-XELTEK

764 San Aleso Ave., Sunnyvale,
CA 94086 U.S.A.
Tel: (408) 745-7974, Fax: (408) 745-1401

UNIPRO

Universal Programmer

Reference Manual

XELTEK

Copyrights

764 San Aleso Ave. Sunnyvale, CA 94086
TEL(408)745-7974 FAX(408)745-1401

PAREIL AU HI LO ALL-01 ?
CARTE d'INTERFACE CSA-02 ?

XELTEK UNIPRO INTERFACE SOFTWARE

UNIPRO Software V 3.0 Release notes

Introduction

Version 3.0 includes LOPS (Library-Operated Programming System). This has two versions: the standard version and the advanced version. The standard version will let the users vary the programmer's algorithm and voltages. The advanced version will let the users create or copy and update the library for the chips which are not supported.

The features of PAL and GAL programs include support for JEDEC files created by Tango-PLD and an entirely new user interface. Both programs as well as LOPS feature a pull-down menu interface which allow operation of functions from the menus, by typing in the key letter of each command. The on-screen prompt lines assist your selection of options.

Attached to these release notes are newly illustrated chapters for the UNIPRO reference manual which fully document the operation of the new PAL and GAL programs.

Files on the Release Disk(s)

BPR.EXE
Menu-driven routine used to program Bipolar PROM devices.

GAL.EXE
Menu-driven routine used to program GAL devices.

ICT.EXE
Menu-driven routine used to test TTL, CMOS, and memory devices.

PAL.EXE
Menu-driven routine used to program PAL devices.

LOPS.EXE
Menu-driven routine used to program E(EPROM's nd single-chip microcontrollers.

TTL.EXE
Stores IC test patterns. This must be loaded to test chips using ICT program.

CORP.INX
Manufacturer index file

CORP.DTA
Manufacturer data file

EPRS.INX
Manufacturer index file

EPRS.DTA
Manufacturer data file

> NOTE: Even though they are not listed in here there are some data files which will be generated after the execution of each program.

Contents

GAL PROGRAMMING **Chapter 5**

BIPOLAR PROM PROGRAMMING Chapter 6

LOGIC IC & RAM TESTING Chapter 7

1 Introduction

Congratulations on your purchase of the UNIPRO universal programmer! To protect your investment in the UNIPRO hardware and software, be sure to fill out and return the Registration Card enclosed with the package. Registration is required for warranty service and notification of software updates.

1.1 What Is the UNIPRO?

The UNIPRO is an affordable, reliable, and extremely fast universal device programmer. The UNIPRO is designed to operate with the IBM-PC/XT/AT/386 series of computers and their compatibles. The UNIPRO's menu-driven, interface software makes loading, editing, and downloading files easy to use.
The programming hardware includes the following features:

- The sleek, low-profile programming module includes a 40-pin ZIF socket, for programming and testing 8-pin to 40-pin devices, a "busy" LED indicator, and a "good" LED indicator.
- The UNIPRO's high-speed, parallel interface card, when plugged into your PC expansion slots, speeds the uploading and downloading of files.
- Device models that are PC-resident in software permits easy updates by diskette.
- Protection for short circuits
- Hardware timer in the programming module ensures correct programming time.

- Four E(E)PROMs may be programmed serially with optional four socket adapter. This saves time for mounting one by one.

The UNIPRO interface software includes these features:

- An easy-to-learn user interface, including pull-down menus, selection and entry boxes, and on-line help prompts.
- Integrated text editors for modifying JEDEC files and test vectors.
- Programming support for a large list of devices, including the most popular manufacturers of PROMs, EPROMs, EEPROMs, PALs, PEELs, PLDs, EPLDs, and GALs. IC testing capability for TTL, CMOS, and memory devices.
- The contents of E(E)PROMs, PALs, GALs, Bipolar PROMs, and 8748/51 series single-chip CPU's, can be read and saved onto disk.
- Displayed CHECKSUM and BLOWCOUNT assures correct programming of parts.
- Vector testing is available to test functions for PAL devices.
- Intel (extended) HEX, Motorola S1, S2, and S3, and Tektronix (extended) format files may be loaded, and are automatically converted to BIN format.

This document explains how to install and use your UNIPRO programmer and interface software. We hope you will enjoy using our products, and that they will help you create high-quality, programmed devices quickly and efficiently.

1.2 Manual Organization

The UNIPRO Reference Manual consists of three main parts: Introduction, User's Reference, and Appendices.

INTRODUCTION
Chapters 1 - 3
An introduction to the UNIPRO, including features, system requirements, and set-up of the hardware and software.

USER'S REFERENCE
Chapters 4-9
Detailed operating instructions for each of the software routines are used to program devices on the hardware.

APPENDIXES
Appendixes A-C
Information on the JEDEC file format, XELTEK customer support, and self maintenance.

1.3 Manual Conventions

The following conventions are used in this manual:

1. The names of all non-alphanumeric keys and key sequences are enclosed in angle brackets < >. For example, the Enter (or Return) key is shown as <Enter>; the PgUp (Page Up) key is shown as <PgUp>.

   The cursor keys are shown as follows:
   Left arrow key = <LeftArrow>
   Right arrow key = <RightArrow>
   Up arrow key = <UpArrow>
   Down arrow key = <DownArrow>

2. All text that appears on the screen is shown in bold characters. These characters are either manually typed or are displayed by the program. For example:

   gal

3. The keystroke **<Enter>** is pressed at the end of each DOS command and will not be shown in the manual.

4. Unless stated otherwise, keystrokes are not case-sensitive. This means that you can enter either upper-case or lower-case characters. For example, if you are asked to type the letter A, you can type either **A** or **a**.

5. The manual refers to a menu command name using the menu name first, followed by the command name as listed in the menu box. For example, if the manual directs you to select the **Load** command in the **File** menu, it will simply say to select the File Load command.

   The manual includes WARNING and NOTE paragraphs, which are indented and printed in *italics*. A WARNING refers to information that can prevent damage to hardware or software, including irretrievable loss of your data files. A NOTE contains additional topic information and recommendations.

2 System Requirements

2.1 Hardware and Software Requirements

The minimum hardware requirements for using the UNIPRO hardware and software are as follows:

- **IBM-PC/XT/AT/386 computer or compatible.**
- **Open Slot.** A minimum of one half-length slot for inserting the parallel interface card.
- **640 Kbytes RAM.**
- **MS-DOS or PC-DOS, Version 2.1 or greater.**
- **One Disk Drive.** One diskette drive (360Kbyte, 720Kbyte, 1.2Mbyte or 1.44Mbyte) for copying and saving program and data files, and optionally one hard disk drive for operation.
- **Color or Monochrome Graphics Adapter and Monitor.** The UNIPRO software operates in text mode and supports all graphics devices, color, or monochrome.

2.2 DOS Considerations

This manual assumes you are familiar with the Disk Operating System (DOS) of the computer. Standard operations such as disk formatting, copying files and disks, listing directory files, and so on, must be understood to be able to run the UNIPRO software. You may wish to refer to a DOS manual, your computer's operating manual, and other tutorial and reference documentation on the set-up and use of your computer.

2.3 The UNIPRO Package

Your UNIPRO package contains the following:

- The UNIPRO programming module.
- The half-slot parallel interface PCB card.
- A parallel interface connecting cable.

- The UNIPRO Reference Manual.
- The UNIPRO software diskettes.
- One Registration Form. This form should be filled out and returned as soon as possible. **Registration is required for warranty service and notification of software updates.**
- One Device Update Coupon. Use this coupon to request support for new programmable devices.

2.4 Files on the Release Disk(s)

Please refer to the Release Notes included with this manual for a listing and description of files on the release diskettes.

3 Getting Started

3.1 A Checklist

The following checklist shows the steps required to run the UNIPRO programmer. Each step is explained in detail in this chapter.

- Make backup copies of the UNIPRO software diskettes.
- Install the UNIPRO software in a directory on the hard disk.
- Install the UNIPRO interface card properly in your personal computer.
- Connect the programming module to the installed interface card with the provided cable.
- Power up the programming module.
- Run the UNIPRO software routines to load, edit, and download files to the programmable device.

3.2 Creating Backup Copies

We strongly recommend that you make backup copies of the UNIPRO diskettes, then use these backup diskettes to install the software. Computer diskettes are not indestructible. They can incur damage from any number of calamities, such as drive problems, mishandling, coffee spills, and curious pets.

All UNIPRO programs and files are stored as standard DOS files. There are no copy-protection schemes such as hidden files or corrupt sectors on the diskettes containing the software. It is a simple matter to make any number of backup copies using the DOS COPY command.

To backup your diskettes, first format an equal number of new diskettes. Now copy the files on each UNIPRO diskette to a newly formatted diskette. Label the backup copies with the same names given to the original diskettes, and use them as your working copies of the software. Store the original UNIPRO diskettes in a cool, dry place, away from magnetic fields and direct sunlight.

3.3 Installing The Hardware

The UNIPRO universal programming hardware consists of a desktop programming module with 40-pin ZIF socket, a parallel interface card, which fits in a short slot in your computer, and a parallel interface cable to connect the programmer to the interface card. The programming module derives its power from the PC via the parallel interface; thus, no additional power cord is present.

The first step is to install the interface card in your PC. Turn your computer off and carefully remove its cover. Check the documentation with your computer and the interface cards to be sure that the port address set in the UNIPRO interface card does not conflict with any existing I/O devices. The default setting from the factory is at jumper position #5 which corresponds to port address 280-29F.

