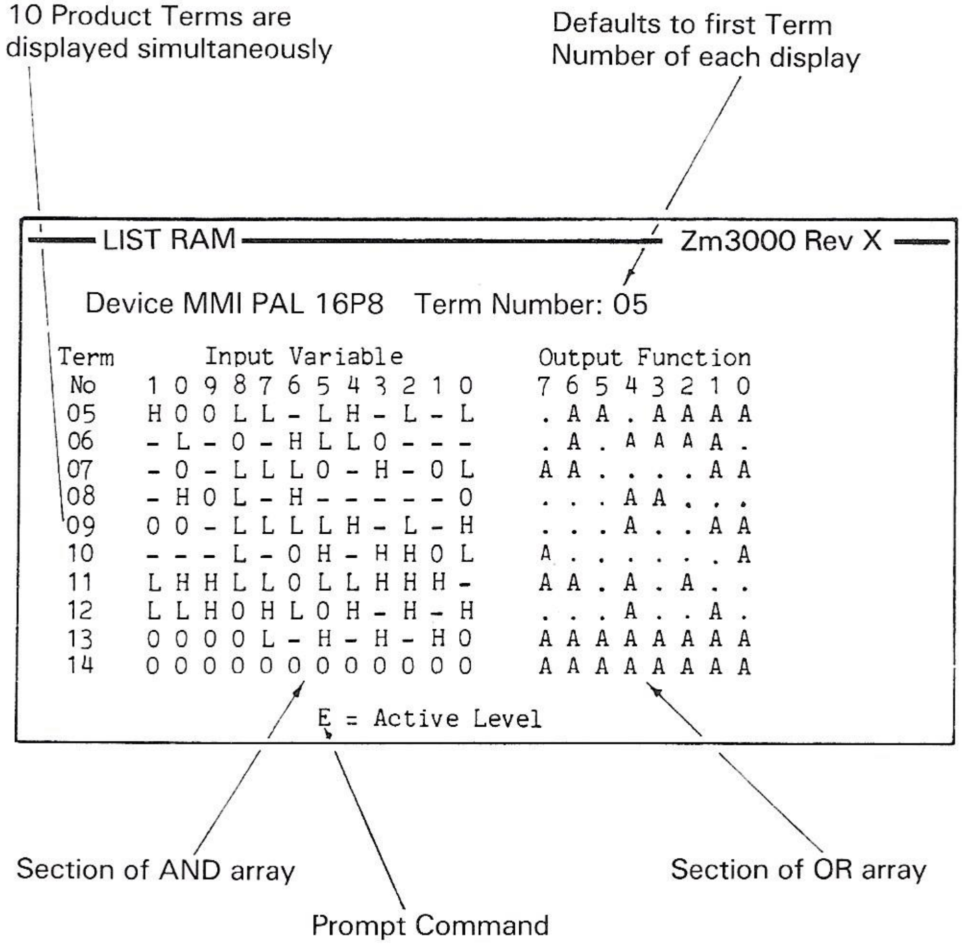
An OR array generating 8 sum of product terms.

An Output Active Level (Polarity) where each output can be programmed as active High or Low.

A section of RAM data can be displayed in three ways:

(i) by entering a two digit Product Term number (00 to 47) using the keyboard.
(ii) by pressing the down cursor key.
(iii) by pressing the prompt command key . . . E to display the Output Active Level.

An AND/OR array display for the RAM when configured for an 82S160 FPLA is shown below:




```
—— LIST RAM ——————————————————— Zm3000 Rev X ——

  Device MMI PAL 16P8   Term Number: 05

Term        Input Variable          Output Function
 No    1 0 9 8 7 6 5 4 3 2 1 0     7 6 5 4 3 2 1 0
 05    H 0 0 L L - L H - L - L     . A A . A A A A
 06    - L - 0 - H L L 0 - - -     . A . A A A A .
 07    - 0 - L L L 0 - H - 0 L     A A . . . . A A
 08    - H 0 L - H - - - - - 0     . . . A A . . .
 09    0 0 - L L L L H - L - H     . . . A . . A A
 10    - - - L - 0 H - H H 0 L     A . . . . . . A
 11    L H H L L 0 L L H H H -     A A . A . A . .
 12    L L H 0 H L 0 H - H - H     . . . A . . A .
 13    0 0 0 0 L - H - H - H 0     A A A A A A A A
 14    0 0 0 0 0 0 0 0 0 0 0 0     A A A A A A A A

                  E = Active Level
```

Scanning of the device can continue by implementing instructions (i) or (ii) again.

Dedicated AND array
O = Inactive
H = True Input (High)
L = Complemented Input (Low)
- = Don't Care

An Active Level (Polarity) display for the RAM when configured for an 82S160 FPLA is shown below:

Press key . . . E

```
—— LIST RAM ——————————————————————————— Zm3000 Rev X ——

   Device Signetics 82S106/1        Term Number: AL


              Active Level  7 6 5 4 3 2 1 0
                    AL      L L H H H H L L






                        E = Active Level
```

Active Level (Output Polarity)
H = High
L = Low

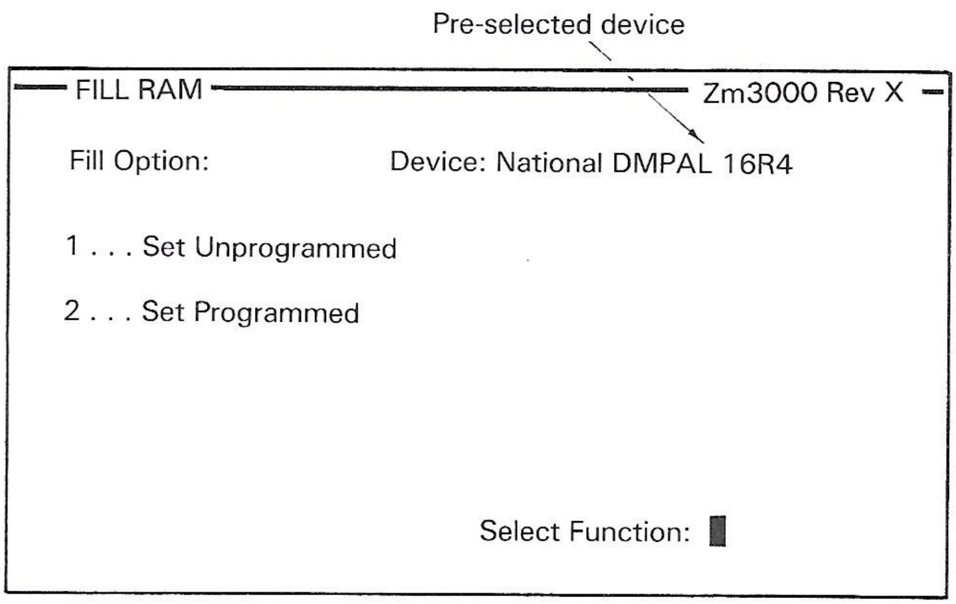
11.2 Fill RAM

This feature enables the RAM to be filled with either the programmed or the unprogrammed state of a PLD.

This is useful for setting the RAM to a known state prior to Editing.

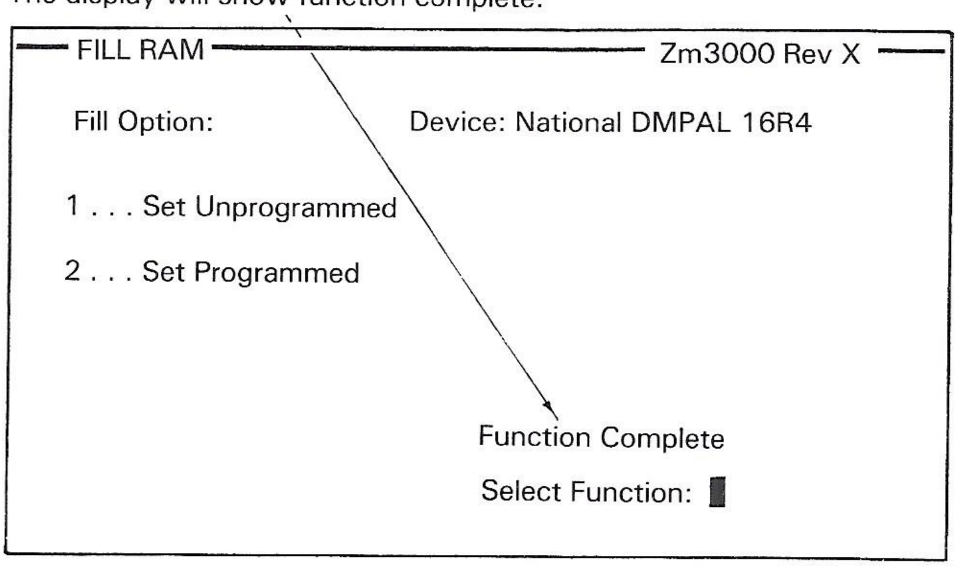
The FILL RAM menu may be entered from the system menu by pressing key . . . 4

The display will show:



Press key 1 or 2 for the required fill

The display will show function complete:



Pressing the RESET key will return to the system menu.

11.3 **Edit RAM**

This feature allows the actual content of the RAM to be directly modified by use of the keyboard.

The displayed data is structured in such a way as to copy the matrix (a single Term at a time) within any pre-selected PLD. (See List Device section).

For uniformity all the display headings, acronyms and prompt commands used in Edit RAM are duplicated in the List Device, List RAM and List Compare.

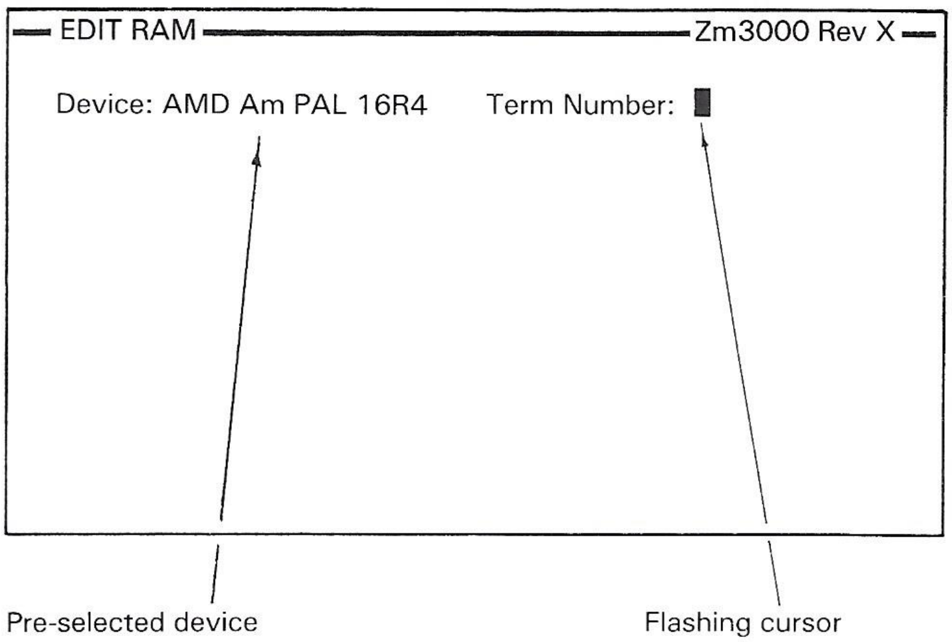
Two device examples are shown in this section an IFL using the - HLA · logic notation and a PAL using the X - logic notation.

Device examples:

(a) a Texas 20 pin PLA 16R4 (Code 40032)
(b) a Signetics 20 pin FPLS 82S159 (Code 14008)

To EDIT the RAM press key . . . 3, followed by key . . . 1

The display will show:



The display shows a flashing cursor indicating that a two digit number should be entered to display a single Term.

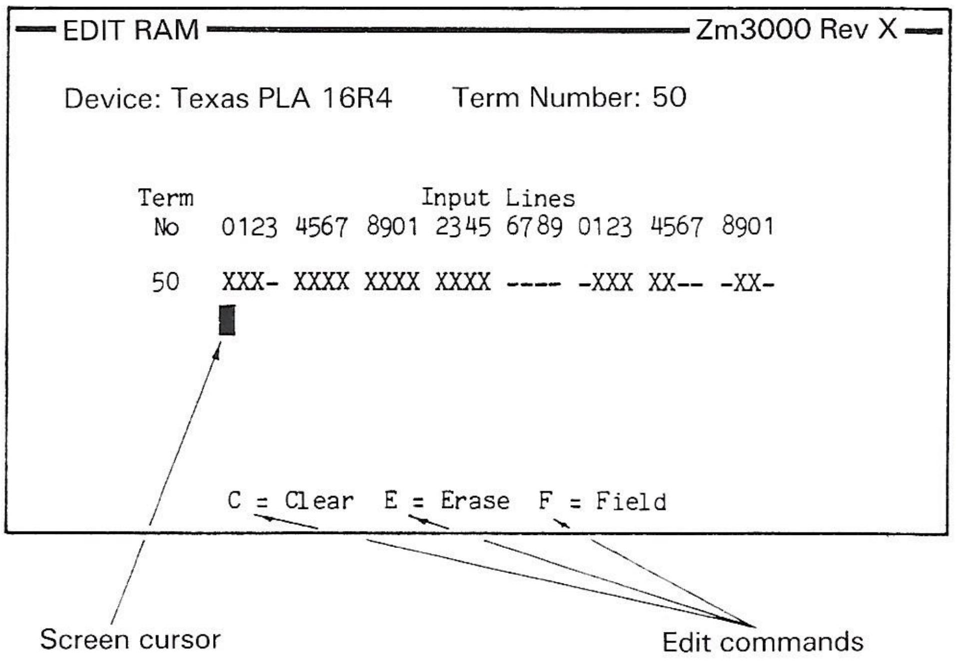
11.3.1 **EXAMPLE (a): A Texas 20 pin PLA 16R4**

A section of RAM can be displayed in 2 ways:

(i) by entering the required two digit Term number by use of the keyboard.
(ii) by pressing the down cursor key.

Note: For ease of use in the Edit RAM mode, data is shown only one Term at a time. A screen cursor controlled from the keyboard can be made to move horizontally below the displayed Term to pinpoint an individual fuse.

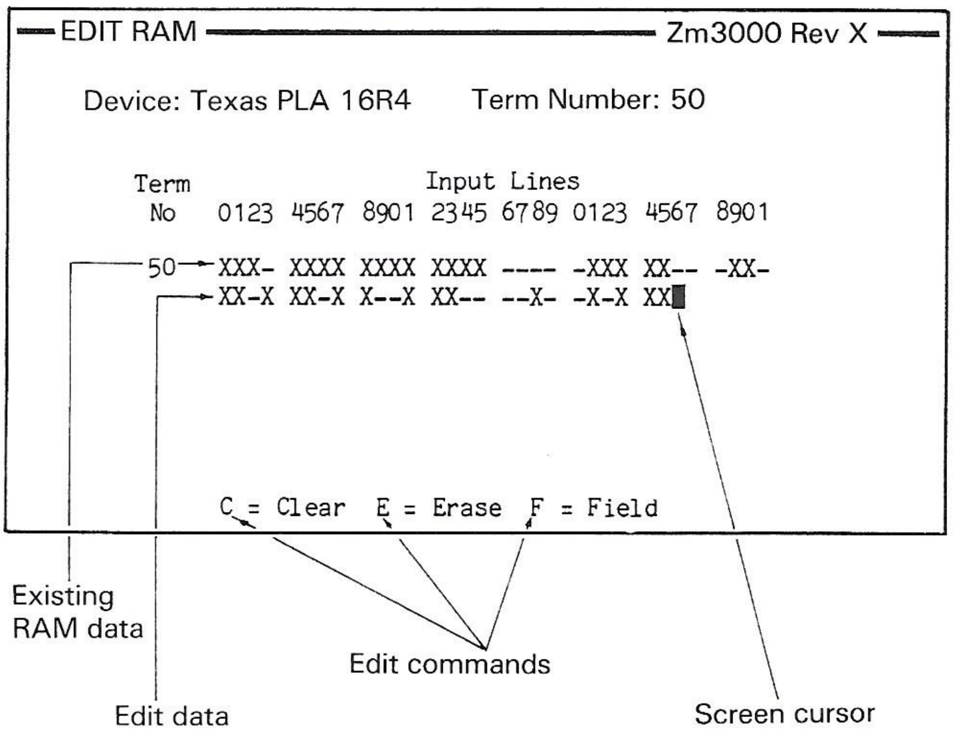
A single Product Term display for the RAM when configured for a 16R4 PAL is shown below:



Edit data can be displayed below the selected Term by use of the **X** - notation keys or duplicated by use of the right cursor key.

This makes possible a visual comparison between existing RAM data and new data prior to entry.

For example:



The left cursor key will delete any new data prior to entry.

The ''Edit help commands'' speed up data entry and deletion:

C = Clear

Pressing key . . . C will clear the Edit Term from the display and will default the existing Term to its unprogrammed state: X.

E = Erase

Pressing key . . . E will erase the Edit Term but will not corrupt the existing RAM data.

F = Field

Pressing key . . . F will speed up data entry by duplicating 8 fuse locations of existing data at a time.

Pressing the ENTER key will implement the Edit

For example:

```
— EDIT RAM ———————————————————————— Zm3000 Rev X —

  Device: Texas PLA 16R4          Term Number: 50

        Term
         No   0123 4567 8901 2345 6789 0123 4567 8901

   ┌──── 50 → XX-X XX-X X--X XX-- --X- -X-X XX-- -XX-
   │
   │
   │        C = Clear  E = Erase  F = Field
   │
```

New RAM Term

More Terms can be individually displayed for Editing by pressing either of the vertical cursor keys or by directly entering a two digit Term number.

Pressing the RESET key will return to the system menu.

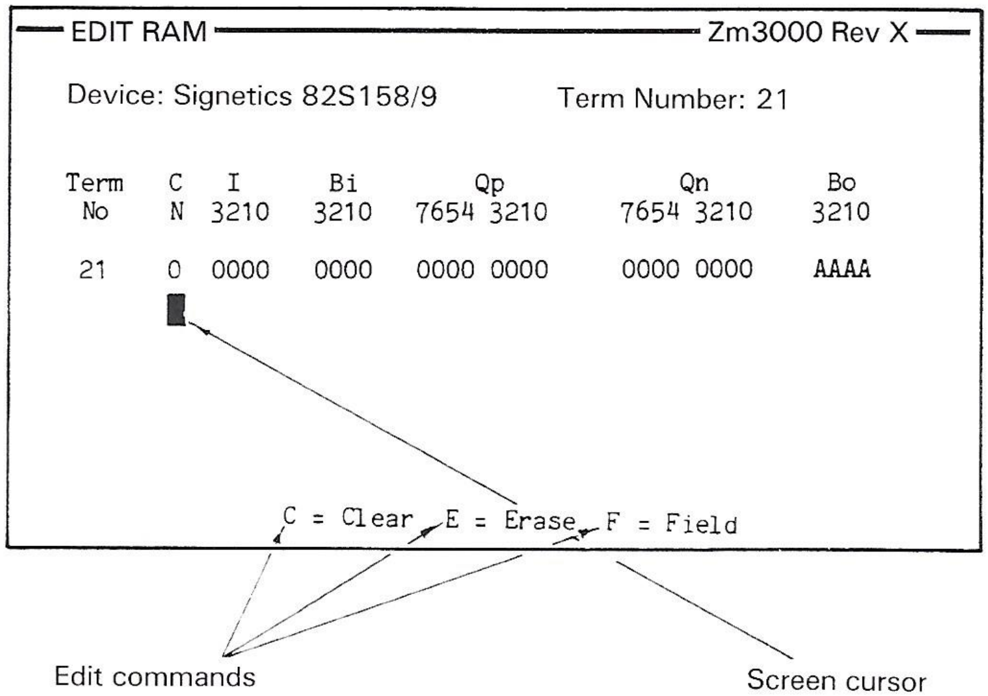
11.3.2 **EXAMPLE (b): A Signetics 20 pin FPLS 82S159**

A section of RAM can be displayed in 3 ways:

(i) by entering the required two digit Term number by use of the keyboard.
(ii) by pressing the down cursor key.
(iii) by pressing any of the prompt command keys: 7, 8, 9, E or A.