If an address change is necessary, relocate the jumper position according to the following table:

Jumper Position	I/O Address (Hex)
#1	200-21F
#2	220-23F
#3	240-25F
#4	260-27F
#5	280-29F (Default)
#6	2A0-2BF
#7	2C0-2Df
#8	2E0-2FF

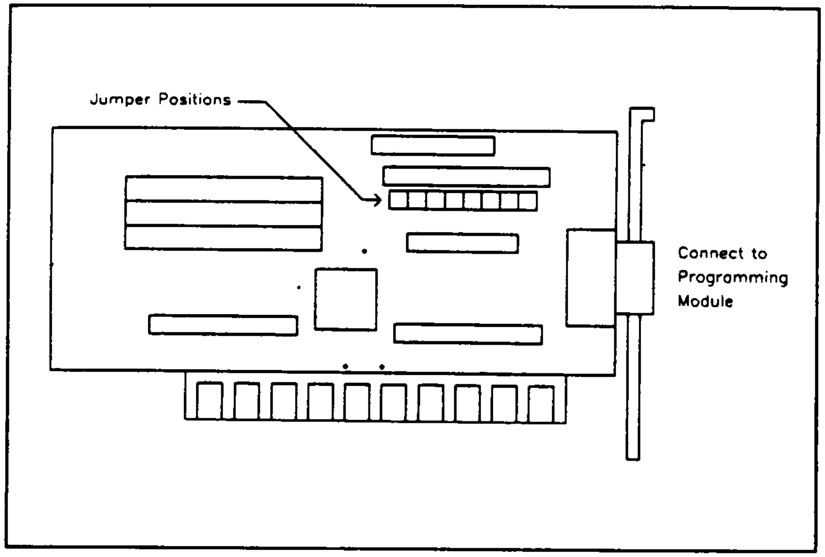


Figure 3-1. The UNIPRO parallel interface card.

If you changed the jumper position on the interface card, you may have to change the port address information within each of the UNIPRO software routines. When you first run the GAL program, for instance, a communication error will be issued because the software is set up initially to access the programming module via the default address, 280-29F. Each of the UNIPRO routines includes a command to reset the port address, which should resolve the problem.

Once you have the jumper set as desired, carefully insert the card into an open slot in your computer, screw the bracket down securely to your computer's chassis, and replace the cover.

Now connect the parallel interface cable to the programming module on one end, and connect the parallel interface card connector to the back of your computer. Now turn on your computer, then the programmer. You are now ready to begin!

3.4 Installing The Software

We recommend a hard disk installation to speed access to your files. After loading DOS, make sure that drive C (the hard disk) is the default device. The following prompt should be displayed:

```
C>
```

Move to the directory of your choice, and copy all the files on the UNIPRO diskettes into that directory. The following procedure details how to copy the UNIPRO files to a hard disk directory called **unipro**. First, make sure that the current directory on the hard disk is the root directory. Type:

```
cd\
```

To create the directory UNIPRO, type:

```
md unipro
```

Now change the current directory to the new UNIPRO directory by typing:

```
cd unipro
```

With each UNIPRO diskette successively inserted in drive A, copy all the files from the diskette to the directory UNIPRO by typing:

```
copy a:*.*
```

3.4.1 Setting The DOS Path

By setting the DOS PATH variable (described in detail in the DOS Manual), you can run the UNIPRO software from any directory. You do not have to be in the \UNIPRO directory to execute the program. This makes it possible to set up a separate directory for each schematic design.

To set the DOS path, assign the pathname **\UNIPRO** to the PATH variable. This is usually done in the AUTOEXEC.BAT file, so you do not have to set the path each time you turn the computer off and on. A typical PATH variable assignment is shown below:

```
PATH=C:\DOS;C:\UTIL;C:\UNIPRO
```

In this example, whenever a program or batch file is invoked, DOS searches for it in the current directory. If the operating system can't find the program or batch file in the current directory, it searches all directories listed in the PATH setting.

3.5 Running The Software

In general, you should always turn ON the UNIPRO programming module *before* starting any of the software routines, and turn the module OFF *before* quitting the routines and returning to DOS.
To run the UNIPRO software for programming bipolar PROM , type:

```
bpr
```

To run the UNIPRO software for programming GAL , type:

```
gal
```

To run the UNIPRO software to test TTL, CMOS and memory devices, type:

```
ict
```

To run the UNIPRO software for programming PAL , type:

```
pal
```

To run the UNIPRO software for programming single chip microcontrollers and EPROMs, type:

```
iops
```

3.6 Possible Errors

A communication error may also occur if a device is locked in the programmer's ZIF socket when invoking any of the UNIPRO software routines. Just be sure the socket is empty when invoking any of the routines.

Once the menu is on the screen, you can lock socket by pulling down the lever.
As noted above, a communication error may also occur if you have changed the port address on the interface card. Match the port address within the UNIPRO software routine.

Warning

Please be sure to turn on your programmer after the computer is on to avoid possible damage.

4 PAL Programming

4.1 Introduction

The UNIPRO may be used to program a wide range of popular Programmable Array Logic (PAL), Programmable Logic Array (PLA) and Programmable Electronically Erasable Logic (PEEL) devices. A menu-driven software program provides an easy interface for loading, editing, and downloading JEDEC files to the device programmer.
Files on the release disk used for PAL, PLA and PEEL programming are:

PAL.EXE The program.

PAL.DAT A configuration file which is written to disk each time you quit the PAL program, and keeps a record of options selected. The file is read by the PAL program when it is re-started so that previously selected options are restored.

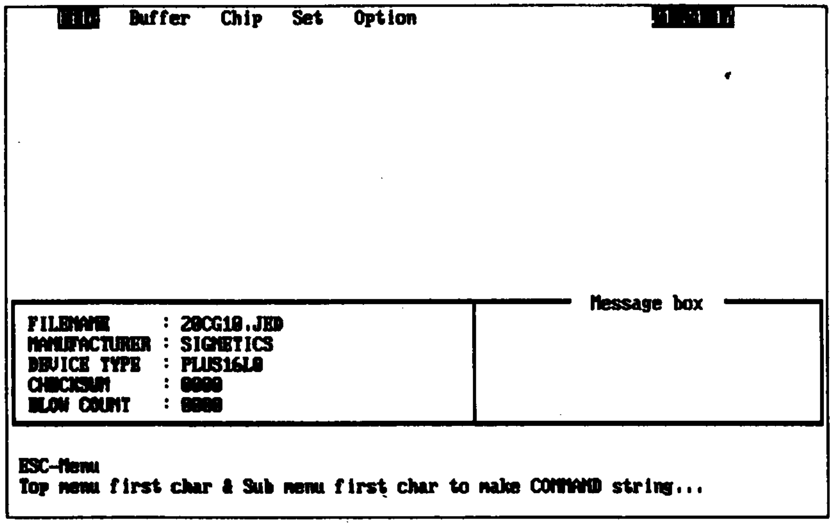
4.2 Programming PALs and PEELs

To program PALs, PLAs and PEELs, first be certain that the UNIPRO hardware is properly installed, and that the programmer is turned on. To start the program, type:

```
pal <Enter>
```

4.3 The Main Screen

Upon executing the program, the main screen will appear, as shown in figure 4-1. (If an error occurs at this point, refer to the troubleshooting section below.) The user interface is the pull-down menu variety, with options either being selected from the menus or by typing in a command's key letter on the keyboard. We'll briefly describe the information provided on the main screen.



```
FILENAME     : 20CG10.JED
MANUFACTURER : SIGNETICS
DEVICE TYPE  : PLUS16L8
CHECKSUM     : 0000
BLOW COUNT   : 0000

ESC-Menu
Top menu first char & Sub menu first char to make COMMAND string...
```

Figure 4-1. The PAL program's main screen

4.3.1 The Menu Line

Across the top of the screen is the menu line, which presents the main menu options, File, Buffer, Chip, Set and Option. Press <Esc> to access the menus. Then use the cursor (arrow) keys on your keyboard move the cursor from one option to another. Then press <Enter> to select that option. A pull-down menu will appear with sub-menu options. On the right side of the menu line is clock which displays the time of day based on your system's clock.

4.3.2 The Status Box

In the lower left corner of the screen is the status box. Displayed in the status box are:

FILENAME Name of the file currently in the memory buffer being edited or downloaded.

MANUFACTURER Manufacturer of the device currently selected for programming.

DEVICE TYPE The part name of the device currently selected for programming.

CHECKSUM The checksum on the fuse map portion of the file currently in the memory buffer.

BLOW COUNT The number of fuses blown or to be blown on the device.

4.3.3 The Message Box

In the lower right hand corner of the screen is the message box. Messages are displayed here during various operations of the program to indicate their status and ultimate completion.

4.3.4 The Input Line

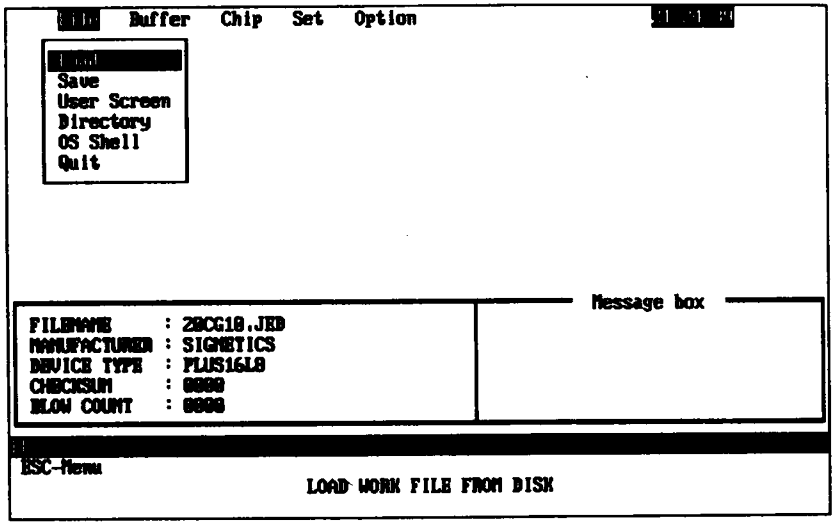
As indicated above, commands may be selected from the pull-down menus, or by typing their key letters on your keyboard. In either case, as the commands are selected, their key letters are displayed on the input line, which is located just below the status box.
Once you have learned the various commands available in the PAL program, you may find it faster to type in a sequence of key letters, which will be displayed on the input line. Then simply press <Enter> and the sequence of commands will be selected and run without pulling down the menus.

4.3.5 The Prompt Line

The last line, at the bottom of the screen, is the prompt line. As each menu option is selected from the menus, the prompt line gives a brief explanation of its purpose and operation. Again, once you are familiar with the program, the prompt line may be all the help you'll need.

4.4 The File Menu

The File menu contains five commands for loading and saving JEDEC files, listing a directory of files on disk, temporarily exiting the program to DOS, and quitting the program.



```
FILENAME     : 20CG10.JED
MANUFACTURER : SIGNETICS
DEVICE TYPE  : PLUS16L8
CHECKSUM     : 0000
BLOW COUNT   : 0000
```

LOAD WORK FILE FROM DISK

Figure 4-2. The File menu

4.4.1 File Load

This command is used to load a fuse map file from disk into the memory buffer. The program will prompt for the filename. The program accepts fuse maps in JEDEC format, which is the standard output of a wide variety of software available for designing PLDs. The UNIPRO supports JEDEC files created by all popular PLD design programs, including ABEL, CUPL, PALASM, Tango-PLD and OrCAD-PLD.

For further information on the JEDEC standard, refer to Appendix C.

4.4.2 File Save

The File Save command permits you to save the fuse map file which is currently in the memory buffer back to disk. This is important, since the PAL program includes editing functions which allow you to make changes in a fuse map you have loaded from disk, read from a previously programmed component, or even created new from scratch. Files are always saved in JEDEC format and include both the fuse map plus any test vectors you may have created.

4.4.3 File User screen

Use this command to view the screen, which can be saved just before the PAL software is invoked. Press <ENTER> to view the previous screen, and hit any key to return to the PAL program.

4.4.4 File Directory

Use this command to obtain a list of files on the disk. When you run the command, an input box is presented for keying in specific files to be listed in a directory. The standard DOS wildcards are also available. Press <Enter> without keying in anything and a complete listing of the directory is displayed. Press <Esc> to return to the main screen.

You may also load files using the File Directory command. After obtaining a listing of files, use the cursor keys to highlight any file and press <Enter>. The file is loaded into the memory buffer.

4.4.5 File OS Shell

The OS Shell command temporarily exits you to DOS. All standard DOS commands are then available. You should refrain from executing any terminate-and-stay-resident (TSR) programs when in the OS shell, since they may interfere with the PAL program's operation when you return. Be aware that the DOS PRINT command is a TSR.

To return to the PAL program from the OS shell, type:

```
exit  <Enter>
```

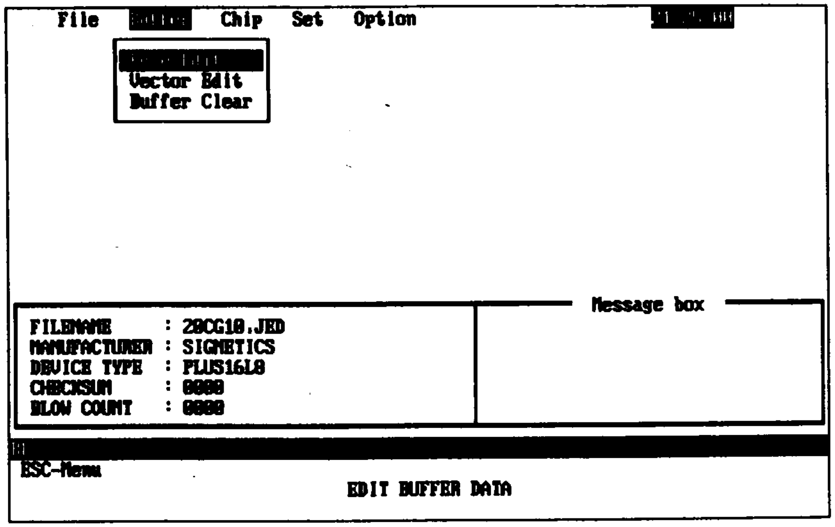
4.4.6 File Quit

This command quits the program and returns you to DOS. Each time you quit the PAL program, a configuration file is automatically saved with the filename PAL.DAT. This file contains the various settings you have selected, including the name of the JEDEC file currently opened in the memory buffer, the port address, the selected manufacturer and device, and so on. Thus when you start the program again, the options are restored to their previous settings.

> *WARNING*
>
> *Be sure to turn the power off on the programming module before quitting the program and returning to DOS. Leaving the power on and quitting to DOS could result in damage to the programming module.*

4.5 The Buffer Menu

The Buffer menu contains three commands for editing fuse map files, editing test vector files, and for clearing the memory buffer.



```
FILENAME     : 20CG10.JED
MANUFACTURER : SIGNETICS
DEVICE TYPE  : PLUS16L8
CHECKSUM     : 0000
BLOW COUNT   : 0000
```

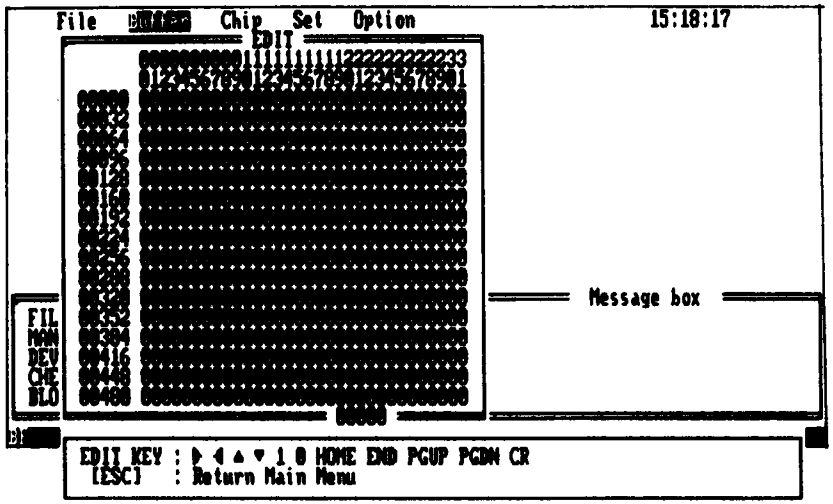
EDIT BUFFER DATA

Figure 4-3. The Buffer menu

4.5.1 Buffer Fuse Edit

Upon executing the Buffer Fuse Edit command, an editing window is displayed. Contained in the window are the contents of the fuse map file currently open in the memory buffer. You may edit a fuse map loaded with the File Load command or read in from a previously programmed device with the Chip Read command.

If the memory buffer is empty, the editing window is clear and a new fuse map may be created from scratch (provided you have an infinite amount of patience and concentration).



```
EDIT KEY : ▶ ◀ ▲ ▼ 1 0 HOME END PGUP PGDN CR
 [ESC]   : Return Main Menu
```

Figure 4-4. The Buffer Fuse Edit window

The editing window includes fuse numbers to the left, and column numbers along the top. Across the bottom is a display of the keys available for use in the editing window. They include: Cursor pad keys. Right, Left, Up, Down, Home, End, PgUp, PgDn, and CR (Carriage Return) for moving within the editing window.

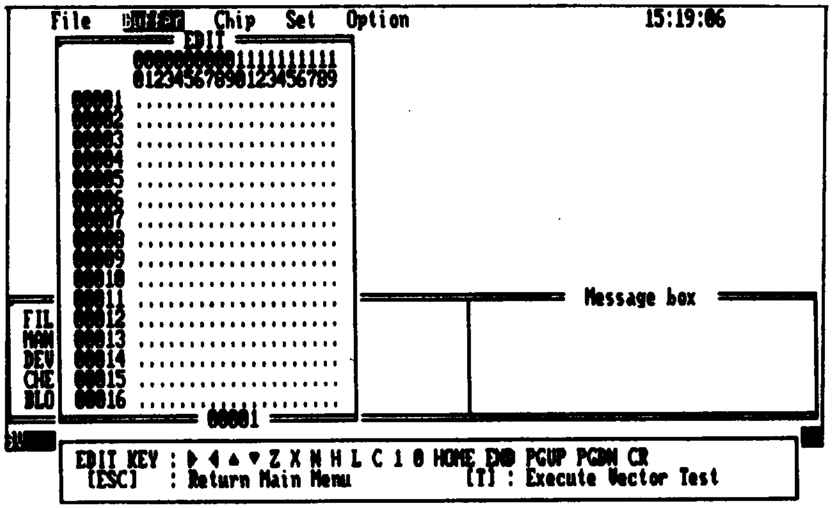
1	Program (blow) this fuse, set the value to 1.
0	Don't program (blow) this fuse, set the value to 0
<Esc>	Return to the Buffer menu.

When you exit the fuse edit window with <Esc>, any changes you made remain intact. However, they have not yet been saved to disk. It is a good idea to run the File Save command after exiting the fuse editor to save your changes.

You cannot undo changes made to the fuse map in the memory buffer, apart from re-keying in the correct information, or re-loading the original file from disk or from the chip into the memory buffer.