Note: For ease of use in the Edit RAM mode, data is shown only one Term at a time. A screen cursor controlled from the keyboard can be made to move horizontally, below the displayed Term to pinpoint an individual fuse.

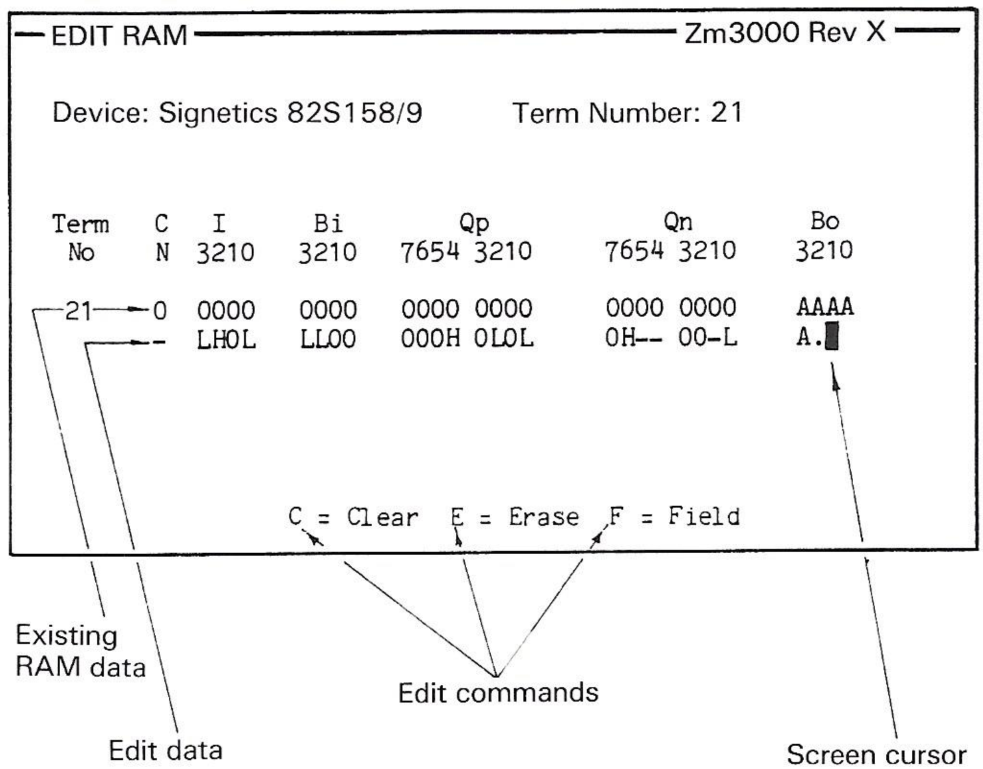
A single Product Term display for the RAM when configured for an 82S159 IFL is shown below:



Edit data can be displayed below the selected term by use of the **- XLA ·** notation keys or duplicated by use of the right cursor key.

This makes possible a visual comparison between existing RAM and new data prior to entry.

For example:



The left cursor key will delete any new data prior to entry.

The "Edit help commands" speed up data entry and deletion

C = Clear

Pressing key . . . C will clear the Edit Term from the display and will default the existing Term to its unprogrammed state: 0 and A.

E = Erase

Pressing key . . . E will erase the Edit Term but will not corrupt the existing RAM data.

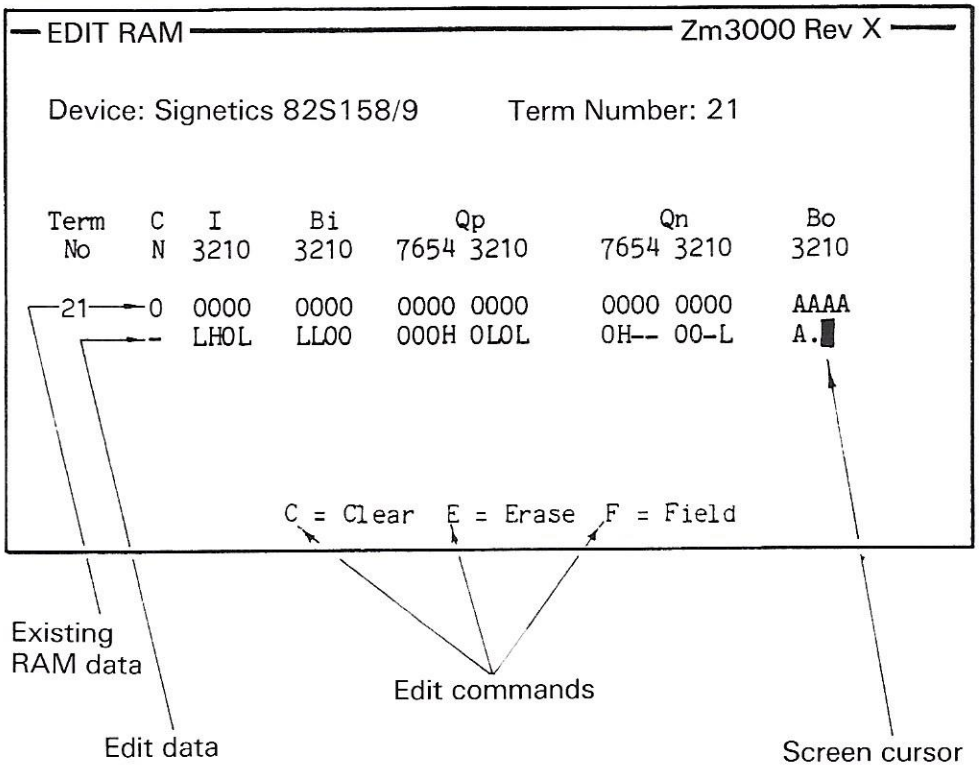
F = Field

Pressing key . . . F will speed up data entry by duplicating whole sections of an existing Term ie. C, I, Bi, Qp, Qn, Bo.

Edit data can be displayed below the selected term by use of the **- XLA ·** notation keys or duplicated by use of the right cursor key.

This makes possible a visual comparison between existing RAM and new data prior to entry.

For example:



The left cursor key will delete any new data prior to entry.

The "Edit help commands" speed up data entry and deletion

C = Clear

Pressing key . . . C will clear the Edit Term from the display and will default the existing Term to its unprogrammed state: 0 and A.

E = Erase

Pressing key . . . E will erase the Edit Term but will not corrupt the existing RAM data.

F = Field

Pressing key . . . F will speed up data entry by duplicating whole sections of an existing Term ie. C, I, Bi, Qp, Qn, Bo.

Pressing the ENTER key will implement the Edit

For example:

```
— EDIT RAM ———————————————————————— Zm3000 Rev X ——

  Device: Signetics 82S158/9      Term Number: 21

  Term  C   I     Bi       Qp          Qn        Bo
   No   N  3210  3210  7654 3210   7654 3210   3210

  —21——- -  LH0L  LL00  000H 0L0L   0H-- 00-L   A...



              C = Clear  E = Erase  F = Field
```

New RAM Term

More Terms can be displayed for Editing by pressing either of the vertical cursor keys, by directly entering a two digit Term number or by pressing one of the command keys 7, 8, 9, E or A.

Pressing the RESET key will return to the system menu.

11.4 Checksum

Checksum calculates the JEDEC link checksum of the RAM content and can be entered at any time from the System Menu.

To initiate the function press key . . . 8 and the checksum will be displayed on the System menu.

For example:

```
— SYSTEM MENU ———————————————————————————— Zm3000 Rev X —
  Function Codes:

  0 . . . Set Family            Device:    Signetics 82S152/3/A
  1 . . . Set Device
                                LOAD      . . . Load Device
  3 . . . Edit RAM              I/O       . . . Input/Output
  4 . . . Fill RAM              TEST      . . . Test Device
                                PROG      . . . Program Device
  5 . . . List RAM
  6 . . . List Device
  7 . . . List Compare

  8 . . . Checksum              Checksum: 45C9
  9 . . . Link Count
  A . . . Set System            Select Function:
```

Checksum

Pressing the RESET key will return to the standard system menu.

11.5 **Link Count**

Link Count shows the number of high impedence connections present in the RAM pattern and can be entered at any time from the System Menu.

To implement the function press key . . . 9 and the link count will be displayed on the System Menu.

For example:

```
—SYSTEM MENU ———————————————————————— Zm3000 Rev X—

 Function Codes:

 0 . . . Set Family          Device:     Signetics 82S152/3/A
 1 . . . Set Device
                             LOAD    . . . Load Device
 3 . . . Edit RAM            I/O     . . . Input/Output
 4 . . . Fill RAM            TEST    . . . Test Device
                             PROG    . . . Program Device
 5 . . . List RAM
 6 . . . List Device
 7 . . . List Compare

 8 . . . Checksum            Link Count: 00564
 9 . . . Link Count
 A . . . Set System          Select Function:
```

Link Count

Pressing the RESET key will return to the standard system menu.

SECTION 12

stag

Sophisticated systems for the discerning engineer.

12

12 Vector Testing

12.1 An Introduction to Vector Testing

Why Test Programmable Logic?

The fuses in PLDs are programmed and read by the application of supervoltages (voltages greater than TTL levels) to the device pins. Also, because of the way the fuses are addressed and read, not all of the circuitry is used during verification. For these reasons, it is possible to have a PLD which contains the desired fuse pattern but which does not function correctly at operational voltages. It is therefore desirable to perform logic tests on a device to ensure its correct operation.

What is Vector Testing?

Vector testing is a method of testing the logic function of a device. A sequence of test vectors is applied to a device and its pins monitored for the expected states. A test vector defines voltages to be applied to device pins and the expected conditions of other pins. A typical test vector for a 24-pin device is shown below:

0001 C 1 0 1 1 1 1 0 0 1 1 N H L H L Z Z H H L H L N

0001 is the test vector number. The rest of the vector describes input and output conditions for the test, each character representing a device pin. The first character is for pin 1, the second for pin 2, etc.

The characters are interpreted as follows:

0 - drive pin low
1 - drive pin high
C - drive pin low, high, low
K - drive pin high, low, high
N - power pin
L - test pin low
H - test pin high
Z - test pin for high impedance
X - don't care
F - float pin
P - pre-load

In the example shown, the test vector is for a 24-pin PAL* and so pins 12 and 24 have N characters as these are the power pins.

Test vectors are applied in numerical sequence and steady levels are applied before the C and K clock tests.

Why Vector Test?

Most PALs* have pins which may be inputs or outputs depending on the fuse pattern programmed into the device. It is impossible for a standard automatic test procedure to cope with this as a pin's function is dependent on the state of other pins. This can result in inconsistent results when logic verification is attempted. With registered PALs* the sequence in which inputs are applied becomes important as the internal registers must be tested for correct sequencing. This is impossible to test automatically (e.g. by applying pseudo-random sequences to the device) since, for instance, if a RESET line is low then the registers will not be sequenced.

It is therefore essential to test a logic device in a manner tailored to its function. The best way of achieving this is to apply test vectors which are generated with consideration of the logical function of the device.

Vector testing can reduce testing time and properly test all types of PLD. It should be noted that vector testing is not confined to programmable logic devices but is a technique which can be applied to all logic devices.

Test vectors can be loaded into the programmer's RAM via one of the serial interface ports in JEDEC format or via the keyboard. Test vector numbers must be within the range 0000 to 9999 (decimal).

12.2 **Edit Vectors**

Keyboard Entry of Test Vectors

Press key 3 followed by key 2. This will display the 'Edit Vectors' Sub-menu which provides 4 options:

L - List Vectors

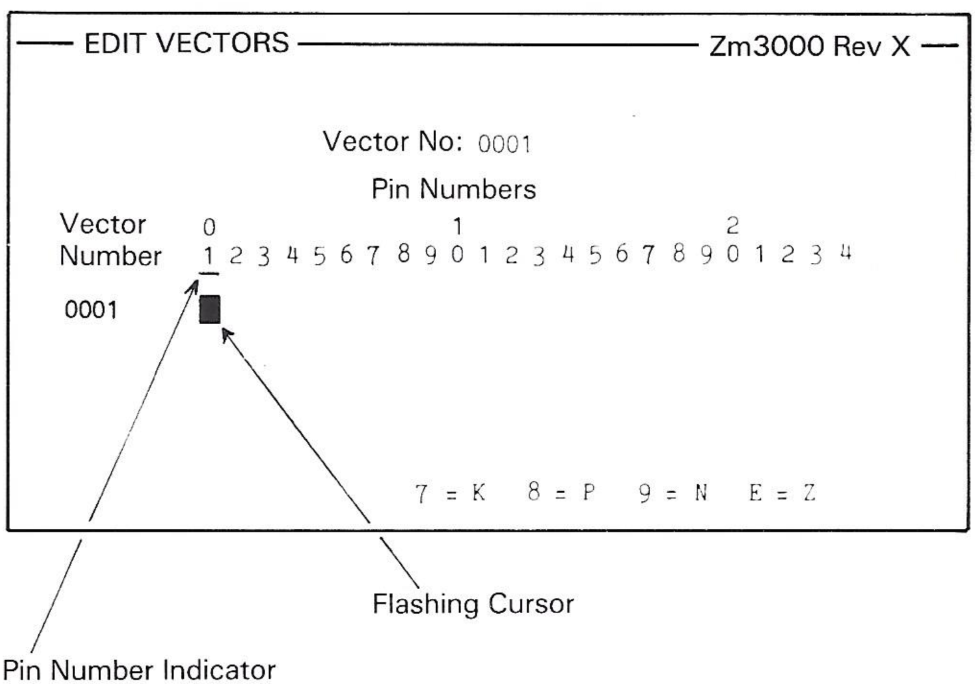
A - Add Vectors

D - Delete Vectors

X - Modify Vectors

Press the relevant key to select any of these. If no vectors have been input, selecting any option other than 'Add Vectors' will result in the message 'No Vectors' being displayed. In this event:
Press key A. The prompt, 'Vector No': will appear. Vector numbers can be in the range 0000 to 9999 (decimal). Enter a four digit vector number, for example 0001.

The display will show:



Select the required Vector characters using the keyboard.

0, 1, C, K, N, L, H, Z, X, F and P are available (for K, N, Z and P use keys 7, 8, 9 and E).

For example:

```
— EDIT VECTORS ——————————————————————— Zm3000 Rev X —

                    Vector No: 0001 █

                      Pin Numbers
  Vector    0                 1                   2
  Number    1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4

  0001      C 1 0 1 1 1 1 0 0 1 1 N H L H L Z Z H H L H L N




                          7 = K   8 = P   9 = N   E = Z
```

Pressing the left cursor key will delete individual characters.

When all characters have been entered the screen cursor will skip back to its original position.

To enter another Vector press either the down cursor key or select another four digit number.

When the required number of Vectors have been entered, press the ENTER key to return to the Vector sub-menu. (Pressing RESET will return to the system menu).

Vector Renumbering

If a series of vectors has already been entered, 0000, 0001, 0002, 0003, 0004, 0008 and 0009 for instance, and a new Vector 0002 is added, the existing vectors 0002, 0003 and 0004 will be renumbered as 0003, 0004 and 0005. The vectors 0008 and 0009 will remain unchanged.

List Vector

Press key L as prompted by the Vector sub-menu to display:

```
— EDIT VECTORS ——————————————————————— Zm3000 Rev X —

                    Vector No: 0000 █

                  Pin Numbers
 Vector   0                  1                   2
 Number   1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4



 ----------------------------------------------------------

 ----------------------------------------------------------
```

Press the down arrow key or enter a pre-set four digit Vector Number from the keyboard to display:

```
— EDIT VECTORS ——————————————————————— Zm3000 Rev X —

                  Vector No: 0012 █

 Vector     0     Pin          1          Numbers   2
 Number     1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4

 0001       K 0 0 1 H L 0 0 C H H N H P 0 1 1 1 H H H L H N
 0002       C 1 0 1 1 1 1 0 0 1 1 N H L H L Z Z H H L H L N
 0004       C Z 0 1 1 0 C X K H H N L L 0 1 0 1 1 1 C F F N
 0011       X 0 X F H H H H L 0 1 N K P 0 0 1 H L L H Z L N
 ----------------------------------------------------------
 0012       X 0 0 0 P 1 1 0 H L 0 N 0 0 F H H F L L L 0 1 N  <-
 ----------------------------------------------------------
 0021       0 0 1 1 1 F Z Z L 0 L N P K L 0 1 1 X H L P L N
 0025       0 X H H L 1 1 0 0 L 0 N P H H 0 1 0 1 1 F H H N
 0100       0 X H H L 1 1 0 0 L 0 N P H H 0 1 0 1 1 F H H N
 0147       C 0 0 H Z F H L 0 1 1 N 1 1 0 X F H L K H H X N
                              7 = K  8 = P  9 = N  E = Z
```

Selected Vector

All entered Vectors can be scanned by use of the up and down cursor keys.

Press the ENTER key to return to the Vector Function list.
(Pressing RESET will return to the system menu).

Delete Vector

This can be done in two ways:

i. Use the up and down cursor keys to scan all entered Vectors.

When the Vector to be deleted is located press key D.
The display will show:

```
— EDIT VECTORS ——————————————————————————— Zm3000 Rev X —

                      Vector No: 6398

 Vector   0      Pin        1        Numbers    2
 Number   1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4

 6398     X 0 C K C K 0 1 1 1 H N H L F 0 0 1 0 0 1 1 1 N
```

Press any of the cursor keys to delete the displayed vector and to return to the Vector sub-menu. (Pressing RESET will return to the system menu).

ii. Press key D and enter the four digit number of the Vector to be deleted.

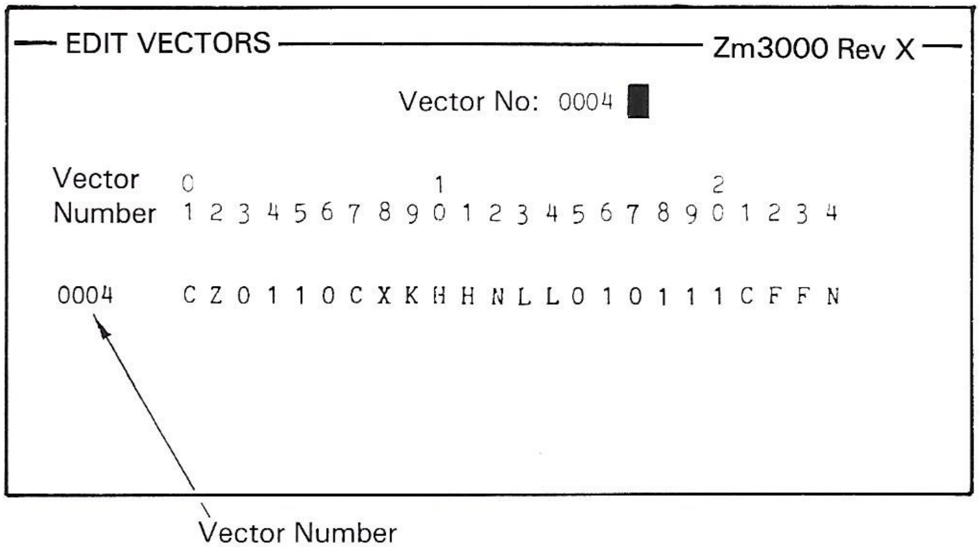
The Vector will be automatically removed and the display will return to the sub-menu.