4.5.2 Buffer Vector Edit

When you run the Buffer Vector Edit command, an editing window similar to that described above is displayed. If the fuse map file in the current memory buffer contains test vector data, it will be displayed in the window.



```
       00000000001111111111
       01234567890123456789
00001  ....................
00002  ....................
00003  ....................
00004  ....................
00005  ....................
00006  ....................
00007  ....................
00008  ....................
00009  ....................
00010  ....................
00011  ....................
00012  ....................
00013  ....................
00014  ....................
00015  ....................
00016  ....................
              00001

EDIT KEY : ▶ ◀ ▲ ▼ Z X N H L C 1 0 HOME END PGUP PGDN CR
 [ESC]   : Return Main Menu            [T] : Execute Vector Test
```

Figure 4-5. The Buffer Vector Edit window

You may edit test vectors contained in a JEDEC file loaded with the File Load command. If the memory buffer is empty, the editing window is clear and new test vectors may be created.

The editing window includes test vector numbers to the left, and column numbers along the top. The columns correspond to the pins on the device with the column number equal to the pin number minus one. For example, column 0 corresponds to pin 1, column 1 with pin 2, column 2 with pin 3, and so on.

Across the bottom is a display of the keys available for use in the editing window. They include:

N	: Power pins and outputs not tested.
H	: Test the output for High.
Z	: High impedance (disable).
L	: Test the output for Low.
C	: Drive the input Low, High, Low (clocked).
0	: Drive the input Low, set the value to 0
1	: Drive the input High, set the value to 1
X	: Output not tested, input default value ("Don't care").
T	: Run the test vectors. This option runs the vectors as displayed and steps to the first test vector which fails.
<Esc>	: Return to the Buffer menu.

Cursor pad keys : Right, Left, Up, Down, Home, End, PgUp, PgDn, and CR (Carriage Return) for moving within the editing window.

When you exit the vector edit window with <Esc>, any changes you made remain intact. However, they have not yet been saved to disk. It is a good idea to run the File Save command after exiting the vector editor to save your changes.

You cannot undo changes made to the test vectors in the memory buffer, apart from re-keying in the correct information, or re-loading the original file from disk into the memory buffer.

4.5.3 Buffer Clear

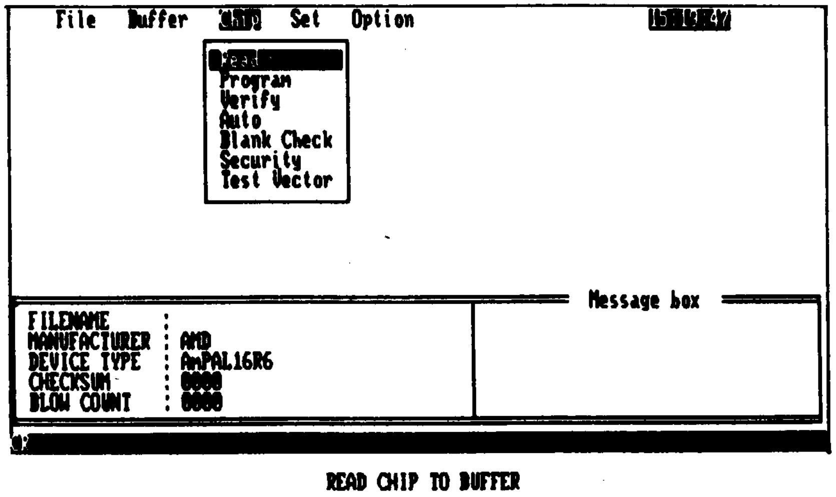
The Buffer Clear command erases all fuse map and test vector data from the current memory buffer.

WARNING

Be sure you save any changes you have made to the fuse map or test vectors using File Save prior to clearing the memory buffer.

4.6 The Chip Menu

The Chip menu contains commands which cause the software to interrelate with the programming module. Included are commands to read the contents of a previously programmed device into the memory buffer, to program a device with the fuse map contained in the memory buffer, to verify the programming of a device, to perform a blank check and to set the security fuse on a device. Each of these commands may be run independently of each other, or alternately, you may blank check, program, verify and set security all with a single command.



```
FILENAME     :
MANUFACTURER : AMD
DEVICE TYPE  : AmPAL16R6
CHECKSUM     : 0000
BLOW COUNT   : 0000
```

READ CHIP TO BUFFER

Figure 4-6. The Chip menu

4.6.1 Chip Read

The Chip Read command copies the contents of the device which is currently located on the programming module into the memory buffer. To correctly read the contents of the device, you must first select the appropriate manufacturer and device with the Set Device Type command. The manufacturer and the device type will then be displayed in the status box and must match that of the part on the programmer.
Once read from the chip, the copied fuse map may be edited, saved to disk or used to program other similar devices.

Fuse maps may only be read from devices which have not had the security fuse programmed. If the security fuse has been set in a device, the Chip Read command will read an all zero or all one fuse map from the device to the memory buffer.

4.6.2 Chip Program

This command downloads the fuse map data from the memory buffer to the programming module which in turn programs the device placed in its socket. After programming, the Chip Program command automatically performs a verification, comparing the contents of the memory buffer to that of the programmed device. Should an error occur, a message will be displayed with the first fuse number which was programmed incorrectly.

4.6.3 Chip Verify

The Chip Verify command allows you to verify that the contents of a device currently located on the programming module match the fuse map in the memory buffer. While the Chip Program command verifies at the time of programming, with the Chip Verify command you may perform verification at any time. Should an error occur, a message will be displayed with the first fuse number which was programmed incorrectly.

4.6.4 Chip Auto

The Chip Auto command is a time-saver. Itautomatically runs the following commands in sequence:

- Chip Blank Check
- Chip Program
- Chip Verify
- Chip Secure

4.6.5 Chip Blank Check

This command checks if the device currently located on the programming module is blank. Should any data be contained in the device, an error message is displayed with the first fuse number that is not blank (has been programmed).
Devices which have had their security fuse set will fail the blank check.

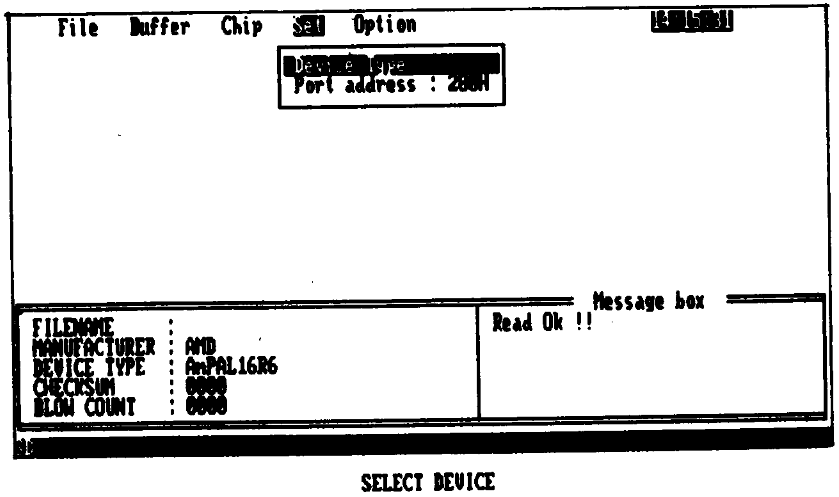
4.6.6 Chip Security

Nearly all programmable logic devices include a fuse known as the security bit which when set makes it impossible to read its contents. The Chip Security command sets the security fuse on the device currently located on the programming module.

Once the security fuse is set, running the Chip Read command will produce a blank fuse map. However, the secured device will fail when you run the Chip Blank Check command. By running these two commands, you should be able to determine if any device is either blank or secured.

4.7 The Set Menu

The Set menu includes two commands used for selecting a device to be programmed and for setting the port address for communications between your PC and the programming module.



```
FILENAME      :
MANUFACTURER  : AMD
DEVICE TYPE   : AmPAL16R6
CHECKSUM      : 0000
BLOW COUNT    : 0000
```

```
Read Ok !!
```

SELECT DEVICE

Figure 4-7. The Set menu

4.7.1 Set Device Type

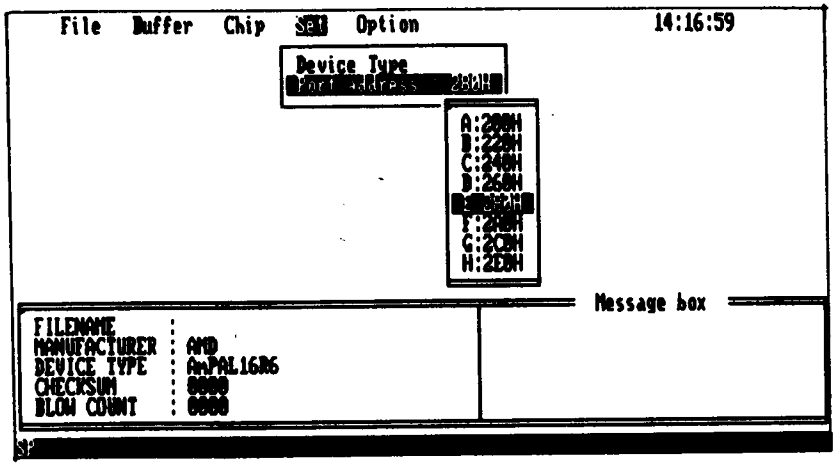
The Set Device Type command is used to select the manufacturer and type of device to be programmed. When you run the command, a list box appears containing the names of manufacturers currently supported by the software. Use the cursor keys to highlight the desired manufacturer and press <Enter>. A second list box appears containing a list of supported devices for that manufacturer by part name. Use <PgUp> or <PgDn> to scroll through long lists. Use the cursor keys to highlight the desired dèvice and press <Enter> to select that device.

You should run this command before the others because selecting the device "sets up" the software for correct operation with subsequent commands. For example, selecting a 24 pin device automatically sets up the test vector edit window with 24 columns. You must also select the appropriate device prior to running the Chip Read command.