Modify Vector

Press key X as prompted by the Vector sub-menu.

Use the up and down cursor keys or enter a four digit number to locate the Vector to be modified.

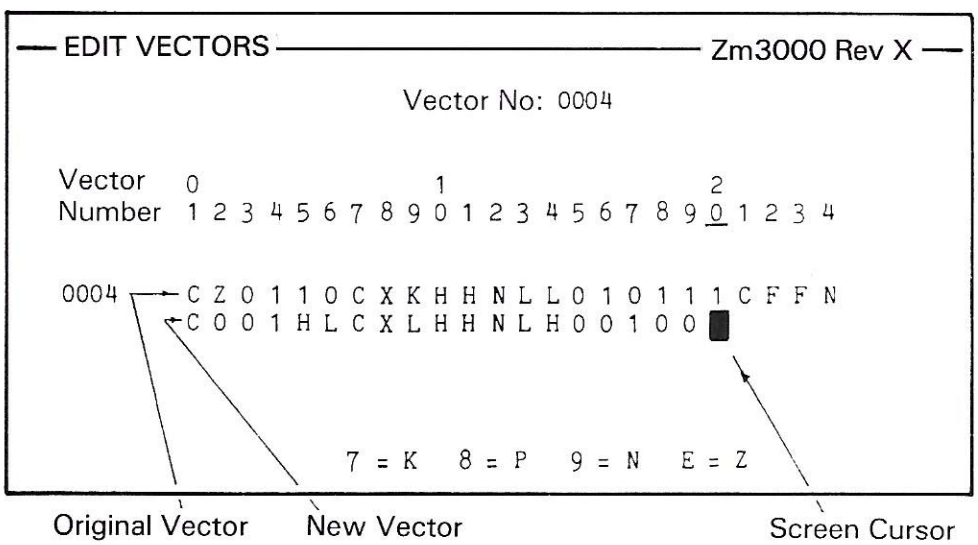
For example:



New characters can be entered under the displayed Vector by use of the keyboard:

0, 1, C, K, N, L, H, Z, F and P are available (for K, N, Z and P use keys 7, 8, 9 and E).

For example:



Pressing the right cursor key will duplicate a character.

Pressing the left cursor key will delete a character.

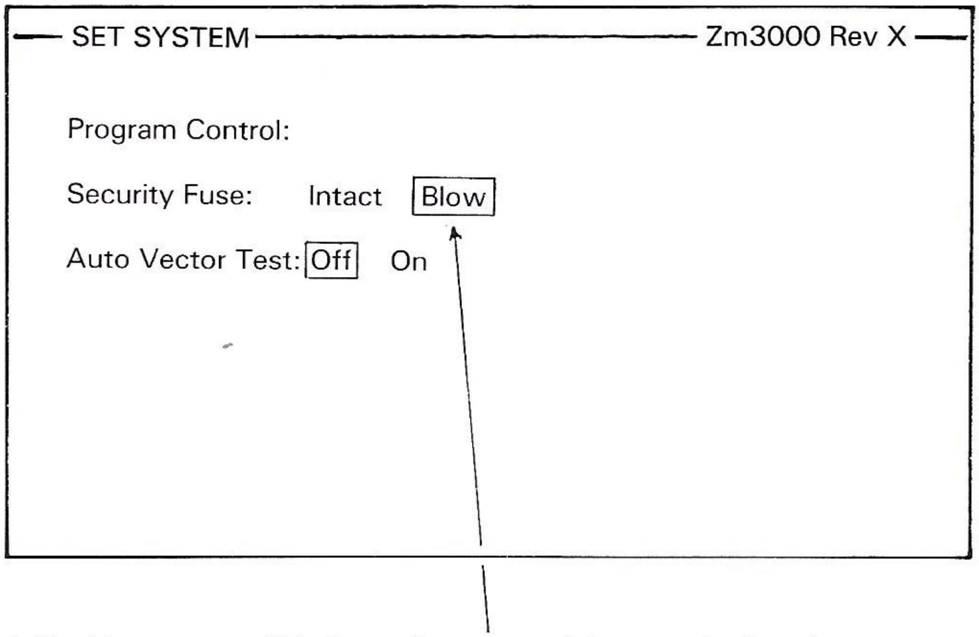
When the last character has been entered the original Vector will be automatically replaced by the modified version.

Pressing the ENTER key will return to the Vector sub-menu. (Pressing RESET will return to the system menu).

12.3 **Selecting the Automatic Vector Test (Post-Program)**

On power-up the Auto Vector Test will be either On or Off.

To check or change the condition press key . . . A followed by key . . . 3 to display.



A Flashing cursor will indicate the status of the security fuse feature.

To change the condition of the Vector Test press the down cursor key and press either the left or right cursor keys.

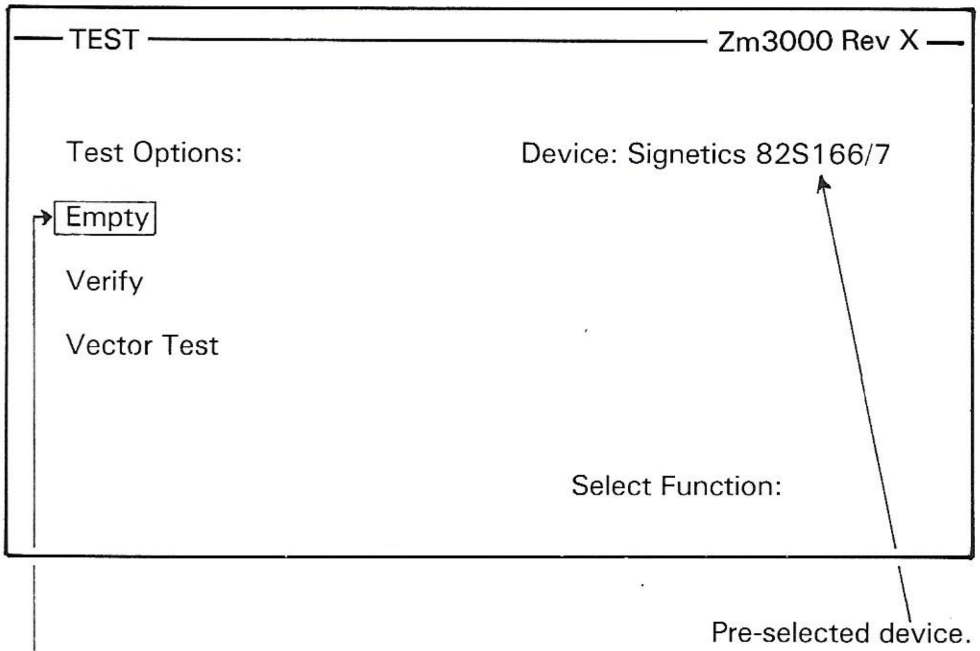
Pressing the ENTER key will implement the selection and return to the set system menu.

Pressing the RESET key will return to the main system menu.

12.4 **Manual Vector Test (by use of the TEST key)**

This is a manual VECTOR TEST and as such can be applied at any time, unlike the in-program VECTOR TEST which is part of an automatic sequence.

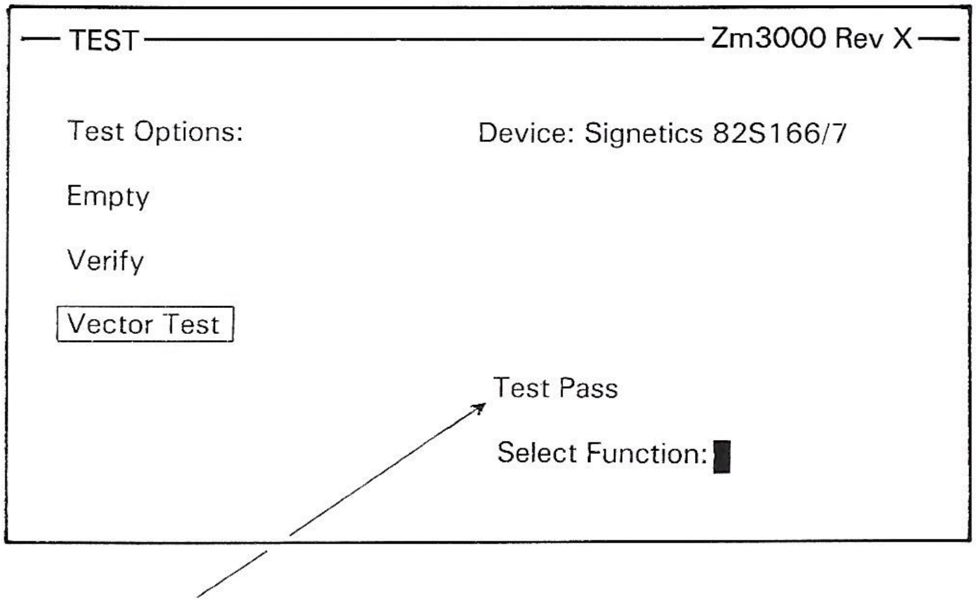
Press the TEST key to display the TEST Menu.



The screen cursor will indicate the "EMPTY TEST" default condition. The "VECTOR TEST" can be selected by pressing the down cursor key twice.

Vector Test Pass

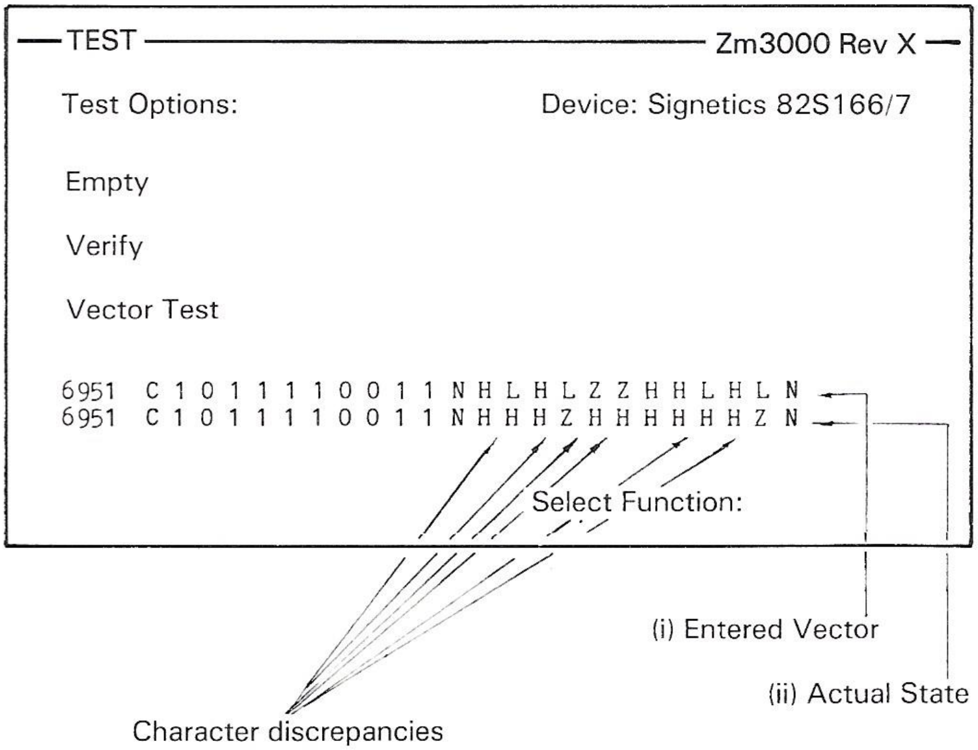
To implement the VECTOR TEST press the ENTER key. If all the device pins monitored are of the expected state the display will show that the test has passed.



Vector Test Pass

Vector Test Fail

When a test fails the display will show two comparative Vectors:



(i) The first is the Vector entered by the user which causes the failure.

(ii) The second Vector shows the actual state of the device pins monitored during the test.

The device pins that fail the test are indicated by character discrepancies between the two Vectors.

Pressing the ENTER key will show more Vector test fails if they are present.

When all Vector fails have been scanned, the display will show ''Vector Test Fail''.

Pressing RESET will return to the system menu.

SECTION 13

stag

Sophisticated systems for the discerning engineer.

13 Interface

13.1 Setting the I/O interface parameters for both RS232C ports and the IEEE-488

On power up the I/O defaults to the last used parameters.

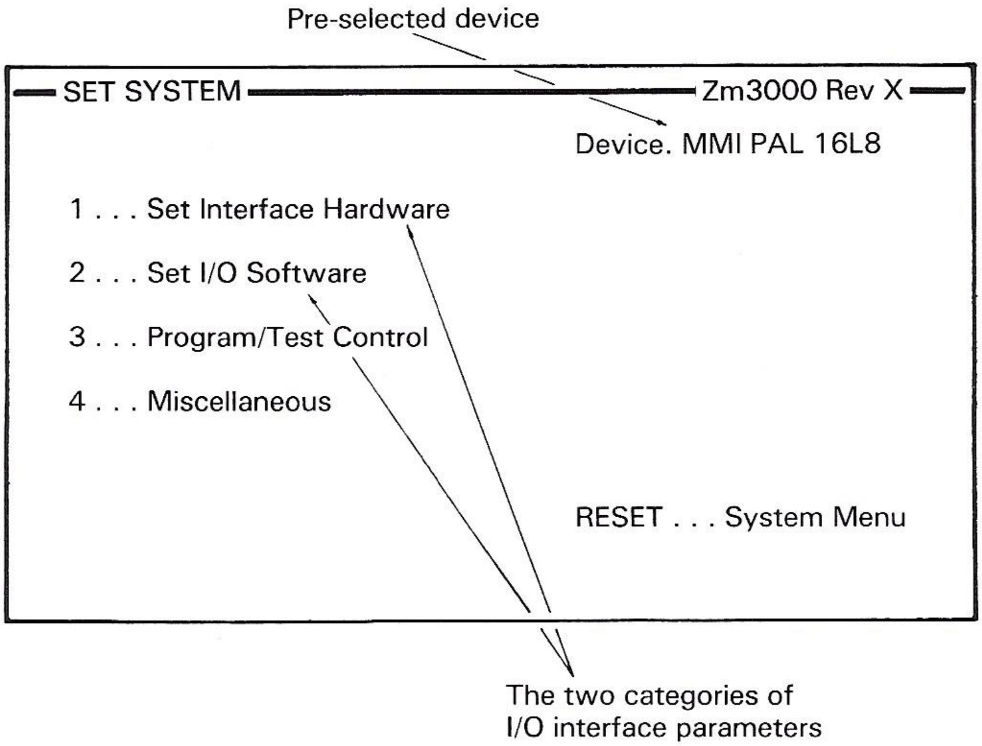
There are five RS232C categories of pre-set I/O interface parameter available on the Zm3000.

They are divided to come under two separate headings: ''Set Interface Hardware'' and ''Set I/O Software'' which are displayed on two separate screen pages:

Interface parameters on Zm3000

Set Interface Hardware (Page 1)	Set I/O Software (Page 2)
Baud Rate Word Length Parity Stop Bits	Format Selection (plus control Z option)

Entry into both of these categories is first made by pressing key . . . A from the system menu to display:



''Set Interfac

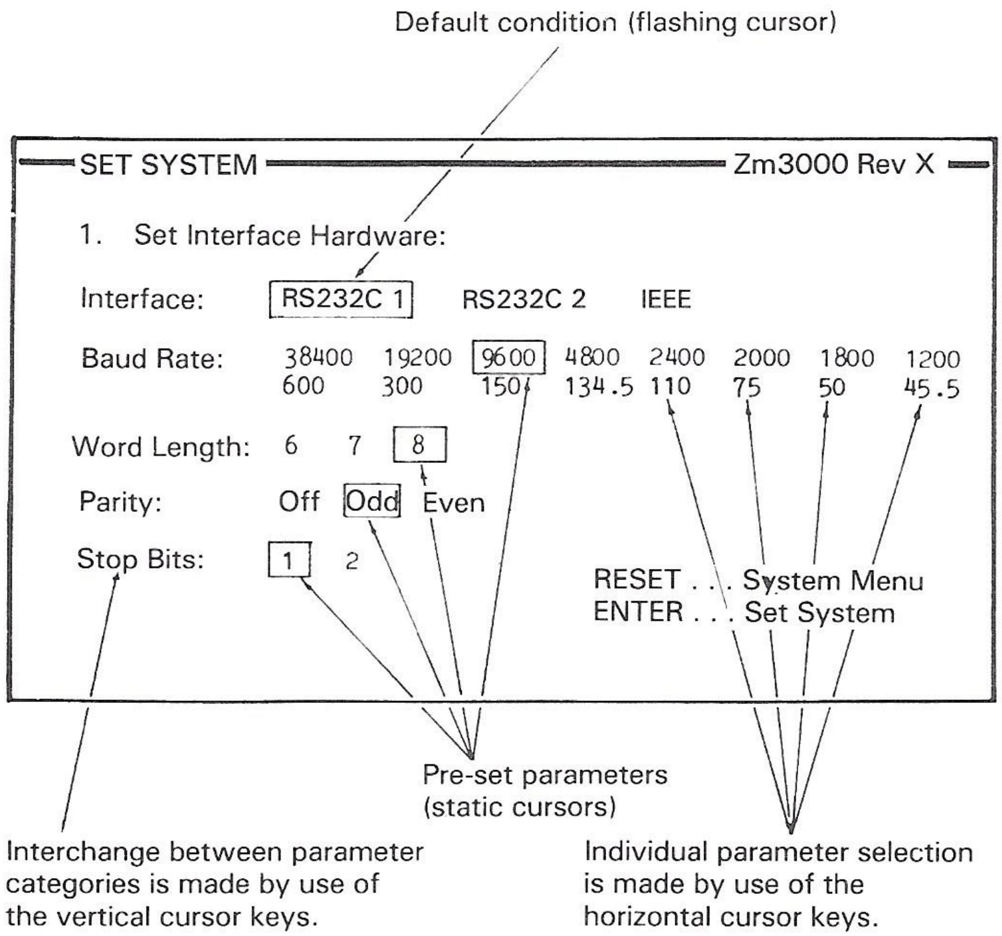
Set Interface Hardware

Independent hardware parameters can be entered for both of the serial interface ports available on the System 3000.

Selecting Hardware parameters for the RS232C 1 (female connector).

Pressing key . . . A as prompted by the system menu followed by key . . . 1 will show a flashing screen cursor defaulted to the heading RS232C 1 plus four other static cursors indicating the last entered I/O parameters for this particular port.

For example:



Entry into a particular parameter category is made by use of the vertical cursor keys and is indicated by the applicable screen cursor flashing.

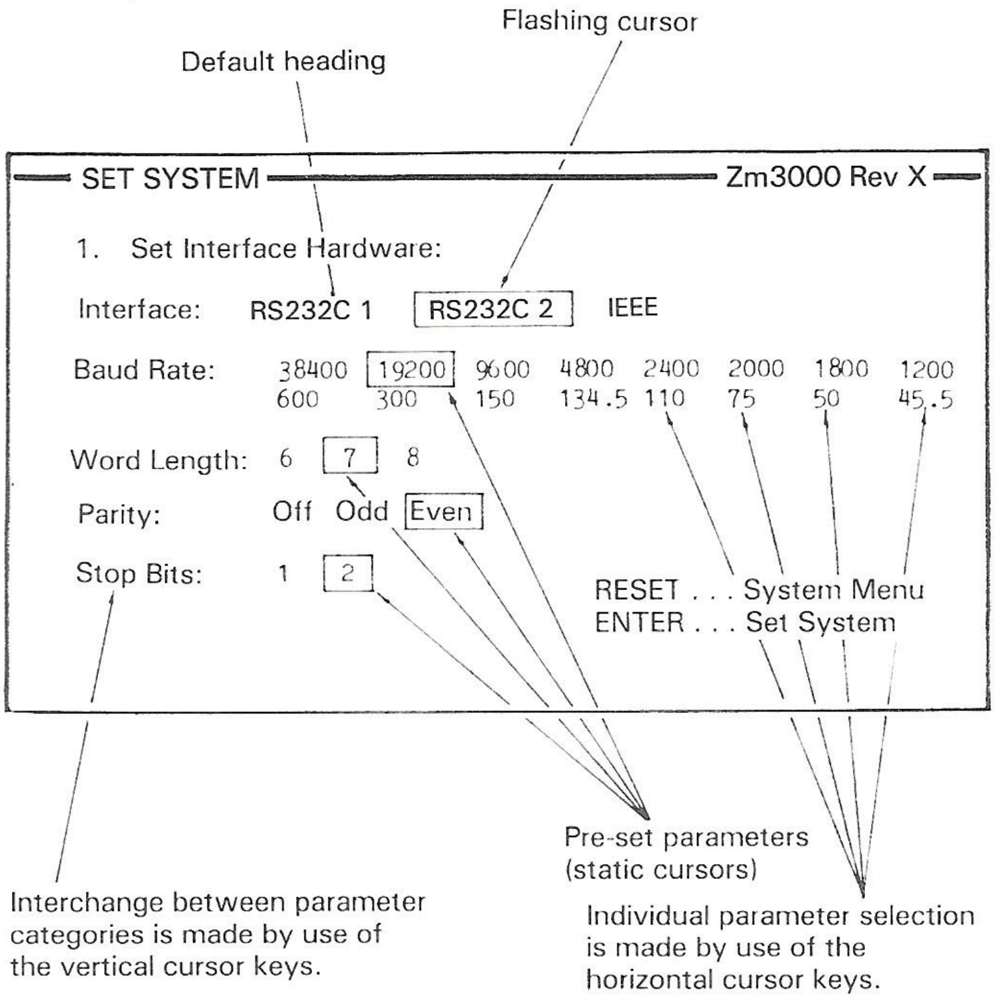
Individual parameter selection within each category is made by use of the horizontal cursor keys which direct the screen cursor from the last entered selection to the required alternative.

Pressing the ENTER key will implement the selections and pressing RESET will return to the system men

Selecting Hardware parameters for the RS232C 2 (male connector).

Pressing key . . . A from the system menu followed by key . . . 1 will show a flashing "screen cursor" defaulted to the heading RS232C 2. By pressing the right cursor key the flashing "screen cursor" will skip onto the next heading RS232C 2 where four other static screen cursors will indicate the last entered I/O parameters for this particular interface port.

For example:



Entry into a particular parameter category is made by use of the vertical cursor keys and is indicated by the applicable screen cursor flashing.

Individual parameter selection within each category is made by use of the horizontal cursor keys which direct the screen cursor from the last entered selection to the required alternative.

Pressing the ENTER key will implement the selections and pressing RESET will return to the system menu.

Selecting the address parameter for the IEEE-488 Talker/Listener Interface Port

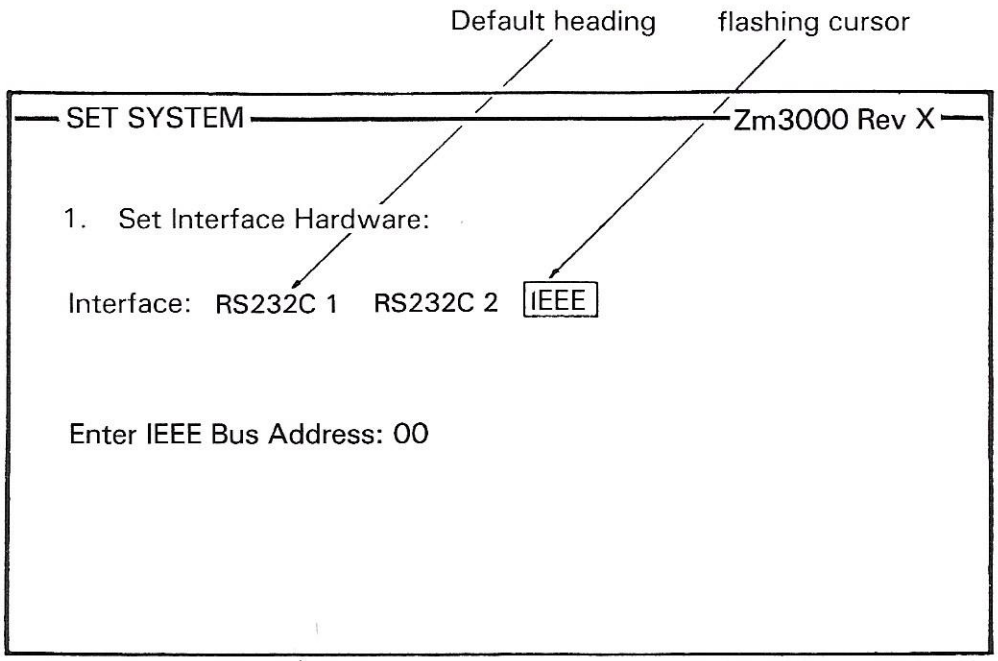
A two digit coded address can be entered that will allow a remote-controller connected via the IEEE interface port to identify the System 3000.

To select the address parameter from the SYSTEM MENU:

PRESS KEY A followed by KEY 1

This will display the screen cursor defaulted to the RS232C 1 heading. To select the IEEE:

PRESS KEY → twice to display:



Pressing the down cursor key (↓) will make the address available for selection, indicated by the pre-set parameter flashing. The new address can now be entered from the keyboard. To confirm this selection:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

Set I/O Software (Data Transfer Formats)

Three data transfer formats are available on the Zm3000 for logic devices. These are:

Signetics
X-Plot
JEDEC

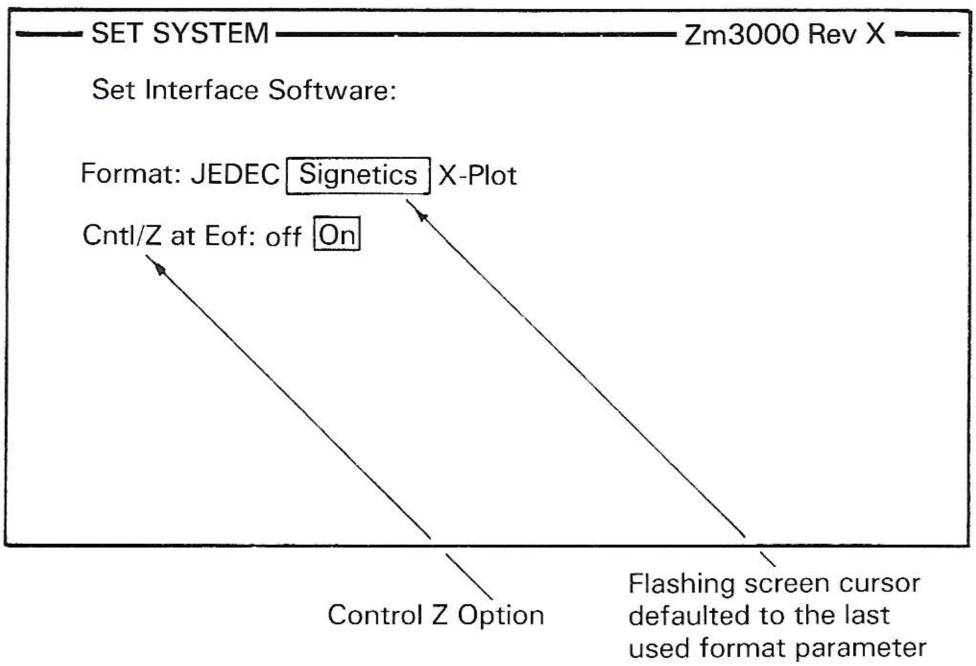
Signetics is a dedicated IFL data transfer format.

X-Plot is a dedicated PAL data transfer format.

JEDEC is independent of device architecture.

Selecting the format parameter

Press key . . . A as prompted by the system menu followed by key . . . 2 to display:



The display shows a flashing screen cursor defaulted to the last used Format Parameter plus a second static cursor indicating the availability of the Control Z option.

Selection can be made by use of the horizontal cursor keys which direct the screen cursor from the last entered selection to the required alternative.

The control Z option can be selected by use of the down cursor key and specified as either ON or OFF by use of the horizontal cursor keys.

Pressing the ENTER key will implement the selection and pressing the RESET key will return to the system menu.

13.2 I/O Mode

Only two of the available interface ports can be used for INPUT and OUTPUT when the System 3000 is in the LOCAL MODE of operation; these are RS232C 1 and RS232C 2.

The IEEE-488 interface is a talker/listener port and may only be used in REMOTE CONTROL.

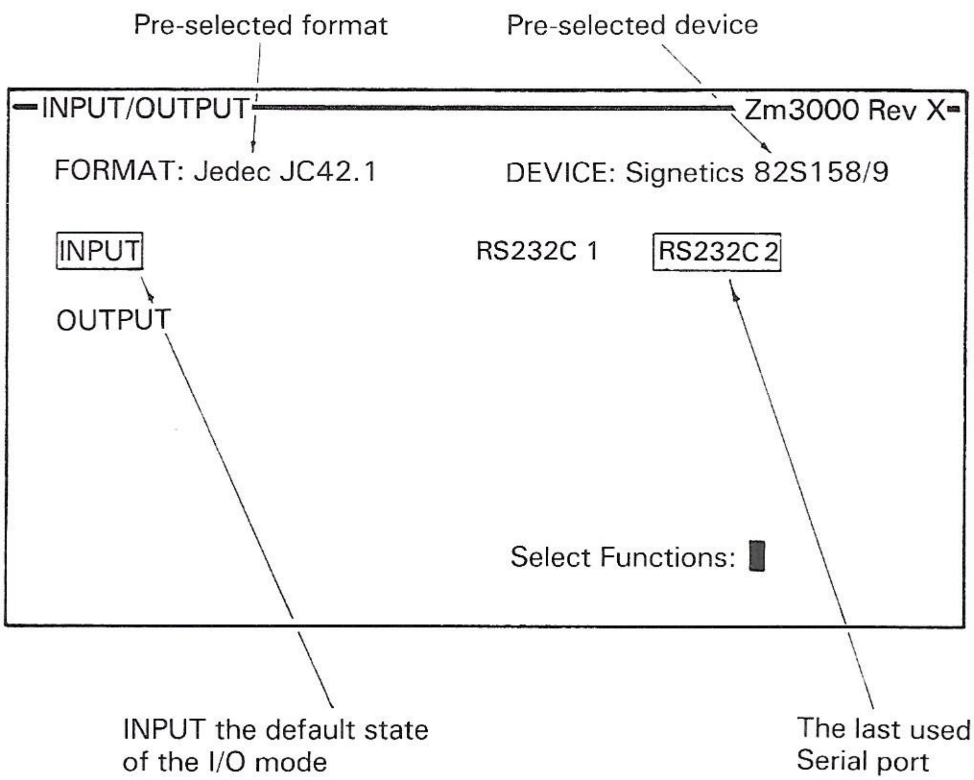
The RS232C ports can be selected to operate in either INPUT or OUTPUT modes.

Selecting the I/O mode

To enter into the I/O mode press the I/O key.

The display will show the I/O default conditions in video reverse.

For example:



INPUT is the default state of the I/O mode, OUTPUT can be selected by use of the down cursor key.

The required serial port can be selected by use of the horizontal cursor keys.

NOTE: Should the Pre-selected format and device prove incompatible when pressing the I/O key the display will show "Invalid Format Code". i.e. X-Plot/IFL and Signetics/PAL

13.2.1 Input

When the required Serial port has been selected, pressing the ENTER key will implement the INPUT.

The display will initially show "loading" in progress followed by "Load Complete". i.e.

```
— INPUT/OUTPUT ———————————————————————— Zm3000 Rev X —
  FORMAT: Jedec JC42.1          DEVICE: Signetics 82S158/9

 [INPUT]                        RS232C 1    [RS232C 2]

  OUTPUT



                                 Load Complete

                                 Select Functions: █
```

Pressing the RESET key will return to the SYSTEM MENU.

ERROR REPORTING ON INPUT

Four possible messages can be displayed by the System 3000 at the end of a data INPUT instead of "Load Complete"

These are:

1. Illegal Character

2. Illegal Field Spec

13.2.2 **Output**

When selection of OUTPUT and the required Serial port have been made, press the ENTER key to implement the OUTPUT.

The display will initially show ''Dumping'' in progress followed by ''Dump Complete'' i.e.:

```
— INPUT/OUTPUT ——————————————————————— Zm3000 Rev X —
  FORMAT: Jedec JC42.1          DEVICE: Signetics 82S158/9

  INPUT                        RS232C 1   [RS232C 2]

 [OUTPUT]



                                Dump Complete

                                Select Functions: █
```

Pressing the RESET key will return to the SYSTEM MENU.

SECTION 14

stag

Sophisticated systems for the discerning engineer.

14 Format Descriptions

14.1 JEDEC JC 42.1 1981

The JEDEC format is a data transmission format for the transfer of information between a data preparation system and a device programmer and is independent of device architecture. The information is divided into five categories:

1. The design specification identifier.
2. The device to be programmed.
3. Fusing information for the implementation of the design specification.
4. Test vectors for structured functional testing.
5. Additional information.

The Design Specification Identifier

The design specification identifier consists of:

(i) ASCII STX (Start transmission).

(ii) User's name and company.

(iii) The design spec date, part number and revision.

(iv) The part number of the manufacturer's device.

(v) Other information.

(vi) An asterisk.

The Device to be Programmed

A field is defined to specify the device to be programmed. The device is specified by a four-digit code. The code is preceded by a D and terminated with an asterisk.

Fusing Information

Each fuse in a device is allocated a decimal fuse number. The state of each fuse is indicated by a '0' for an intact fuse and a '1' for a blown fuse. Fuse information is presented in three fields (F, L and C). The F field specifies the default state for fuses not otherwise defined. The L field specifies the states of the fuses. The C field is a checksum.

The F field must appear before the other fields. The F field commences with an F followed by 0 or 1 and terminates with an asterisk.

The L field defines the states of specified fuses in the device. It commences with an L followed immediately by decimal characters indicating the starting fuse number for a string of data. The first 0 or 1 is preceded by a space and the string is terminated with an asterisk. The data string may be of any convenient length and more than one L field may be used to specify a device. If a fuse is specified more than once the latest information will be used.

The C field is a checksum field. It commences with a C which is followed by four hex. characters representing the checksum and is terminated with an asterisk. The checksum is calculated by the 8-bit addition of 8-bit bytes constructed from the specified state of each fuse in the device, bit 0 of byte 0 being fuse number 0, bit 1 being fuse number 1 etc. The word containing the last fuse of the device is constructed by setting to zero all bit locations corresponding to fuse numbers higher than the last fuse in the device.

Structured Test Information

A field is defined to allow the loading of test vectors for functional testing of the device. A test vector defines a combination of input and output states for a device. A test vector must be preceded by a V. This is followed immediately by decimal characters representing the number of the test vector. This number is followed by a space and then the test variables. Each test vector is terminated by an asterisk. Vectors are applied in numerical order. If a test vector number appears more than once the latest test vector will be used.

Additional Information

Additional fields may be defined in the future using other letters.

Ending the Format

The transmission must be terminated by ETX (End Transmission). A checksum of four hex. characters must follow immediately. This checksum is the 16-bit sum of the ASCII values of the transmitted characters between, and including, the STX and ETX. The parity bit must be excluded during this calculation.

STX = 02 Hex
ETX = 03 Hex

Example

	STX	
Design	Peter Collins	user's name
Specification	Stag	user's company
	12.12.86	design spec. date
Identifier	DS 123-2345	design spec. part number
	PN ALT. EP310	design manufacturer's part number
Fusing	*F0	fusing default state
Information	*L00000 00110101001010	states of fuses
	*L00014 0111100010	
	*C0187	fuse checksum
	*ETX4976	

14.2 **Signetics IFL Formats**

82S152/153

A typical format for the 82S152/153 is as shown below:

```
STX
*POLHLLLHLLLLH
*P 00 * I - - - - HH- -   * BI - - - - - - - - - -   * BO. . . A. . . . . .
*P 01 * I - - - - HL- -   * BI - - - - - - - - - -   * BO. . A. . . . . . .
*P 02 * I - - - - LL- -   * BI - - - - - - - - - -   * BO. A. . . . . . . .
*P D1 * I - - - - - - - -   * BI - - - - - - - - - -
*P D0 * I - - - - - - - -   * BI - - - - - - - - - -
ETX
```

The start of the data field is marked with STX. The output polarity is then entered starting with the identifier *POL followed by the required states of the outputs B9 to B0. The P-terms are then entered using the P-term identifier *P followed by the P-term number. Note that the control term numbers start with D. The input variables are then entered using the identifier *I followed by the data for I7 to I0. The I/O variables are then entered with the identifier *BI and data B9 to B0. The output functions are then entered identified by *BO and data B9 to B0. The field is terminated with ETX. Note that the control terms have no output function.

A typical JEDEC format for the 82S152/153 is shown below:

```
*D1405
*G0
*L0000 1111010111111111111111111111111111111110111111
*L0046 1111100111111111111111111111111111111101111111
*L1796 000000000000000000000000000000000000
*L1832 0111011110
*C71D7
*A040
```

A typical format for the 82S158/159 is as shown below.

```
STX  *EAA *F/FA.A.A.A.*POLLHLH
*T 00 *C.  *I HHHH *BI HHHH *QPHHHHHH- HH *QNHHHHHHHHH *BOA. A.
*T 01 *C.  *I HHHH *BI HHHH *QPHHHHHHHHH *QNHHHHHHHHH *BOA. A.
*T 02 *C-  *I HHHH *BI HHHH *QPHHHHHHHHH *QNHHHHHHHHH *BOA. A.
*T FC *C.  *I HH-H *BI HHHH *QPHHHHHHHHH
*T PB *C-  *I LLLL *BI LLLL *QPLLLLLLLLL
*T RB *C.  *I HHHH *BI HHHH *QPHHHHHHHHH
*T LB *C-  *I LLLL *BI LLLL *QPLLLLLLLLL
*T PA *C-  *I LLLL *BI LLLL *QPLLLLLLLLL
*T RA *C.  *I -HHH *BI HHHH *QPHHHHHHHHH
*T LA *C.  *I HHHH *BI HHHH *QPHHHHHHHHH
*T D3 *C-  *I LLLL *BI LLLL *QPLLLLLLLLL
*T D2 *C.  *I HHHH *BI HHHH *QPHHHHHHH-H
*T D1 *C-  *I LLLL *BI LLLL *QPLLLLLLLLL
*T D0 *C.  *I HHHH *BI HHHH *QPHHHHHHHHH
EXT
```

The data field is started with STX. The information for the enables is then entered as *E and then EB and EA. This is followed by the flip-flop type information and the output polarities. The transition terms are then entered as *T and the T-term number followed by the input data, the I/O data, the present state and the next state. The field is terminated with ETX.

14.3 Fuse-Plot

The fuse-plot format is provided to give a readable format for 20 and 24 pin PLAs. It follows closely the logic diagram of these circuits.