XELTEK is constantly adding new devices to its list of supported parts, and distributes these new devices through periodic software updates. For additional information on updates, or on how to request support for new devices which are of interest to you, contact Xeltek or an authorized UNIPRO dealer or'distributor

4.7.2 Set Port Address

The Set Port Address command is used to change the port address for the programmer interface card which is installed in your computer. Normally, the port address need not be changed unless the jumper setting on the interface card is changed. This may become necessary if there is an address conflict caused by another I/O device with the same address. In this case, the port address should be set to match the address to the jumper position selected on the interface card. The default address is 280H. For more information on selecting the correct port address, consult the installation section of Chapter 2.



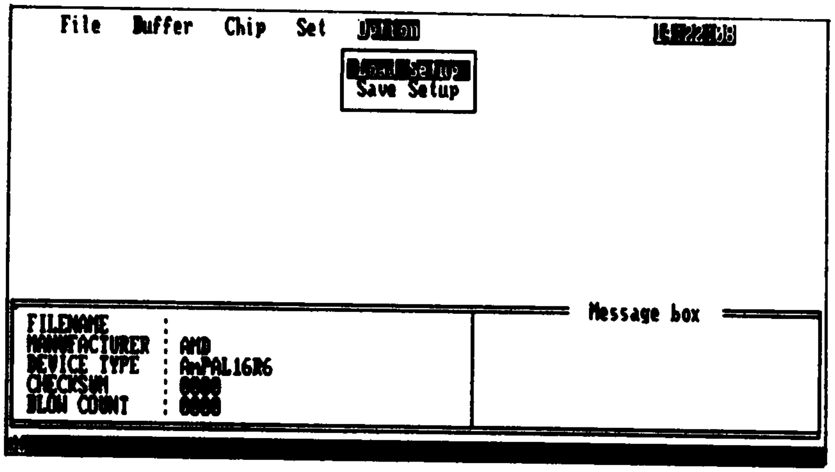
```
FILENAME      :
MANUFACTURER  : AMD
DEVICE TYPE   : AmPAL16R6
CHECKSUM      : 0000
BLOW COUNT    : 0000
```

SELECT PORT ADDRESS

Figure 4-8. Setting the port address

4.8 The Options Menu

The Options menu contains two commands which permit you to save and load setup files. The setup file includes a record of the various settings you have selected, including the name of the JEDEC file currently opened in the memory buffer, the port address, the selected manufacturer and device, and so on.



```
FILENAME      :
MANUFACTURER  : AMD
DEVICE TYPE   : AmPAL16R6
CHECKSUM      : 0000
BLOW COUNT    : 0000
```

LOAD CONFIGURATION FILE WITH SET OPTOIN

Figure 4-9. The Options menu

4.8.1 Options Load Setup

Use this command to load a setup file previously saved. To load a file, type the filename in the input box and press <Enter>.

4.8.2 Options Save Setup

The Options Save Setup command allows you to save your current setup under any standard DOS filename of your choice. You may later restore the current settings by loading your setup file with the Options Load Setup command.

4.9 Trouble Shooting Communication Errors

Communication errors may result from a number of factors, including:

1. Failure to have the programming module turned on when starting the PAL program.

2. Improper installation of the interface card.

3. Improper installation of the data cable.

4. Selection of a port address on the interface card which clashes with existing I/O devices in the computer.

5. Having a device inserted in the socket on the programming module when starting the PAL program.

6. Computers with clock speeds higher than 25 MHz could cause communication errors.

If an error exists when starting the PAL program, an intermediate menu appears with five options which may help alleviate the problem:

1. View Error Message. Select this option to get a full-screen error message with suggestions on how to resolve the communication problem.

2. Change Port Address. This option allows you to try a different port address. A list box appears with the addresses which are available. The address selected should match the address to the jumper position selected on the interface card. The default address is 280H.

3. Retry. Select this option after you have checked to see that the power to the programming module is on and that the cable connecting it to the interface card is properly installed.

4. Ignore. Select this option is you wish to ignore the communication problem. This will allow you to access the rest of the program so that you may load fuse map files from disk, edit them and save them without having the programming module or interface card attached or operative. Of course certain functions which require proper communications, such as Chip Read, Chip Program, and the balance of the Chip commands, will not operate under these conditions.

5. Quit. Return to DOS.

5 GAL Programming

5.1 Introduction

The UNIPRO may be used to program today's popular Generic Array Logic (GAL) devices. A menu-driven software program provides an easy interface for loading, editing, and downloading JEDEC files to the device programmer.
Files on the release disk used for GAL programming are:

GAL.EXE	The program.
GAL.DAT	A configuration file which is written to disk each time you quit the GAL program, and keeps a record of options selected. The file is read by the GAL program when it is re-started so that previously selected options are restored.

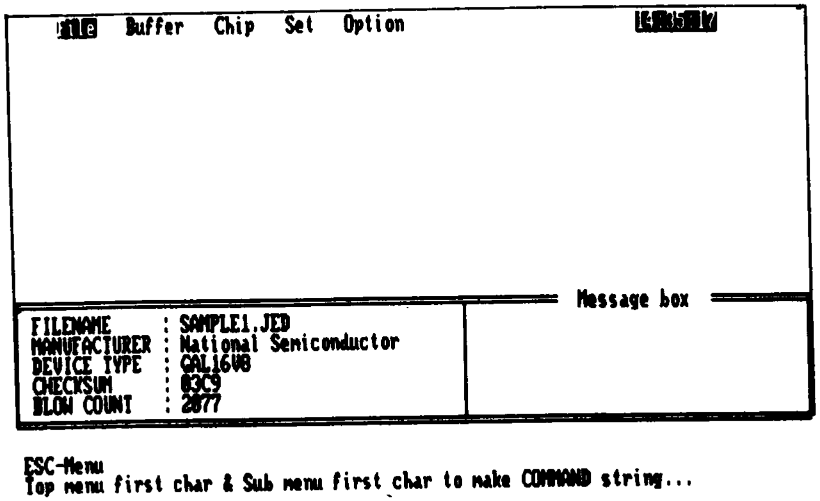
5.2 Programming GALs

To program GALs, first be certain that the UNIPRO hardware is properly installed, and that the programmer is turned on. To start the program, type:

```
gal <Enter>
```

5.3 The Main Screen

Upon executing the program, the main screen will appear, as shown in figure 5-1. (If an error occurs at this point, refer to the troubleshooting section below.) The user interface is the pull-down menu variety, with options either being selected from the menus or by typing in a command's key letter on the keyboard. We'll briefly describe the information provided on the main screen



```
FILENAME     : SAMPLE1.JED
MANUFACTURER : National Semiconductor
DEVICE TYPE  : GAL16V8
CHECKSUM     : 03C9
BLOW COUNT   : 2077

ESC-Menu
Top menu first char & Sub menu first char to make COMMAND string...
```

Figure 5-1. The GAL program's main screen

5.3.1 The Menu Line

Across the top of the screen is the menu line, which presents the main menu options, File, Buffer, Chip, Set and Option. Press <Esc> to access the menus. Then use the cursor (arrow) keys on your keyboard move the cursor from one option to another. Then press <Enter> to select that option. A pull-down menu will appear with sub-menu options. On the right side of the menu line is clock which displays the time of day based on your system's clock.

5.3.2 The Status Box

In the lower left corner of the screen is the status box. Displayed in the status box are:

FILENAME
Name of the file currently in the memory buffer being edited or downloaded.

MANUFACTURER
Manufacturer of the device currently selected for programming.

DEVICE TYPE
The part name of the device currently selected for programming.

CHECKSUM
The checksum on the fuse map portion of the file currently in the memory buffer.

BLOW COUNT
The number of fuses blown or to be blown on the device.

5.3.3 The Message Box

In the lower right hand corner of the screen is the message box. Messages are displayed here during various operations of the program to indicate their status and ultimate completion.

5.3.4 The Input Line

As indicated above, commands may be selected from the pull-down menus, or by typing their key letters on your keyboard. In either case, as the commands are selected, their key letters are displayed on the input line, which is located just below the status box.

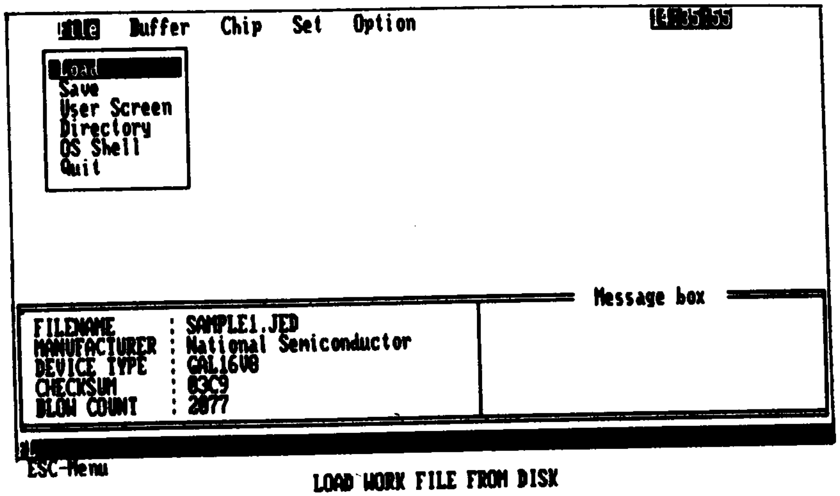
Once you have learned the various commands available in the GAL program, you may find it faster to type in a sequence of key letters, which will be displayed on the input line. Then simply press <Enter> and the sequence of commands will be selected and run without pulling down the menus.

5.3.5 The Prompt Line

The last line, at the bottom of the screen, is the prompt line. As each menu option is selected from the menus, the prompt line gives a brief explanation of its purpose and operation. Again, once you are familiar with the program, the prompt line may be all the help you'll need.

5.4 The File Menu

The File menu contains five commands for loading and saving JEDEC files, listing a directory of files on disk, temporarily exiting the program to DOS, and quitting the program.



```
FILENAME      : SAMPLE1.JED
MANUFACTURER  : National Semiconductor
DEVICE TYPE   : GAL16V8
CHECKSUM      : 03C9
BLOW COUNT    : 2077
```

LOAD WORK FILE FROM DISK

Figure 5-2. The File menu

5.4.1 File Load

This command is used to load a fuse map file from disk into the memory buffer. The program will prompt for the filename. The program accepts fuse maps in JEDEC format, which is the standard output of a wide variety of software available for designing PLDs. The UNIPRO supports JEDEC files created by all popular PLD design programs, including ABEL, CUPL, PALASM, Tango-PLD and OrCAD-PLD.
For further information on the JEDEC standard, refer to Appendix C.

5.4.2 File Save

The File Save command permits you to save the fuse map file which is currently in the memory buffer back to disk. This is important, since the GAL program includes editing functions which allow you to make

changes in a fuse map you have loaded from disk, read from a previously programmed component, or even created new from scratch. Files are always saved in JEDEC format and include both the fuse map plus any test vectors you may have created.

5.4.3 File User Screen

Use this command to view the screen, which can be saved just before the GAL software is invoked. Pres <enter> to view the precious screen, and hit any key to return to the GAL program.

5.4.4 File Directory

Use this command to obtain a list of files on the disk. When you run the command, an input box is presented for keying in specific files to be listed in a directory. The standard DOS wildcards are also available. Press <Enter> without keying in anything and a complete listing of the directory is displayed. Press <Esc> to return to the main screen.

You may also load files using the File Directory command. After obtaining a listing of files, use the cursor keys to highlight any file and press <Enter>. The file is loaded into the memory buffer.

5.4.5 File OS Shell

The OS Shell command temporarily exits you to DOS. All standard DOS commands are then available. You should refrain from executing any terminate-and-stay-resident (TSR) programs when in the OS shell, since they may interfere with the GAL program's operation when you return. Be aware that the DOS PRINT command is a TSR.
To return to the GAL program from the OS shell, type:

```
exit  <Enter>
```

5.4.6 File Quit

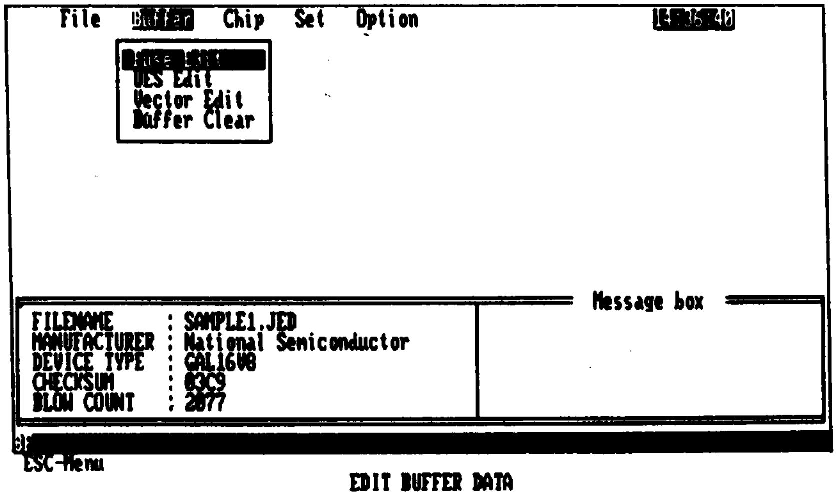
This command quits the program and returns you to DOS. Each time you quit the GAL program, a configuration file is automatically saved with the filename GAL.DAT. This file contains the various settings you have selected, including the name of the JEDEC file currently opened in the memory buffer, the port address, the selected manufacturer and device, and so on. Thus when you start the program again, the options are restored to their previous settings.

WARNING

Be sure to turn the power off on the programming module before quitting the program and returning to DOS. Leaving the power on and quitting to DOS could result in damage to the programming module.

5.5 The Buffer Menu

The Buffer menu contains three commands for editing fuse map files, editing test vector files, and for clearing the memory buffer.



```
FILENAME     : SAMPLE1.JED
MANUFACTURER : National Semiconductor
DEVICE TYPE  : GAL16V8
CHECKSUM     : 03C9
BLOW COUNT   : 2077
```

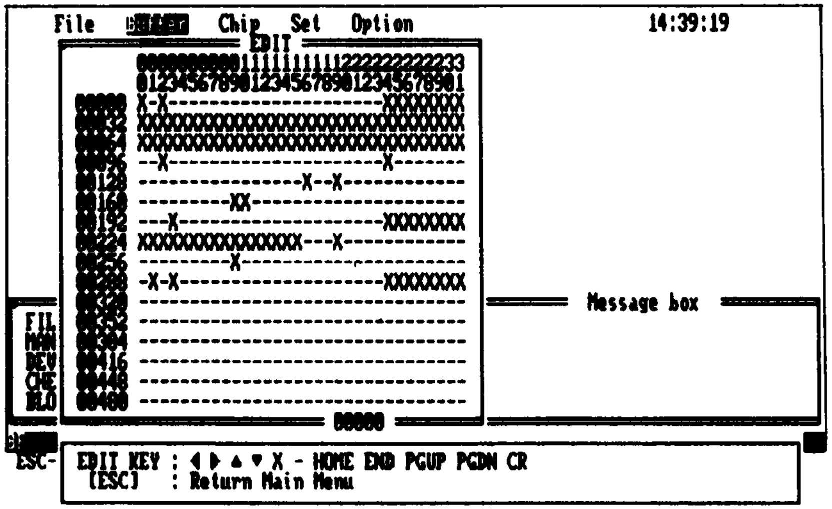
```
EDIT BUFFER DATA
```

Figure 5-3. The Buffer menu

5.5.1 Buffer Fuse Edit

Upon executing the Buffer Fuse Edit command, an editing window is displayed. Contained in the window are the contents of the fuse map file currently open in the memory buffer. You may edit a fuse map loaded with the File Load command or read in from a previously programmed device with the Chip Read command.

If the memory buffer is empty, the editing window is clear and a new fuse map may be created from scratch (provided you have an abundance of patience and concentration)



```
      00000000001111111111222222222233
      01234567890123456789012345678901
00000 X-X---------------------XXXXXXXX
00032 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
00064 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
00096 --X---------------------X-------
00128 ----------------X--X------------
00160 ---------XX---------------------
00192 ---X--------------------XXXXXXXX
00224 XXXXXXXXXXXXXXXX---X------------
00256 ---------X----------------------
00288 -X-X--------------------XXXXXXXX
00320 --------------------------------
00352 --------------------------------
00384 --------------------------------
00416 --------------------------------
00448 --------------------------------
00480 --------------------------------
                  00000
```

```
EDIT KEY : ◀ ▶ ▲ ▼ X - HOME END PGUP PGDN CR
 [ESC]   : Return Main Menu
```

Figure 5-4. Buffer Fuse Edit window

The editing window includes line numbers to the left, and column numbers along the top. Across the bottom is a display of the keys available for use in the editing window. They include:
Cursor pad keys. Right, Left, Up, Down, Home, End, PgUp, PgDn, and CR (Carriage Return) for moving within the editing window.

1 Program (blow) this fuse, set the value to 1 (one).

0 Do not program (blow) this fuse, set the value to 0 (zero).

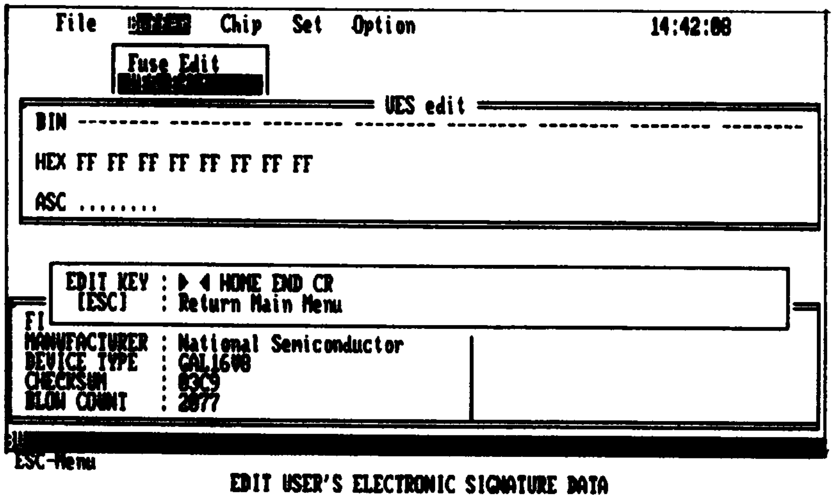
<Esc> Return to the Buffer menu.

When you exit the fuse edit window with <Esc>, any changes you made remain intact. However, they have not yet been saved to disk. It is a good idea to run the File Save command after exiting the fuse editor to save your changes.

You cannot undo changes made to the fuse map in the memory buffer, apart from re-loading the original file from disk or from the chip into the memory buffer.