A typical X—PLOT is shown below:

```
P NO   0123  4567  8901  2345  6789  0123  4567  8901
00     ....  ....  ....  ....  ....  ....  ....  ....
01     ....  ....  ....  ....  ....  ....  ....  ....
02     ....  ....  ....  ....  ....  ....  ....  ....
03     ....  ....  ....  ....  ....  ....  ....  ....
04     ....  ....  ....  ....  ....  ....  ....  ....
05     ....  ....  ....  ....  ....  ....  ....  ....
06     ....  ....  ....  ....  ....  ....  ....  ....
07     ....  ....  ....  ....  ....  ....  ....  ....
32     XXXX  XXXX  XXXX  XXXX  XXXX  XXXX  XXXX  XXXX
33     ----  ----  -X--  ----  ----  ----  -X--  ----
34     XXXX  XXXX  XXXX  XXXX  XXXX  XXXX  XXXX  XXXX
35     ----  ----  ----  ----  ----  ----  ----  ----
36     XXXX  XXXX  XXXX  XXXX  XXXX  XXXX  XXXX  XXXX
37     ----  ----  ----  ----  ----  ----  ----  ----
38     XXXX  XXXX  XXXX  XXXX  XXXX  XXXX  XXXX  XXXX
39     ----  ----  ----  ----  ----  ----  ----  ----
```

20 pin PLA

The fuse matrix is described as a number of P-terms. Each P-term has a two-digit decimal number followed by 32 characters representing the required conditions of the 16 input lines. The characters are X for an intact fuse, - for a blow fuse and . for a "phantom fuse". A typical X-plot input is shown below:

```
STX
00 X - X - - - - - - - X - X - X - X - - - - - - X - - - - - X X -
ETX
```

The data field is started with STX (B̄). This is followed by the P-term number and the fusing data for that P-term. The data field is terminated with ETX (C̄). It is not necessary to provide data for unused terms.

24 pin PLA

The format for the 24 pin devices is similar to that for the 20 pin devices. The main difference is the number of fuses in the P-term, this being 40 for the 24 pin devices.

SECTION 15

15

stag

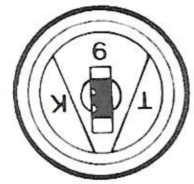
Sophisticated systems for the discerning engineer.

15 Remote Control (Logic Devices only)

15.1 Selecting the operatic modes

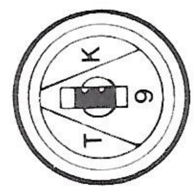
Selection of the operating mode is made by use of the three position key-switch located underneath the hinged flap. By inserting the supplied key one of these modes can be chosen.

LOCAL MODE (arrow vertical)



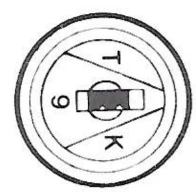
In LOCAL mode all functions of the Programmer are controlled from its own keyboard. All the functions operate interactively with the CRT and Input/Output transmissions take place between the Programmer and the peripheral equipment under 'Local Control'.

REMOTE MODE (arrow pointing to the left)



In REMOTE the Programmer operates under remote control from a computer or a terminal. The keyboard of the Programmer is inoperative at this time and the CRT will only display information as requested under remote command.

EDIT LOCK-OUT MODE (arrow pointing to the right)



The Edit Lock-out mode ensures all editing key functions are inoperative but that all other functions operate normally under LOCAL MODE conditions.

In this mode data corruption or changes to the RAM data cannot take place and security of information is ensured.

Interchange between LOCAL and REMOTE MODE can be accomplished by turning the key-switch to the required direction and pressing RESET.

Interchange between LOCAL and EDIT LOCK-OUT MODE can be made by simply turning the key to the required direction. (Should an edit key function be requested under Edit Lock-Out Mode the CRT will display: EDIT LOCKED OUT)

15.2 **Setting Up Procedure for Remote Control**

Before engaging REMOTE CONTROL, two selections have to be made:

1. the required interface.

2. the REMOTE CONTROL mode.

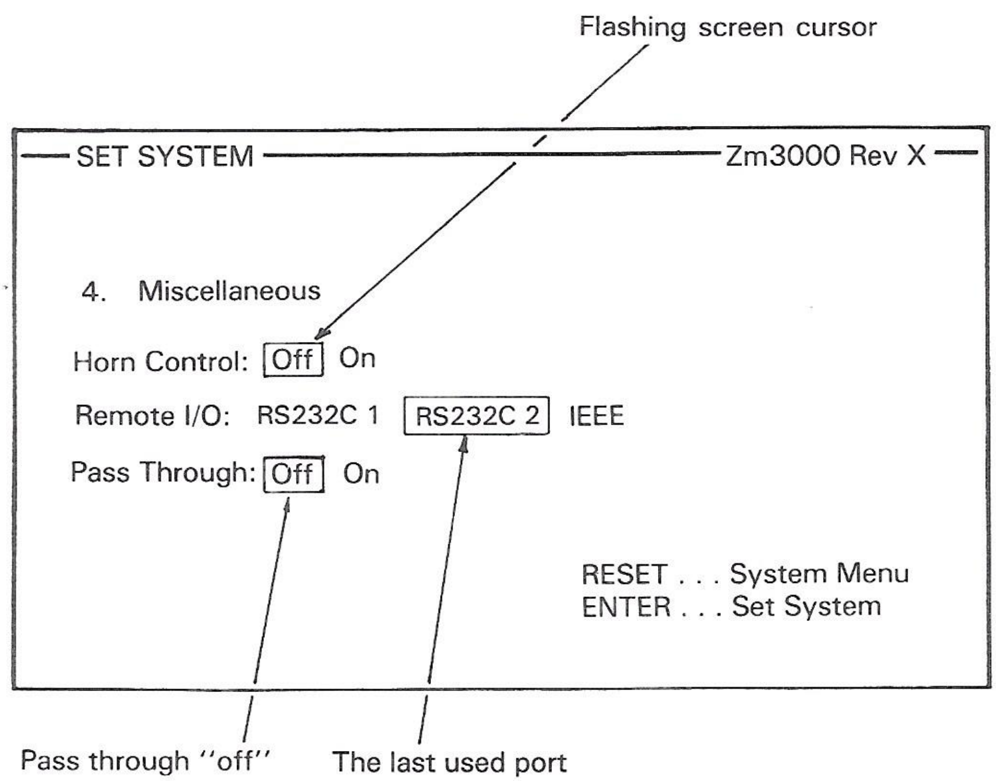
Selecting the required interface

From the SYSTEM MENU:

PRESS KEY A followed by KEY 4

This will display the 'Miscellaneous' page of the Set System option which will show a flashing screen cursor defaulted to HORN CONTROL and two static cursors. One of the static cursors will indicate the interface port last used in the REMOTE CONTROL mode and the other will indicate whether PASS-THROUGH IS 'on' or 'off'.

For example:

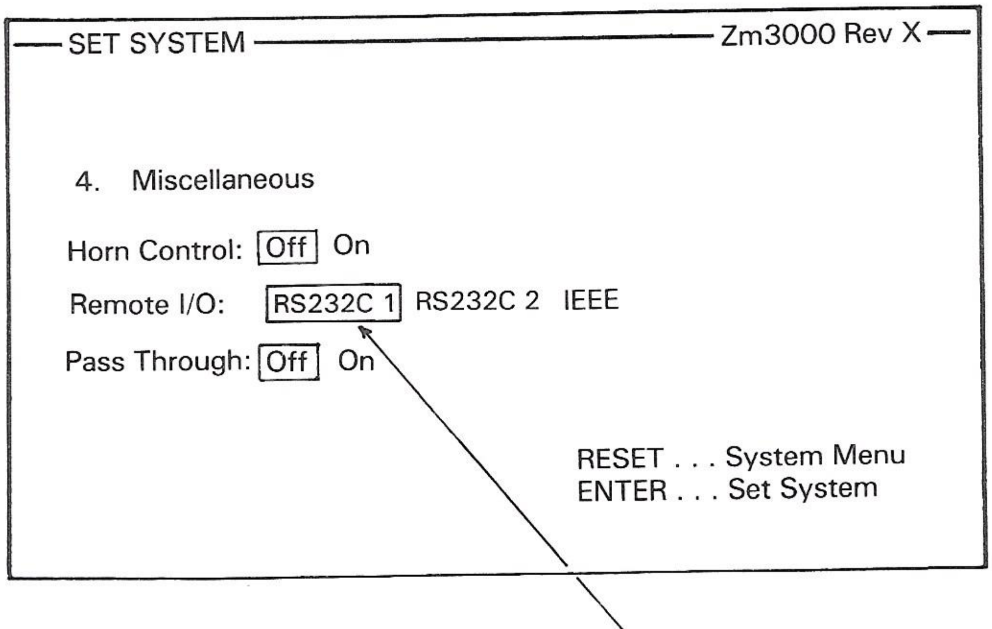


If, as in this example the RS232C 2 port was previously used in REMOTE MODE and RS232C 1 is the new requirement, then it can be selected as follows:

PRESS KEY ↓

This will move the flashing cursor onto RS232C 2. To select RS232C 1:

PRESS KEY ←



Screen cursor flashing at new selection of interface.

To confirm this selection:

PRESS ENTER

To return to the SYSTEM MENU:

Selecting the address parameter for the IEEE

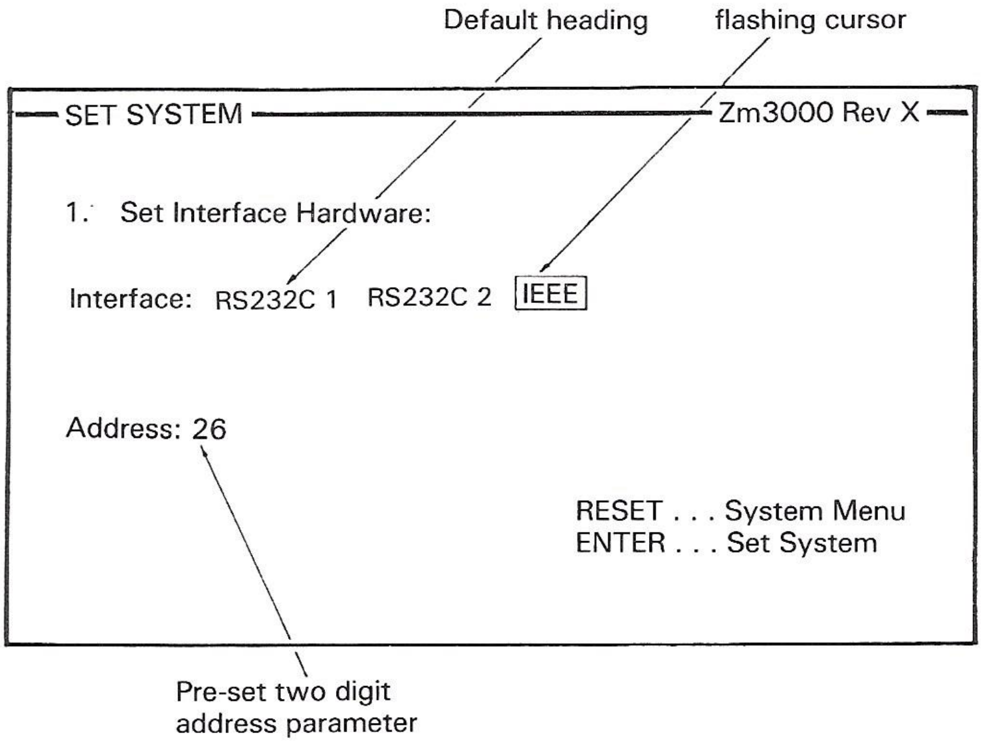
A two digit coded address can be entered that will allow a remote-controller connected via the IEEE interface port to identify the PPZ Programmer.

To select the address parameter from the SYSTEM MENU:

PRESS KEY A followed by KEY 1

This will display the screen cursor defaulted to the RS232C 1 heading. To select the IEEE:

PRESS KEY → twice



Pressing the down cursor key will make the address available for selection, indicated by the pre-set parameter flashing. The new address can now be entered from the keyboard. To confirm this selection:

PRESS ENTER

To return to the SYSTEM MENU:

PRESS RESET

Selecting REMOTE CONTROL mode

Selecting the REMOTE MODE involves the 3-position key switch which is located beneath the hinged flap on the top panel of the Programmer.

To select REMOTE MODE from LOCAL MODE:

TURN THE KEY CLOCKWISE (arrow pointing left)

PRESS RESET

The screen will display:

```
── REMOTE CONTROL ──────────────────── Zm3000 Rev X ──
```

The programmer is now configured for operation in REMOTE CONTROL mode via either of the serial interface ports and the keyboard of the Programmer will be inoperative.

To return to the LOCAL mode:

TURN THE KEY COUNTER CLOCKWISE ONE STOP (arrow pointing down)

PRESS RESET

The SYSTEM MENU will now be displayed.

Structure of the IEEE as a Talker/Listener Port

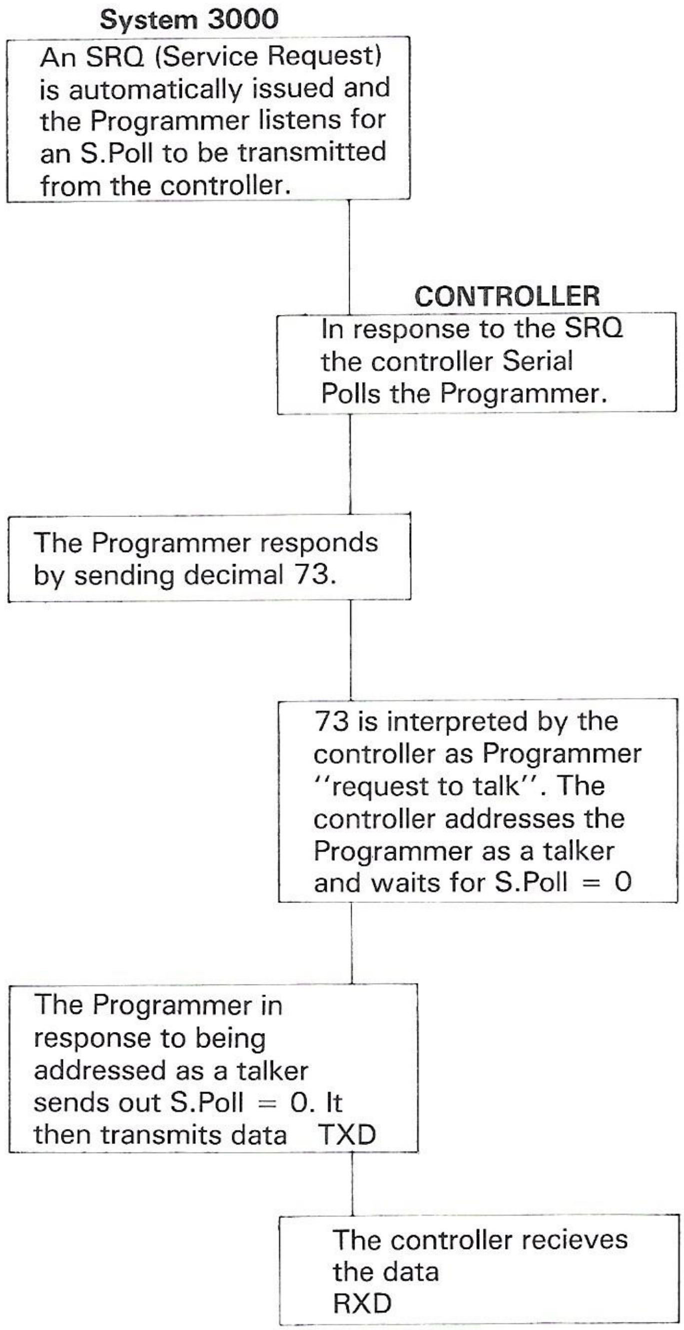
Certain communication procedures govern the set-up and operation of the Programmer in the REMOTE CONTROL mode with a controller via the IEEE interface port.

Some of the communication is inherent to the controller and some is Programmer based but both are interlinked and will respond to interpretations of instructions generated by the other.

Note: The communication procedure must be followed precisely, in order to prevent ''bus lock-up'' (a state whereby communication between the two machines breaks down).

This procedure can be shown in the form of two flow charts.

(i) The Programmer as a talker (TXD)



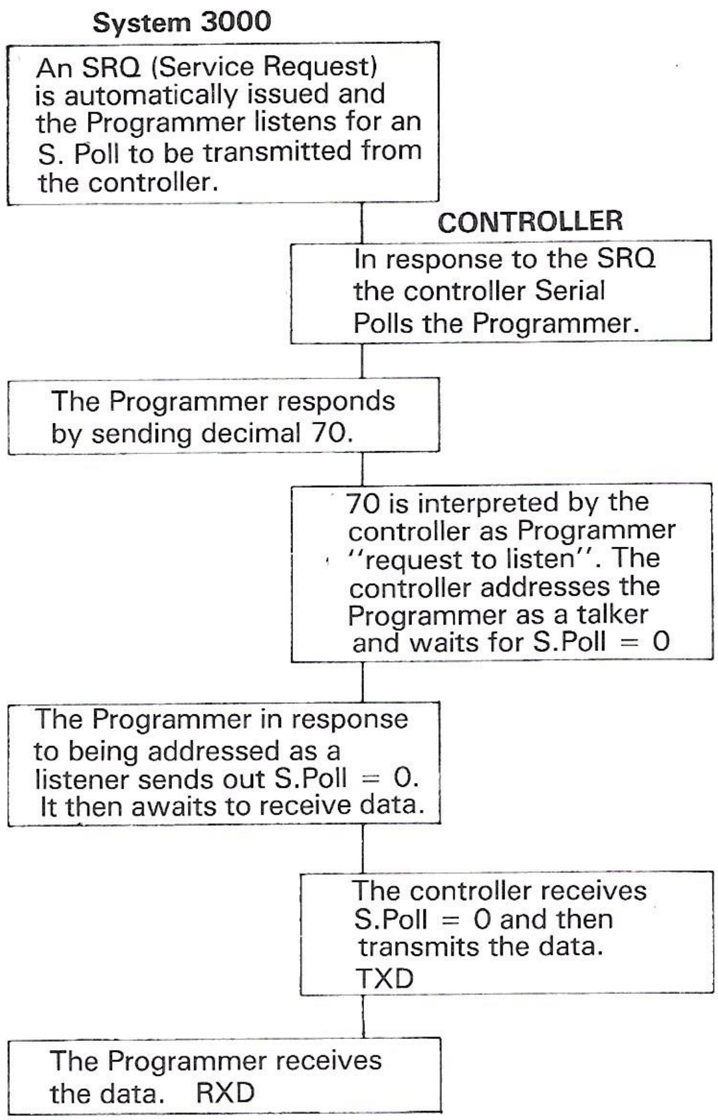
Notes:

(i) on power-up the Programmer transmits a pass response.

(ii) when the Programmer is not transmitting or receiving data, the S.Poll will equal decimal 128. This occurs during programming, load, etc.

(iii) a rolling cursor will indicate a dump is in progress.

(ii) **The Programmer as a listener (RXD)**



Notes:
- (i) on power-up the Programmer transmits a pass response.
- (ii) when the Programmer is not transmitting or receiving data, the S.Poll will equal decimal 128. This occurs during programming, load, etc.
- (iii) a rolling cursor will indicate a load is in progress.

IEEE 488 Interface port capabilities

Note: An abbreviated list of the IEEE capabilities is printed on the back panel below the port itself.

SH1 Full source handshake capability.

AH1 Full acceptor handshake capability.

T2 Full talker function capability and full serial Poll capability.

L2 Full listener function capability.

SR1 Full service request function capability.

RL1 No remote/local function capability.

PP1 Parallel Poll capability without direct configuration from controller.

DCO No capability for device clear function.

DTO No capability for device trigger.

CO No controller function.

E1 Open collector drivers.

The following gives the signals on the IEEE 24 way connector:

Pin No	Name
1	DIO1
2	DIO2
3	DIO3
4	DIO4
5	DOI
6	DAV
7	NRFD
8	NDAC
9	IFC
10	SRQ
11	ATN
12	GND

Pin No	Name
13	DIO5
14	DIO6
15	DIO7
16	DIO8
17	REN
18	GND
19	GND
20	GND
21	GND
22	GND
23	GND
24	GND

15.3 Remote Control Commands

To execute commands via a serial interface, send a CR,LF (Carriage Return, Line Feed).

To execute commands via the IEEE-488, send an EOI.

'Set' Commands

S0	MMDDD	Select the device—MM = manufacturer code, DDD = device code. (In the case of a 4 digit code, insert a zero between the first and second pair of digits).
S1	N	Set I/O format N. See table 1 for list of format codes.
S2	NNNN	Set number of devices for handler. NNNN is in decimal.
S3	N	Set security fuse status. 0 = Intact 1 = Blow

Table 1—Format Codes

Code	Format
0	JEDEC.
1	Signetics.
3	X-Plot

'Read' Commands

R0	Read device code.
R1	Read format currently selected.
R2	Read number of devices set for handler.
R3	Read security fuse status.
R4	Read number of open links.
R5	Read RAM size.
R6	Read software revision number.
R7	Read RAM checksum.
R8	Read handler code (returns FF if no handler).

Device Commands

P0	Program with full autosequence.
P1	Program without pre-program checks.
P2	Program without post-program verify.
P3	Program without pre- or post-programming checks.
P4	Program without vector test.
P5	As P1 but without Vector Test.
P6	As P2 but without Vector Test.
P7	As P3 but without Vector Test.
L	Load device to RAM.
E	Empty check.
V	Verify.
T	Vector Test.

Note: Vector tests will only be carried out if valid test vectors for the device under test have been loaded.

RAM and I/O Commands

I	Input data to RAM from the interface.
O	Output data from RAM to the interface.

Miscellaneous

H nn	Sound horn n times. 'n' is in decimal.	
D string	Display ASCII string.	
K	Wait for any key to be pressed.	
Z	Escape from remote control.	
Y	Output yield data which is of the form:	
	DDDD	Number of devices handled.
	NNNN	Number of good parts.
	BBBB	Number of pre-program failures.
	VVVV	Number of post-program failures.
	TTTT	Number of vector test failures.

Responses to Commands

Some commands require the programmer to output data to the controlling equipment. In general, the data is of the same form as that for the corresponding 'Set' command.

For instance, the RS232C response to the R0 (read device code) command will be:

MMNNN, CR, LF, SS, CR, LF, >

Where SS is the status response.
The IEEE-488 response will be:

MMNNN (EOI), status returned in status byte.

RS232C Status Responses

Hex Code	Status
00	Command executed O.K.
01	No Blow.
02	Verify Error.
03	Illegal Fuse.
04	Not Empty.
05	Connect Error.
06	Reversed Device.
07	Test array failure (mainly IFLs).
08	Out of range address on download.
09	Security fuse fail.
0A	Illegal or Unrecognised command.
0B	Load Error on Download.
0C	Consecutive failures on handler.
0D	RAM Fail.
0E	Vector test failure.
0F	Handler time out.

IEEE 488 Status Responses

The IEEE 488 responses are placed in the status byte. Bit 7 (MSB) of the byte is used to indicate that the programmer is busy and cannot accept remote control commands other than serial and parallel poll commands. If the programmer is addressed as a listener while bit 7 is set, it will assert NRFD (Not Ready For Data) and lock up the bus until the command is completed. If the programmer is addressed as a talker, it will not respond until bit 7 goes to zero.

Bit 6 of the status byte is the SRQ bit and indicates that the programmer is generating an SRQ. The programmer will generate an SRQ on completion of the E, V, T, P and K commands.

Bit 5 of the byte is used to indicate that the programmer has data to be transmitted in response to a command. A zero bit indicates no data and a one bit indicates data. If there is data to be transmitted, the IEEE 488 controller must read it before another command can be accepted.

Bits 4 to 0 provide a five bit code supplying error information. The codes are the same as for the RS232C responses.

Additional IEEE 488 Remote Control Commands

The IEEE 488 remote control commands are similar to those specified for the serial interface ports but an important addition should be noted.

EOI is used to terminate data strings, commands and error responses. The manner in which EOI is sent is dependent upon the way the 'controller' has been configured.

Section 16 — 17

SECTION 16

16

stag

Sophisticated systems for the discerning engineer.

16 RS232C Hardware Description

16.1 RS232C Interface Port Connections

The System 3000 can be linked to peripheral equipment via one or both of its RS232C interface ports.

RS232(1) Serial I Female connector

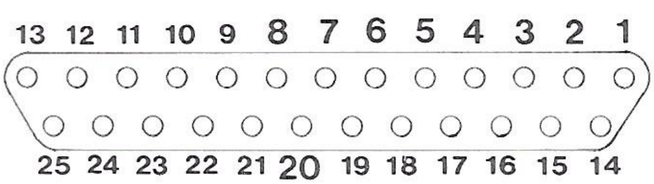
The RS232(1) port is a D type female connector configured as DTE (Data Terminal Equipment) which is able to accept a 25 way male connector.

It is generally used to connect the System 3000 to a computer system or a modem.

In 'local' or 'remote' mode it can be used to load data to the Programmer's RAM or dump data from the Programmer's RAM to peripheral equipment.

Only 9 of the 25 available pin sockets play an active role in data transfer.

These are: numbers 1, 2, 3, 4, 5, 6, 7, 8 and 20.



Individual Pin Functions on the Serial I Connection

1.*	PG	PROTECTIVE GROUND
2.	TXD	TRANSMITTED DATA
3.	RXD	RECEIVED DATA
4.	RTS	REQUEST TO SEND
5.	CTS	CLEAR TO SEND
6.	DSR	DATA SET READY
7.	SG	SIGNAL GROUND
8.	DCD	DATA CARRIER DETECT
20.	DTR	DATA TERMINAL READY

*Pin number 1 is present in all connections, it represents the protective ground surrounding all the other cables.

RS232(2) SERIAL 2 Male connector

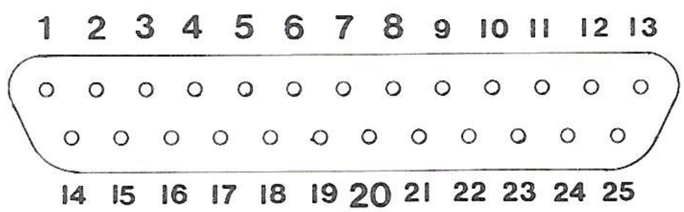
The RS232(2) port is a D-type male connector configured as DCE (Data Communication Equipment) which is able to accept a 25 way female connector (DTE).

It is generally used to connect the System 3000 to a terminal, a tape reader, etc.

In 'local' or 'remote' mode it can be used to load data to the Programmer's RAM or dump data from the Programmer's RAM to peripheral equipment.

Only 9 of the 25 available pins play an active role in data transfer.

These are: numbers 1, 2, 3, 4, 5, 6, 7, 8 and 20.



The interior connectors of the male UART (Universal Asynchronous Receiver Transmitter) are configured in such a way as to make it identical (apart from "pin 6 DSR" or DCE to DCE connection) to the exterior connections of the Female Port.

This is done by inverting the function of individual pins. For instance: transmit data (pin 2) becomes received data (pin 3) and vice-versa.

INDIVIDUAL PIN FUNCTIONS ON THE SERIAL 2 CONNECTOR

1.*	PG	PROTECTIVE GROUND
2.	TXD	RECEIVED DATA
3.	RXD	TRANSMITTED DATA
4.	RTS	DATA CARRIER DETECT
5.	CTS	DATA TERMINAL READY
6.	DSR	DATA SET READY
7.	SG	SIGNAL GROUND
8.	DCD	REQUEST TO SEND
20.	DTR	CLEAR TO SEND

*Pin number 1 is present in all connections, it represents the protective ground surrounding all the other cables.

CONNECTION TYPES

The System 3000 supports the two most popular types of connection these are: XON/XOFF (3 wire cableform connection) and (7/8 wire cableform) hardware handshake.

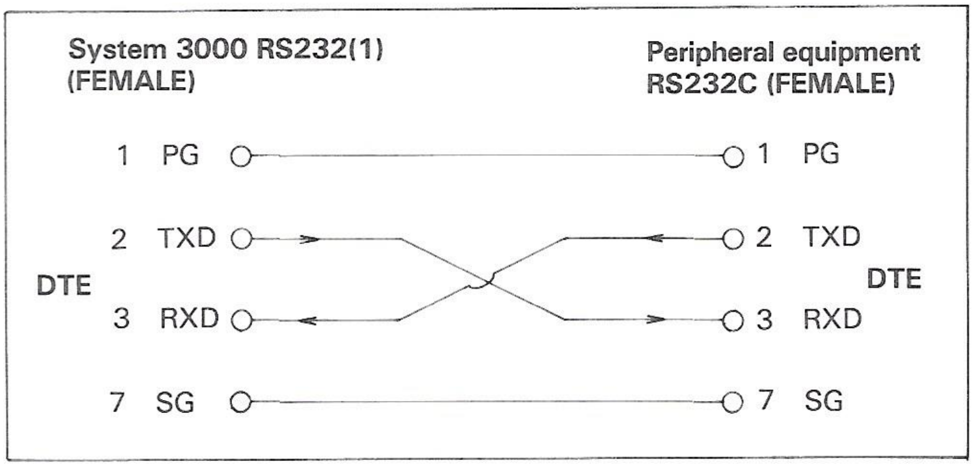
The most straightforward of these two is XON/XOFF.

16.2 XON/XOFF (3 wire cableform connection)

For connection of two alike machines a 'crossover' is needed.

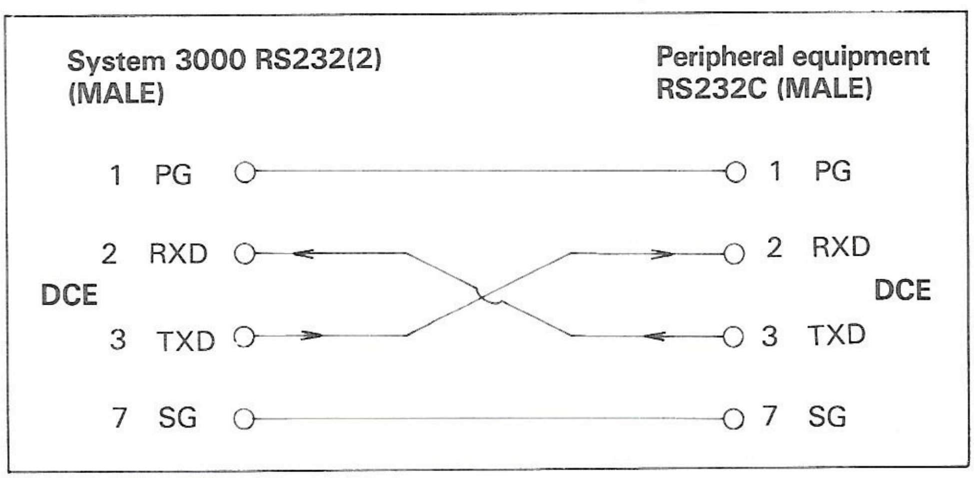
(a) DTE to DTE (crossover)

A 25-WAY MALE to MALE connector is required.



(b) DCE to DCE (crossover)

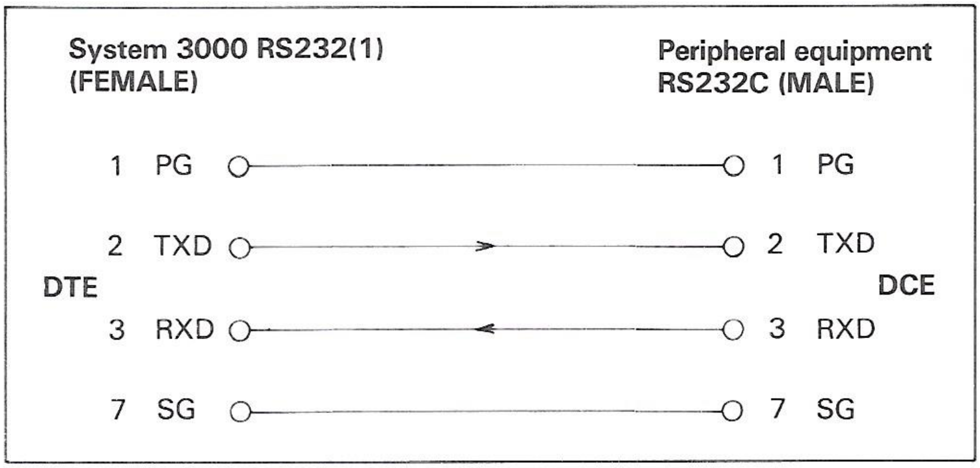
A 25-WAY FEMALE to FEMALE connector is required.



For connection of two unalike machines 'no' cross over is needed.

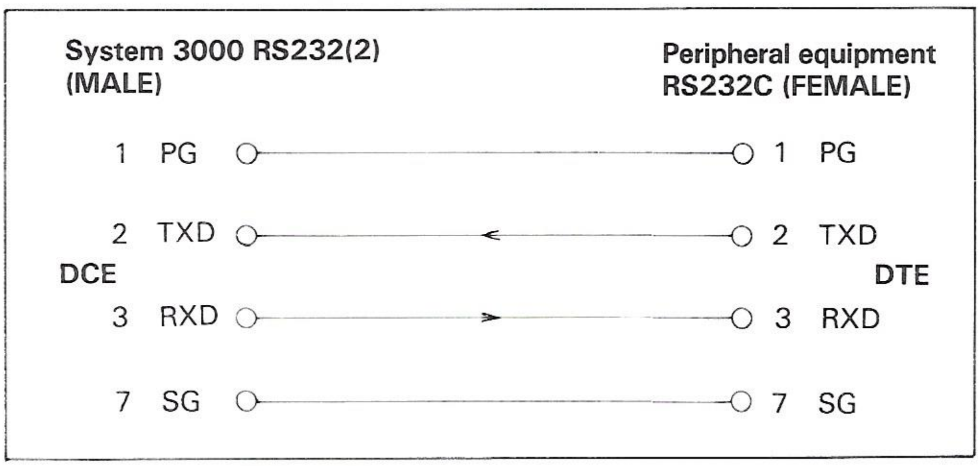
(c) DTE to DCE (straight)

A 25-WAY MALE to FEMALE connector is required.



(d) DCE to DTE (straight)

a 25-WAY FEMALE to MALE connector is required.



Note: Some machines do not have internal pull-ups and require extra connections within the cable form. Pull-ups may be required on pins 5, 6 and 8 of the external device if it is DTE or pins 4, 6 and 20 if it is DCE.

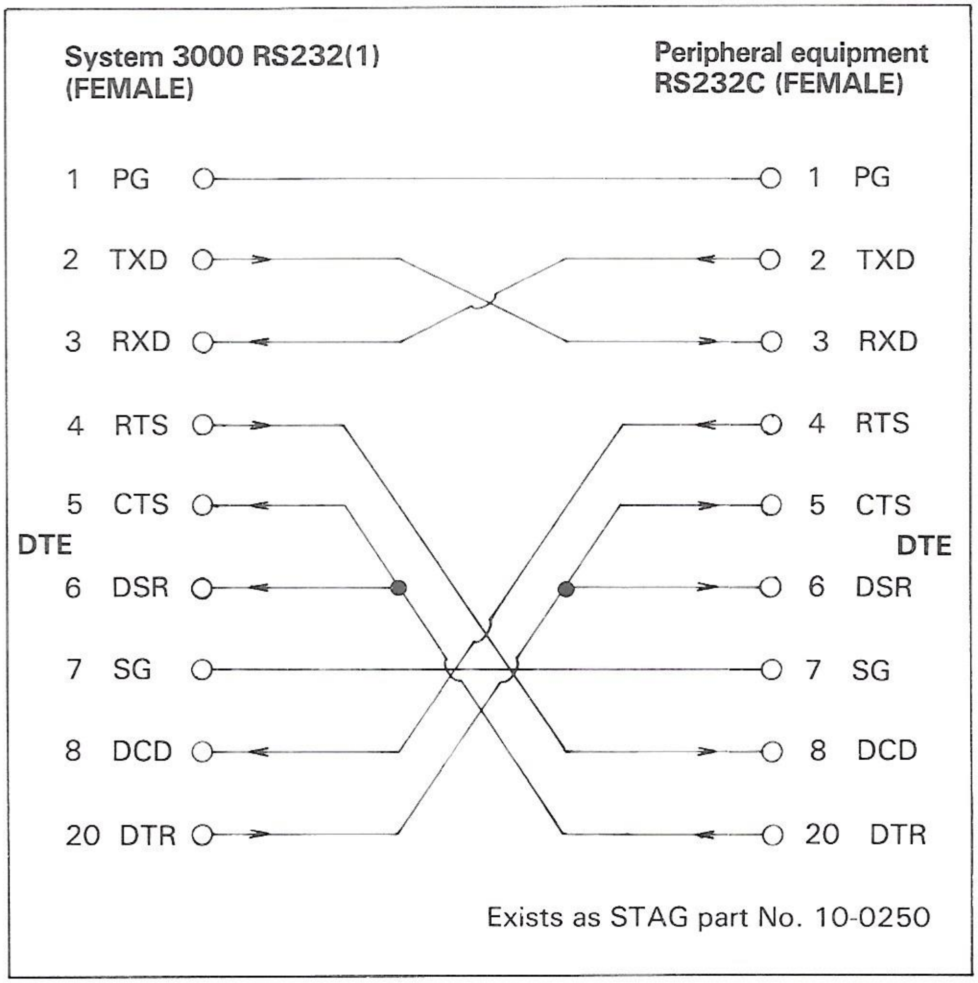
16.3 Hardware Handshake (7 or 8 Wire Cableform)

DTE to DTE and DCE to DCE	Crossover	8 wire cableform
DTE to DCE and DCE to DTE	Straight	7 wire cableform

For connection of two alike machines a 'crossover' is needed.

(a) DTE to DTE (crossover)

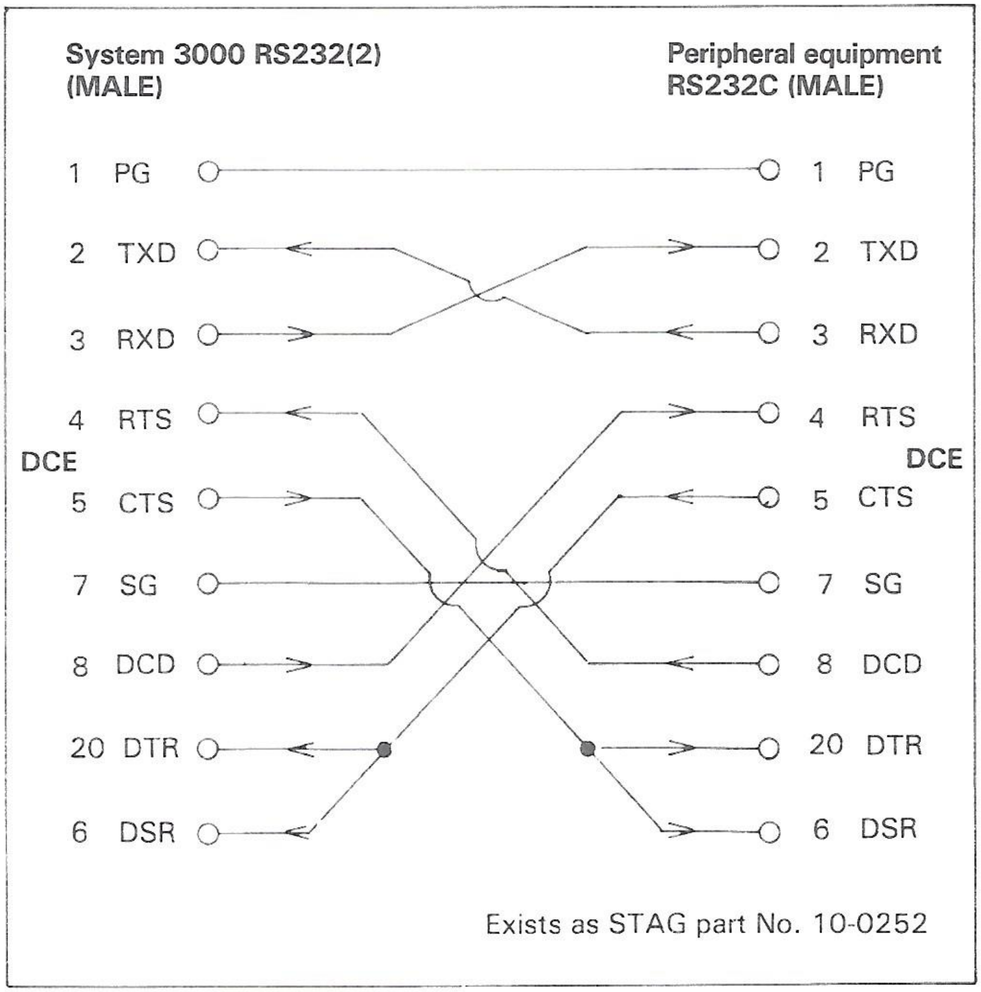
A 25-WAY MALE to MALE connector is required.