5.5.2 Buffer UES Edit

GAL devices include a feature that permits the user to insert eight characters of information into the fuse map. This space is known as the User's Electronic Signature (UES) and may be utilized in any way. Typical uses include revision numbers, date, or other user-specific codes.he Buffer UES Edit command is used to create or make changes to the UES. When you run the command, an editing window pops up. The window will display whatever information is currently held in the UES portion of the file in the memory buffer. If the window is empty, a new UES may be created.You may key in up to eight characters. When finished, press <Esc> to return to the Buffer menu. When you exit the UES edit window with <Esc>, any changes you made remain intact. However, they have not yet been saved to disk. It is a good idea to run the File Save command after exiting the UES editor to save your changes.You cannot undo changes made to the UES in the memory buffer, apart from re-keying in the correct information, or re-loading the original file from disk into the memory buffer

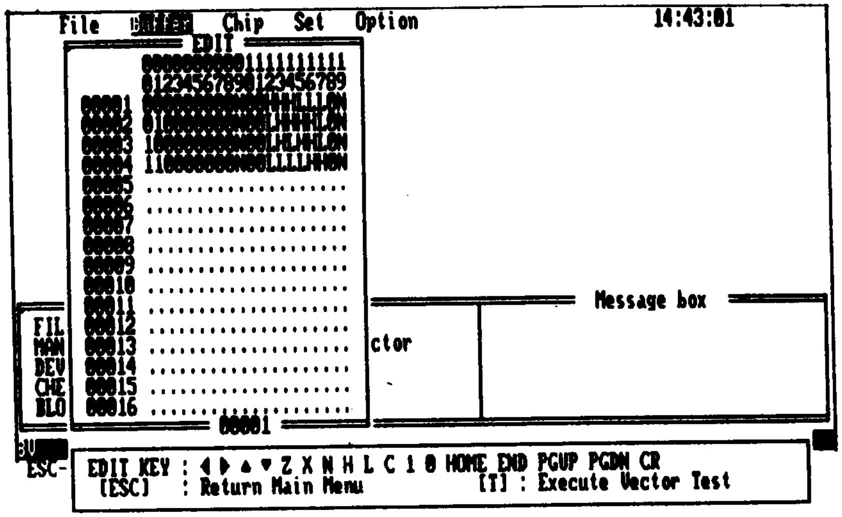


```
14:42:08
                      UES edit
BIN -------- -------- -------- -------- -------- -------- -------- --------

HEX FF FF FF FF FF FF FF FF

ASC ........

 EDIT KEY : ▶ ◀ HOME END CR
  [ESC]   : Return Main Menu
FI
MANUFACTURER : National Semiconductor
DEVICE TYPE  : GAL16V8
CHECKSUM     : 03C9
BLOW COUNT   : 2077

EDIT USER'S ELECTRONIC SIGNATURE DATA
```

Figure 5-5. The Buffer UES Edit window

5.5.3 Buffer Vector Edit

When you run the Buffer Vector Edit command, an editing window similar to that described above is displayed. If the fuse map file in the current memory buffer contains test vector data, it will be displayed in the window.



```
   File  Buffer  Chip  Set  Option                         14:43:01
       ═══════ EDIT ═══════
             00000000001111111111
             01234567890123456789
       00001 000000000000HHHLLLON
       00002 010000000000LHHHHLON
       00003 100000000000LHLHHLON
       00004 1100000000N0LLLLHHON
       00005 ....................
       00006 ....................
       00007 ....................
       00008 ....................
       00009 ....................
       00010 ....................                ════ Message box ════
 FIL   00011 ....................
 MAN   00012 ....................
 DEV   00013 .................... ctor
 CHE   00014 ....................
 BLO   00015 ....................
       00016 ....................
       ═══════ 00001 ═══════
 ESC- EDIT KEY : ◀ ▶ ▲ ▼ Z X N H L C 1 0 HOME END PGUP PGDN CR
       [ESC]   : Return Main Menu            [T] : Execute Vector Test
```

Figure 5-6. The Buffer Vector Edit window

You may edit test vectors contained in a JEDEC file loaded with the File Load command. If the memory buffer is empty, the editing window is clear and new test vectors may be created.

The editing window includes test vector numbers to the left, and column numbers along the top. The columns correspond to the pins on the device with the column number equal to the pin number minus one. For example, column 0 corresponds to pin 1, column 1 with pin 2, column 2 with pin 3, and so on.

Across the bottom is a display of the keys available for use in the editing window. They include:

N	: Power pins and outputs not tested.
H	: Test the output for High.
Z	: High impedance (disable).
L	: Test the output for Low.
C	: Drive the input Low, High, Low (clocked).
0	: Drive the input Low, set the value to 0
1	: Drive the input High, set the value to 1
X	: Output not tested, input default value ("Don't care").
T	: Run the test vectors. This option runs the vectors as displayed and steps to the first test vector which fails.
<Esc>	: Return to the Buffer menu.

Cursor pad keys : Right, Left, Up, Down, Home, End, PgUp, PgDn, and CR (Carriage Return) for moving within the editing window.

When you exit the vector edit window with <Esc>, any changes you made remain intact. However, they have not yet been saved to disk. It is a good idea to run the File Save command after exiting the vector editor to save your changes.

You cannot undo changes made to the test vectors in the memory buffer, apart from re-keying in the correct information, or re-loading the original file from disk into the memory buffer.

5.5.4 Buffer Clear

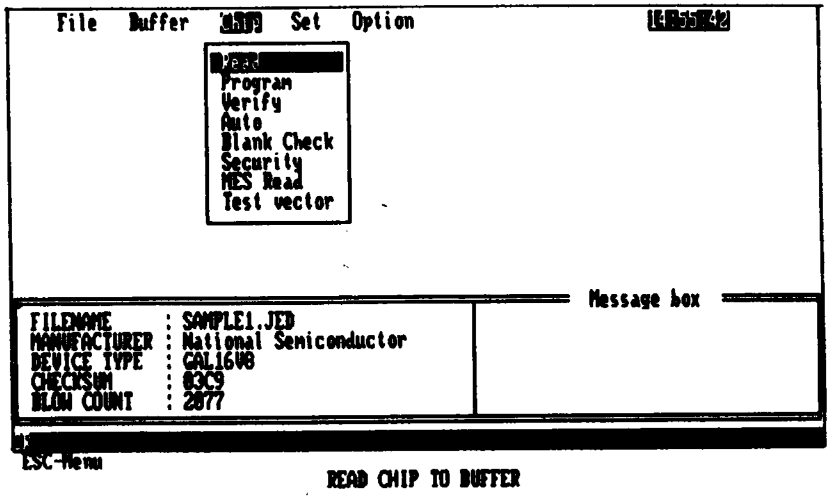
The Buffer Clear command erases all fuse map and test vector data from the current memory buffer.

WARNING

Be sure you save any changes you have made to the fuse map, UES or test vectors using File Save prior to clearing the memory buffer.

5.6 The Chip Menu

The Chip menu contains commands which cause the software to interrelate with the programming module. Included are commands to read the contents of a previously programmed device into the memory buffer, to program a device with the fuse map contained in the memory buffer, to verify the programming of a device, to perform a blank check and to set the security fuse on a device.
Each of these commands may be run independently of each other, or alternately, you may blank check, program, verify and set security all with a single command.



```
FILENAME     : SAMPLE1.JED
MANUFACTURER : National Semiconductor
DEVICE TYPE  : GAL16V8
CHECKSUM     : 83C9
BLOW COUNT   : 2877
ESC-Menu
                READ CHIP TO BUFFER
```

Figure 5-7. The Chip menu

5.6.1 Chip Read

The Chip Read command copies the contents of the device which is currently located on the programming module into the memory buffer.
To correctly read the contents of the device, you must first select the appropriate manufacturer and device with the Set Device Type command. The manufacturer and the device type will then be displayed in the status box and must match that of the part on the programmer.
Once read from the chip, the copied fuse map may be edited, saved to disk or used to program other similar devices.

Fuse maps may only be read from devices which have not had the security fuse programmed. If the security fuse has been set in a device, the Chip Read command will read an all zero or all one fuse map from the device to the memory buffer.

5.6.2 Chip Program

This command downloads the fuse map data from the memory buffer to the programming module which in turn programs the device placed in its socket. After programming, the Chip Program command automatically performs a verification, comparing the contents of the memory buffer to that of the programmed device. Should an error occur, a message will be displayed with the first fuse number which was programmed incorrectly.

5.6.3 Chip Verify

The Chip Verify command allows you to verify that the contents of a device currently located on the programming module match the fuse map in the memory buffer. While the Chip Program command verifies at the time of programming, with the Chip Verify command you may perform verification at any time. Should an error occur, a message will be displayed with the first fuse number which was programmed incorrectly.

5.6.4 Chip Auto

The Chip Auto command is a time-saver. It automatically runs the following commands in sequence:

- Chip Blank Check
- Chip Program
- Chip Verify
- Chip Secure

5.6.5 Chip Blank Check

This command checks if the device currently located on the programming module is blank. Should any data be contained in the

device, an error message is displayed with the first fuse number that is not blank (has been programmed).
Devices which have had their security fuse set will fail the blank check.

5.6.6 Chip Security

Nearly all programmable logic devices include a fuse known as the security bit which when set makes it impossible to read its contents. The Chip Security command sets the security fuse on the device currently located on the programming module.
Once the security fuse is set, running the Chip Read command will produce a blank fuse map. However, the secured device will fail when you run the Chip Blank Check command. By running these two commands, you should be able to determine if any device is either blank or secured.

5.6.7 Chip MES Read

GAL devices include a feature that provides for device-specific information to be inserted into "hidden fuses" within the device. The space is known as the Manufacturer's Electronic Signature (MES).
The Chip MES Read command is used to read the MES from the chip currently on the programming module. You should run the Set Device Type command prior to running the Chip MES Read command and be sure the status box displays the same manufacturer and device as the component on the programming module.

When run, the Chip MES Read command interrogates the device and displays the following information:

CYCLE COUNTER
: The number of times a device has been erased and programmed.