Exists as STAG part No. 10-0250

(b) **DCE to DCE (crossover)**

A 25-WAY FEMALE to FEMALE connector is required.



For connection of two unalike machines NO cross over is required.

(c) DTE to DCE (Straight)

A 25-WAY MALE to FEMALE connector is required.

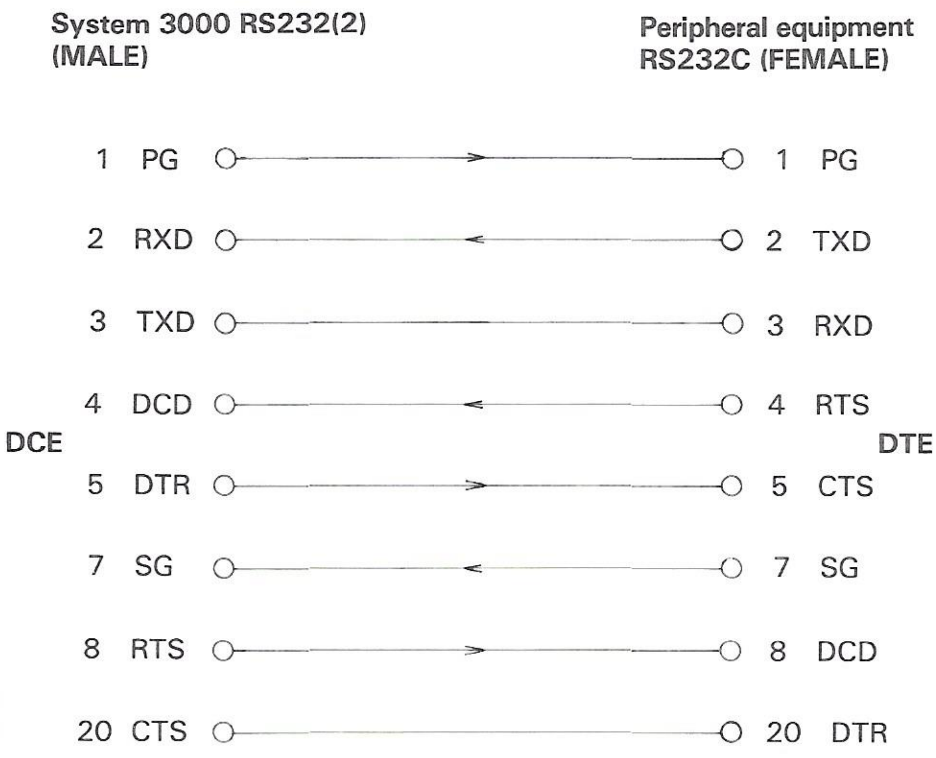
System 3000 RS232(1) (FEMALE) — DTE			Peripheral equipment RS232C (MALE) — DCE	
1	PG	———	1	PG
2	TXD	——→	2	RXD
3	RXD	←——	3	TXD
4	RTS	——→	4	DCD
5	CTS	←——	5	DTR
7	SG	———	7	SG
8	DCD	←——	8	RTS
20	DTR	——→	20	CTS

Note: Pin 6 on the straight version will be pulled up internally by the System 3000.

Exists as STAG part No. 10-0251

(d) **DCE to DTE (Straight)**

A 25-WAY FEMALE to MALE connector is required.



Note: Pin 6 on the straight version will be pulled up internally by the System 3000.

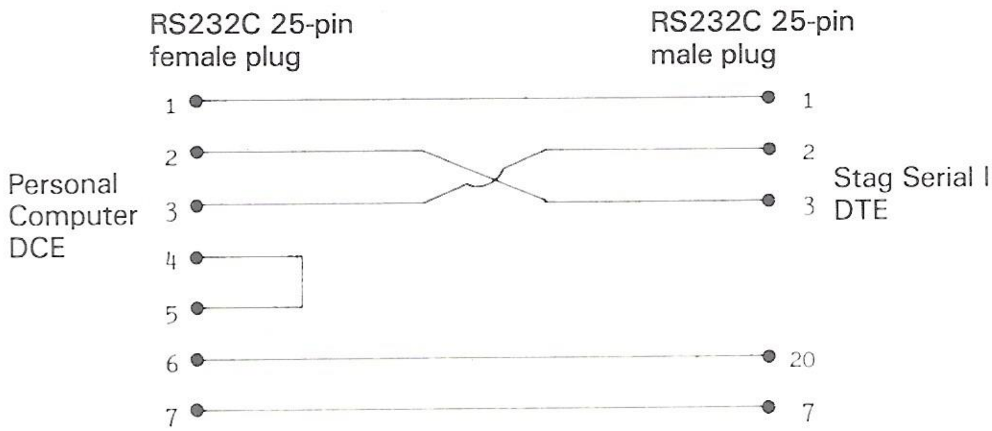
Exists as STAG part No. 10-0251

16.4 **Connection to IBM*PC and AT**

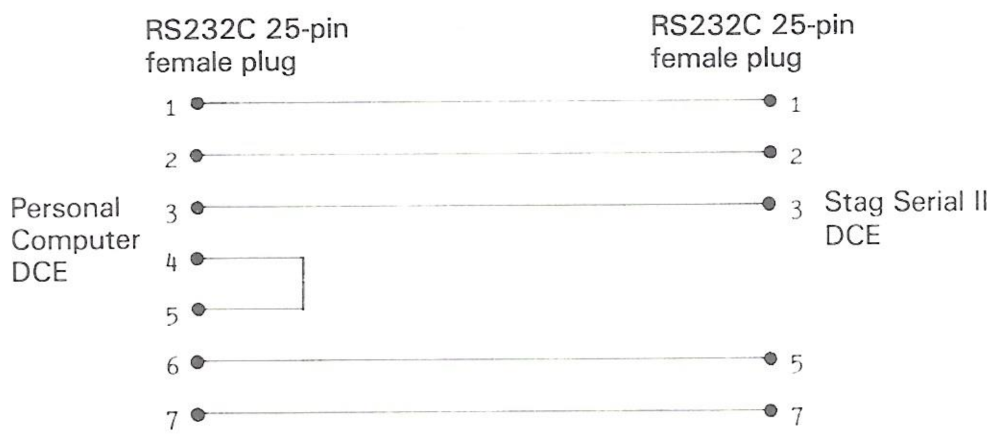
PC to Programmer RS232C interface connections

XON/XOFF

IBM-PC connected to the female (serial I) interface on a Stag Programmer.

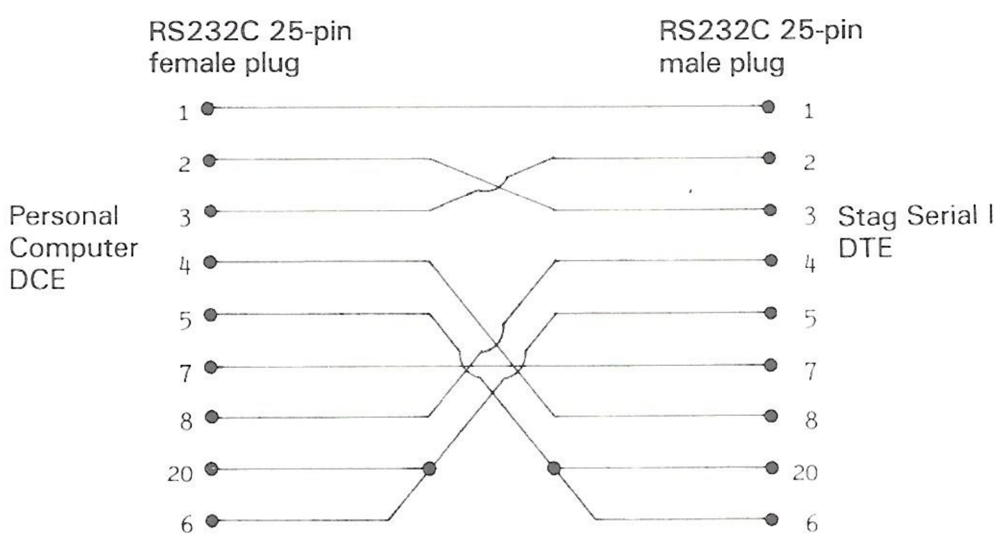


IBM-PC connected to the male (serial II) interface on a Stag Programmer.

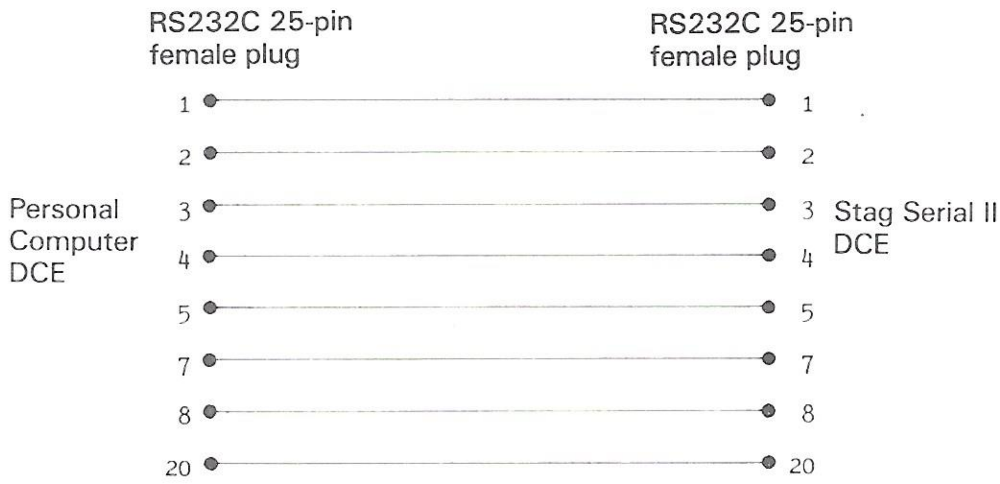


Hardware Handshake

IBM-PC connected to the female (serial I) interface on a Stag Programmer.

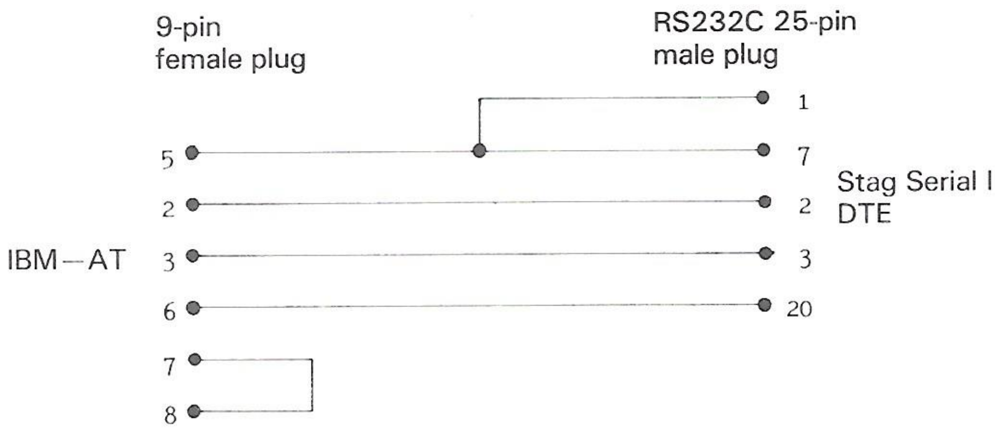


IBM-PC connected to the male (serial II) interface on a Stag programmer.

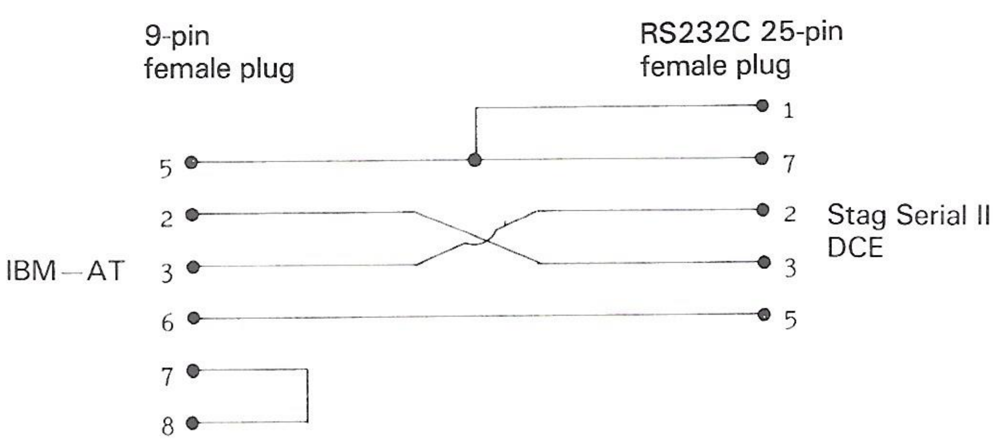


IBM—AT connections

IBM—AT connected to the female (serial I) interface on a Stag programmer



IBM—AT connected to the male (serial II) interface on a Stag programmer



SECTION 17

17

stag

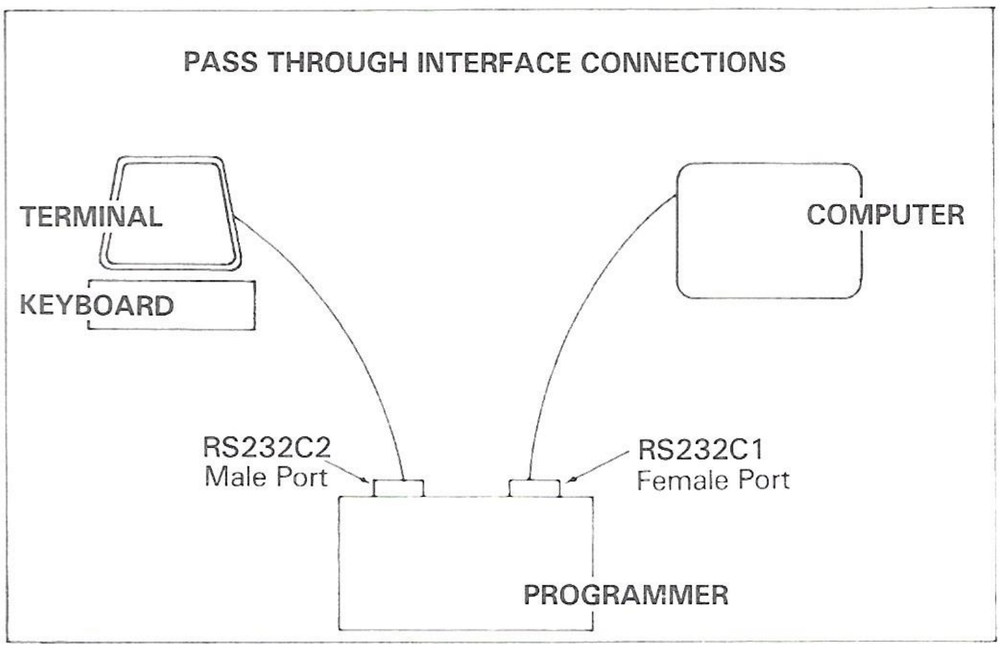
Sophisticated systems for the discerning engineer.

17 Pass Through

17.1 Pass Through

The PASS-THROUGH mode allows communication between a terminal, the programmer and a computer by utilizing only one connection on the terminal and the computer.

Note: In order for PASS-THROUGH to work the computer must be connected to the programmer via the RS232(1) female port (configured as DTE) and the terminal must be connected via the RS232(2) male port (configured as DCE).



In PASS-THROUGH the terminal has three modes of control.
These are:

1. **The terminal controls the computer alone.**

2. **The terminal controls the System 3000 via the computer.**

3. **The terminal can implement the INPUT and OUTPUT of data between itself and the System 3000 or the computer and the System 3000.**

Selection of a particular mode is dependent upon the configuration entered on the System 3000.

1. **The terminal can control the computer alone:** by selecting the system menu on the System 3000.
   When the ''system menu'' is entered on the System 3000 the programmer will be ignored by both the terminal and computer, and will act merely as a channel for data transfer, hence ''pass-through''. (The 'system menu' is the default state of the System 3000 when it is in LOCAL MODE).

2. **The terminal can control the System 3000 via the computer:** by selecting the remote control on the System 3000.

   By turning the keyswitch under the hinged flap to the ''left'' the REMOTE CONTROL can be entered on the System 3000.

   Terminal instructions can be passed through the programmer and sent to the computer. The computer can then motivate the programmer by returning the instructions to it.

   In effect the terminal will give the appearance of controlling the System 3000 on a one to one basis, i.e.: REMOTE CONTROL, with all instructions identical to those specified in the REMOTE MODE.

3. **The terminal can implement the input and output of data between itself and the System 3000 or the computer and the System 3000:** by selecting the I/O mode of the System 3000*.

   This can be shown by using an example, e.g. a data dump from the System 3000 to the terminal.

   To implement a data dump from the System 3000 to the Terminal, OUTPUT via the RS232C2, it must first be selected by entering into the System 3000 I/O mode.

   When these selections have been made pressing the ENTER key will put the System 3000 into a state of readiness for the dump.

   The necessary instruction to implement this dump is sent from the terminal through the programmer to the computer and back to the programmer.

   This will cause the System 3000 to 'dump' data (within preselected limits) back to the terminal along the same route.

   Therefore, selection of either load or dump between the programmer and the computer or the terminal must be made in the System 3000's I/O mode.

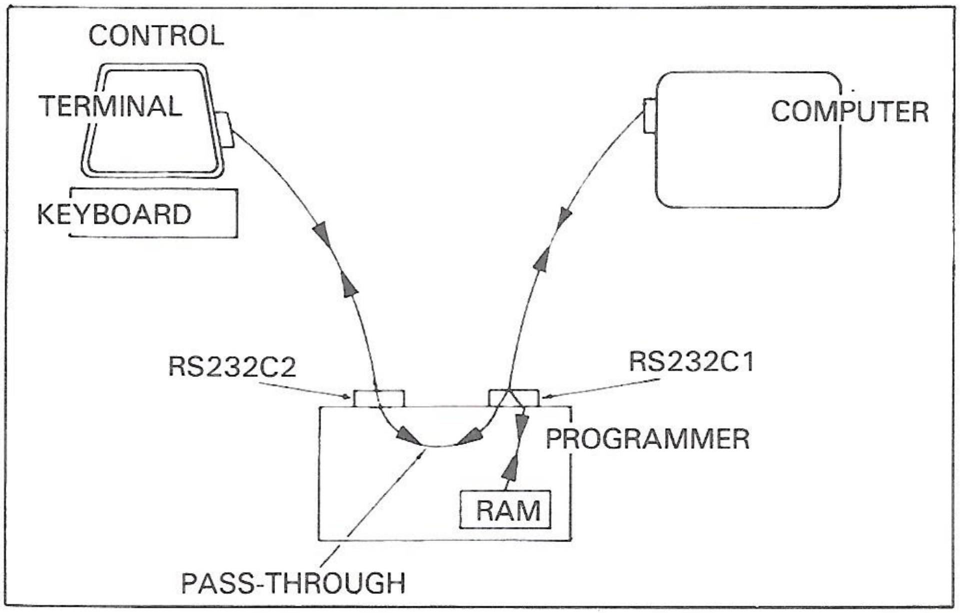
SELECTION OF INPUTS AND OUTPUTS IN THE PASS THROUGH MODE

INPUT RS232C1	Data is loaded from the computer to the System 3000
OUTPUT RS232C1	Data is dumped to the computer from the System 3000
INPUT RS232C2	Data is loaded from the terminal to the System 3000
OUTPUT RS232C2	Data is dumped to the terminal from the System 3000

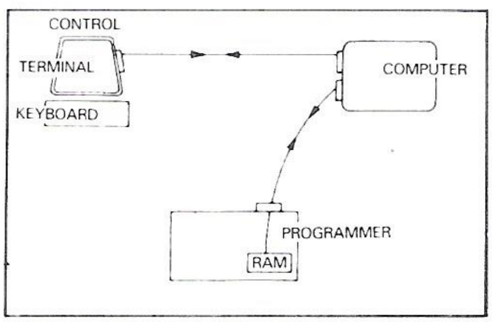
*If required address limits and offsets can also be entered in the System 3000's I/O mode.

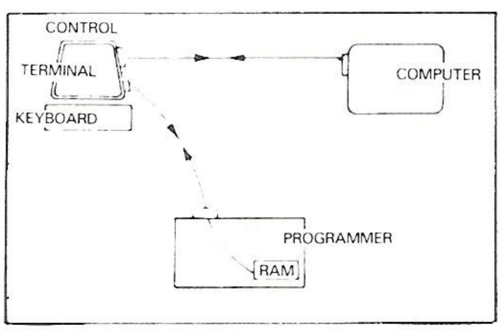
A data file being dumped to the System 3000 in PASS-THROUGH must be pre-fixed with a control—O (ASCII code—OF). This switches the data into the programmer RAM.

OPERATING STRUCTURE OF PASS THROUGH



In effect PASS-THROUGH duplicates the operating structures as shown below.

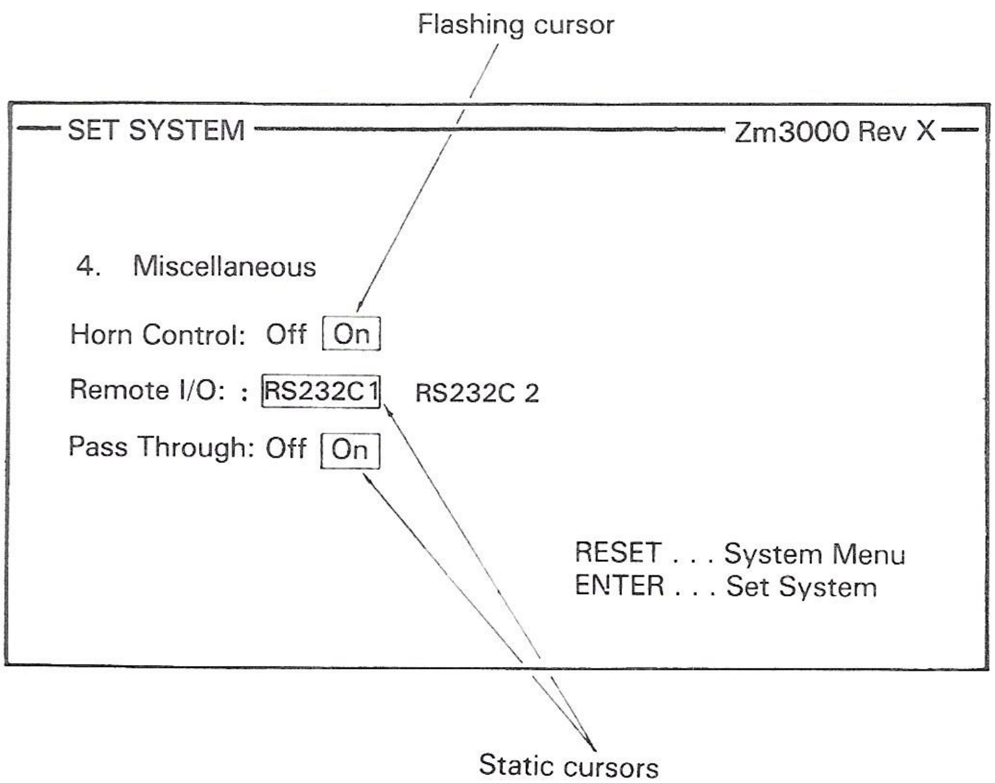




SELECTING "PASS THROUGH"

Pressing key . . . A from the system menu followed by key . . . 4 "Miscellaneous" will show a flashing cursor defaulted to the 'horn control' and two static cursors indicating the last used mode of operation (outside local mode). This could be either 'pass through' itself or 'remote mode' through one of the three displayed interface ports, i.e. RS232C 1, RS232C 2 or IEEE.

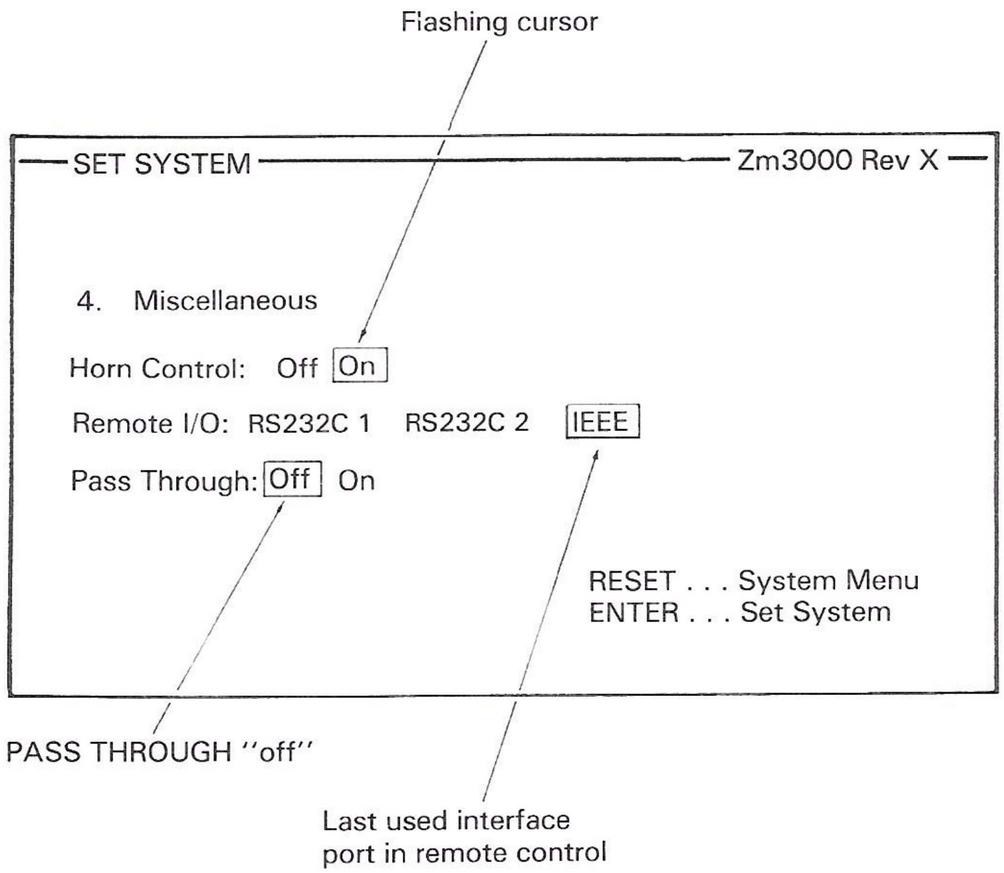
If the last used mode was 'pass through' the display will show:



Note: PASS-THROUGH will always default to the RS232C 1 port.

If the last used operating mode was REMOTE CONTROL the static cursors will show (i) the PASS-THROUGH "off" and (ii) the last interface port used in REMOTE CONTROL.

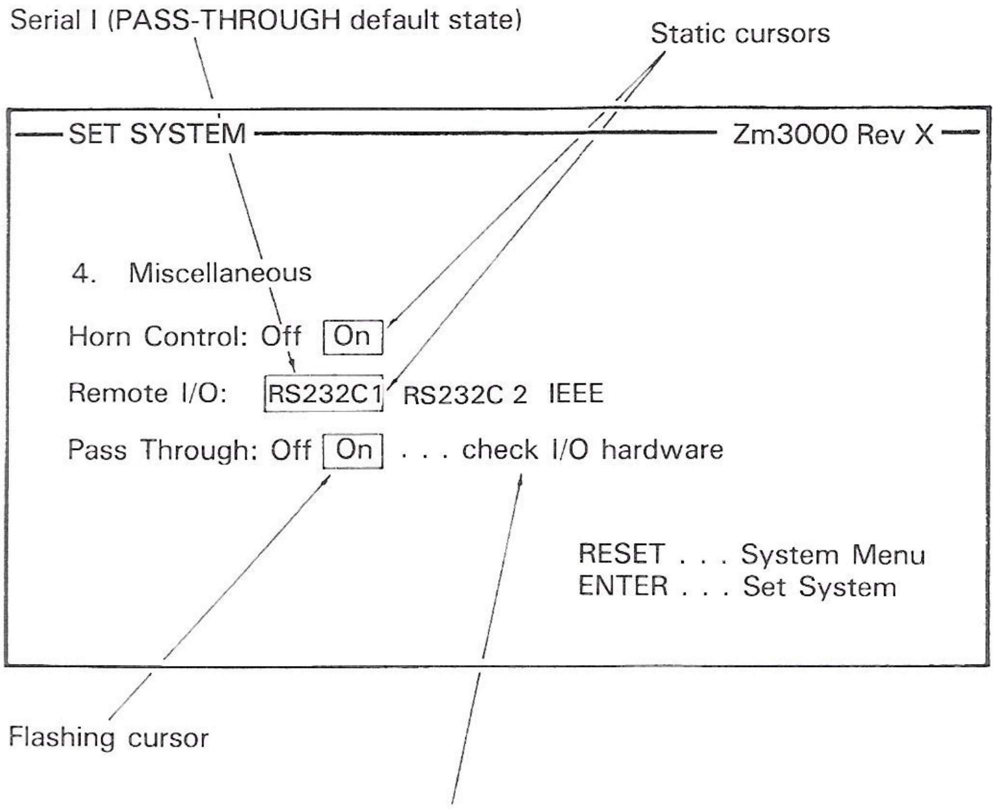
For example:



Selection of PASS-THROUGH can now be made by pressing the down cursor key twice which will transfer the flashing screen cursor onto "pass through: off".

By pressing the right cursor key the System 3000 will be configured for operation in the PASS-THROUGH mode, i.e. "on" indicated once again by the flashing cursor. Simultaneously the middle screen cursor will default to RS232C 1 if it was previously indicating one of the other two ports.

For example:



The display will also show a visual reminder to check that the System 3000's 'I/O hardware' i.e. Baud rate, Parity, etc., correlate for both serial interface ports during PASS-THROUGH.

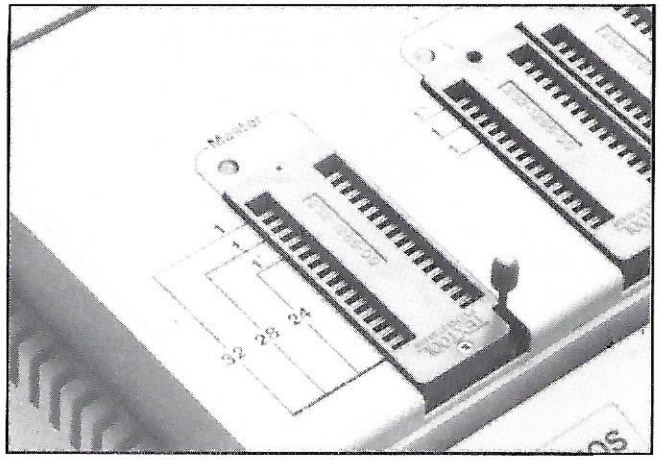
This can be checked or altered by pressing the ENTER key followed by key . . . 1.

The System 3000 is now configured for operation in the PASS-THROUGH mode.

Correct Operation of ZIF Sockets.

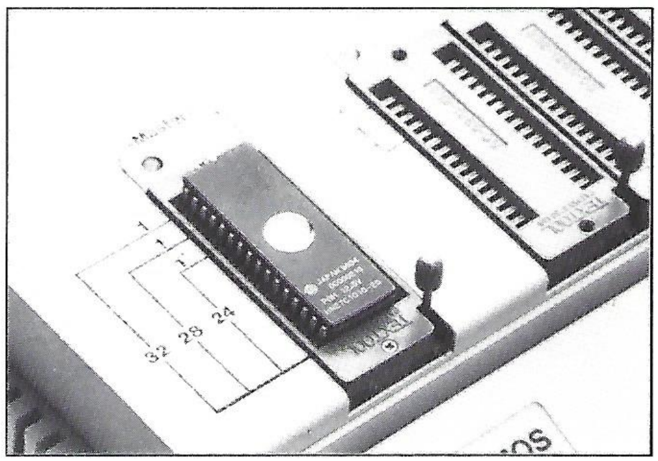
1. Empty ZIF socket.

   Lever in open (up) position.



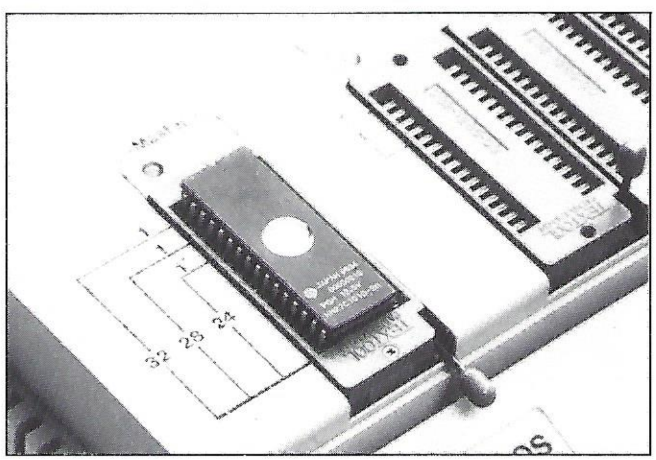
2. Device inserted.

   Socket open.



3. Device inserted.

   Lever in closed (down) position.



SPECIAL OFFER

CUPL: programmable logic compiler.

A special offer is available to the customer for CUPL a high level programmable logic compiler.
For further information and a quotation, please fill in and return this voucher.
Please indicate the type of floppy disc operating system that you are interested in.

- ☐ CU 1020 PC-DOS 2.1 (5¼")
- ☐ CU 1030 MS-DOS 2.1 (5¼")

Programmer:......................................

Purchase order Number:.......................

Date of purchase:..............................

Name:...

Title:..

Company:......................................

Address:..

City:...

State:...

Tel:.................................Ext:..........

CUPL is a tradename of Personal CAD Systems

TELL ME MORE!

ADDITIONAL TECHNICAL INFORMATION IS AVAILABLE FOR THE FOLLOWING:
(Please tick the appropriate box)

- ☐ PPZ Universal Programmer
- ☐ PP39 Portable MOS Programmer
- ☐ PP40 Ganged Copier
- ☐ PP41 Ganged Programmer
- ☐ PP42 Gang/Set Programmer
- ☐ Stag Com 1: Software communication package for PROM programmers
- ☐ Stag Com 2: Software communication package for logic programmers.
- ☐ ZL30 Logic Programmer
- ☐ ZL30A Logic Programmer
- ☐ ZL33 Ganged Logic Programmer
- ☐ CUPL: Programmable Logic Compiler
- ☐ Stag Erasers
- ☐ Please have a sales engineer call me.

Name:...

Title:..

Company:......................................

Address:..

City:...

State:..............................Zip:..........

Tel:.................................Ext:..........

TELL ME MORE!

ADDITIONAL TECHNICAL INFORMATION IS AVAILABLE FOR THE FOLLOWING:
(Please tick the appropriate box)

- ☐ PPZ Universal Programmer
- ☐ PP39 Portable MOS Programmer
- ☐ PP40 Ganged Copier
- ☐ PP41 Ganged Programmer
- ☐ PP42 Gang/Set Programmer
- ☐ Stag Com 1: Software communication package for PROM programmers
- ☐ Stag Com 2: Software communication package for logic programmers.
- ☐ ZL30 Logic Programmer
- ☐ ZL30A Logic Programmer
- ☐ ZL33 Ganged Logic Programmer
- ☐ CUPL: Programmable Logic Compiler
- ☐ Stag Erasers
- ☐ Please have a sales engineer call me.

Name:...

Title:..

Company:......................................

Address:..

City:...

State:..............................Zip:..........

Tel:.................................Ext:..........

AFFIX
STAMP
HERE

STAG MICROSYSTEMS INC.
1600 WYATT DRIVE, SUITE 3
SANTA CLARA, CA 94054

AFFIX
STAMP
HERE

STAG MICROSYSTEMS INC.
1600 WYATT DRIVE, SUITE 3
SANTA CLARA, CA 94054

AFFIX
STAMP
HERE

STAG MICROSYSTEMS INC.
1600 WYATT DRIVE, SUITE 3
SANTA CLARA, CA 94054

GUARANTEE/REGISTRATION: Please return within 14 days.

Name:.. Title:..

Address: ....

City:...

State:.. Zip:..

Telephone Number:............................ Ext:..

Programmer Type:.............................. Serial Number:...............................

Module Type(s):................................ Serial Number(s):............................

Date Received:...................................

WARRANTY: ONE YEAR

Stag Microsystems Inc. warrants that all equipment sold pursuant to any resultant agreement shall be free from Defects in Material or Workmanship at the time of delivery. The buyer must provide notice to Stag within this prescribed Warranty Period of any Defect; if the Defect is not the result of Improper Usage, Service, Maintenance or Installation and equipment has not been otherwise damaged or modified after delivery, Stag shall either replace or repair the defective part or parts of equipment or replace the equipment at Stag's option, after return of such equipment by buyer to Stag. Cost of shipment to and from Stag's facility shall be borne on account of the buyer.

MAINTENANCE SERVICE

An annual maintenance service is available outside the warranty period of one year. For further information and a quotation, please fill in and return this voucher.

Name:.. Title:..

Address: ....

City:... State:................. Zip:......................

Telephone Number:............................ Ext:..

Programmer Type:.............................. Serial Number:...............................

Module Type(s):................................ Serial Number(s):............................

Date Received:...

803 3007 Rev. 2

AFFIX
STAMP
HERE

STAG MICROSYSTEMS INC.
1600 WYATT DRIVE, SUITE 3
SANTA CLARA, CA 94054

stag

AFFIX
STAMP
HERE

STAG MICROSYSTEMS INC.
1600 WYATT DRIVE, SUITE 3
SANTA CLARA, CA 94054