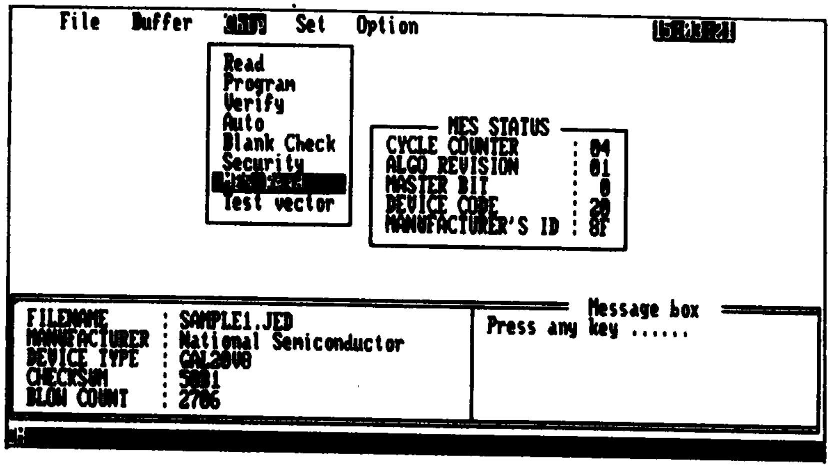
ALGORITHM REVISION
: Manufacturer-specific information.

MASTER BIT
: Manufacturer-specific information.

DEVICE CODE
: Manufacturer-specific information.

MANUFACTUER'S ID
: Manufacturer-specific information.

The cycle counter provides important information because GALs may be erased and re-programmed numerous times. However, the manufacturer guarantees only a finite number of erase/program cycles, so it is good to know if a particular device is within these bounds. Details on the other items may be obtained from the manufacturer.

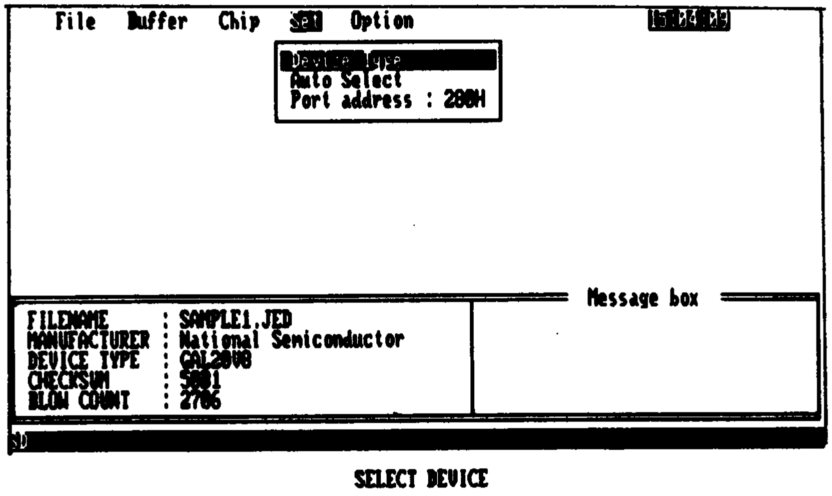


```
MES STATUS
CYCLE COUNTER       : 04
ALGO REVISION       : 01
MASTER BIT          : 0
DEVICE CODE         : 20
MANUFACTURER'S ID   : 8F

FILENAME      : SAMPLE1.JED
MANUFACTURER  : National Semiconductor
DEVICE TYPE   : GAL20V8
CHECKSUM      : 5001
BLOW COUNT    : 2706

Press any key ......

READ MANUFACTURER ELECTRONIC SIGNATURE DATA
```

Figure 5-8. The display of MES data

5.7 The Set Menu

The Set menu includes two commands used for selecting a device to be programmed and for setting the port address for communications between your PC and the programming module.



FILENAME : SAMPLE1.JED
MANUFACTURER : National Semiconductor
DEVICE TYPE : GAL20V8
CHECKSUM : 5001
BLOW COUNT : 2706

Figure 5-9. The Set menu

5.7.1 Set Device Type

The Set Device Type command is used to select the manufacturer and type of device to be programmed. When you run the command, a list box appears containing the names of manufacturers currently supported by the software. Use the cursor keys to highlight the desired manufacturer and press <Enter>. A second list box appears containing a list of supported devices for that manufacturer by part name. Use <PgUp> or <PgDn> to scroll through long lists. Use the cursor keys to highlight the desired GAL device and press <Enter>.

A third list box appears which includes the GAL device and the RAL devices which may optionally be used. Each RAL is a psuedo-device which configures a subset of the GAL's full capability. In this way, the GAL can be programmed in a wide variety of ways. For instance, the RAL10L8 makes a GAL16V8 behave like a 10 input, 8 output PAL with active low outputs.

If you have developed your JEDEC fuse map for a PAL10L8 and now want to use a GAL, you must select the RAL10L8 device to properly program the GAL.

In the third list box, use the cursor keys to highlight the desired GAL or RAL device and press <Enter>.

You should run this command before the others because selecting the device "sets up" the software for correct operation with subsequent commands. For example, selecting a 24 pin device automatically sets up the test vector edit window with 24 columns. You must also select the appropriate device prior to running the Chip Read command.

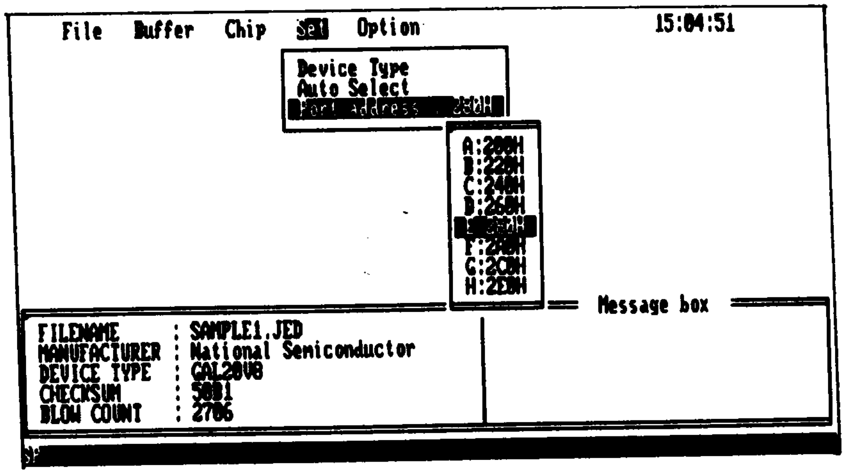
XELTEK is constantly adding new devices to its list of supported parts, and distributes these new devices through periodic software-only updates. For additional information on updates, or on how to request support for new devices which are of interest to you, contact Xeltek or an authorized UNIPRO dealer or distributor.

5.7.2 Set Auto Select

If you have a GAL device on the programming module, the Set Auto Select command will automatically select its correct manufacturer and device for you. When you run the command, a list box appears with two entries, 20 pin and 24 pin. Highlight the number of pins desired and press <Enter>. The program will read the MES data from the device on the programming module and will select the correct manufacturer and device, displaying them in the status box.

5.7.3 Set Port Address

The Set Port Address command is used to change the port address for the programmer interface card which is installed in your computer. Normally, the port address need not be changed unless the jumper setting on the interface card is changed. This may become necessary if there is an address conflict caused by another I/O device with the same address. In this case, the port address should be set to match the address to the jumper position selected on the interface card. The default address is 280H. For more information on selecting the correct port address, consult the installation section of Chapter 2.



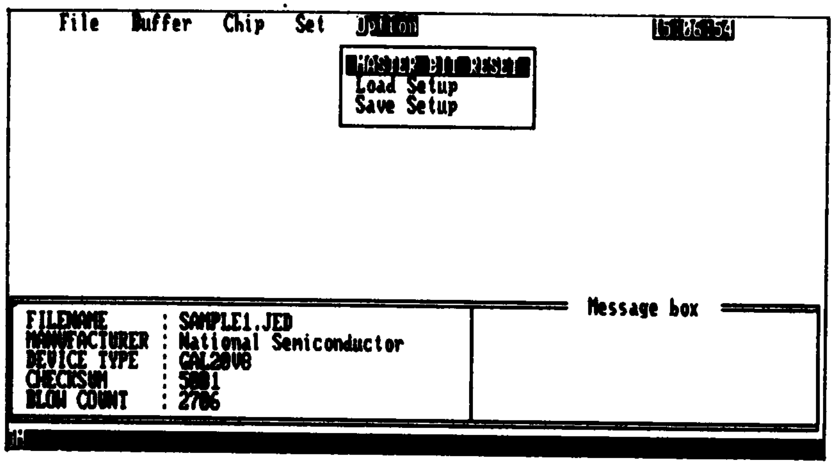
```
FILENAME     : SAMPLE1.JED
MANUFACTURER : National Semiconductor
DEVICE TYPE  : GAL20V8
CHECKSUM     : 5081
BLOW COUNT   : 2706
```

SELECT PORT ADDRESS

Figure 5-10. Setting the port address

5.8 The Options Menu

The Options menu contains a command to set and reset a device's master bit, and two commands which permit you to save and load setup files. The setup file includes a record of the various settings you have selected, including the name of the JEDEC file currently opened in the memory buffer, the port address, the selected manufacturer and device, and so on.



```
FILENAME     : SAMPLE1.JED
MANUFACTURER : National Semiconductor
DEVICE TYPE  : GAL20V8
CHECKSUM     : 5081
BLOW COUNT   : 2706
```

SET OR RESET MASTER DEVICE

Figure 5-11. The Options menu

5.8.1 Options Master Bit Set/Reset

Use this command to set or reset the master bit in a GAL device. To set or reset the master bit, highlight the Options Master Bit Set/Reset command on the menu and press <Enter>. The display of the command name on the menu toggles between Set and Reset. The status of a device's master bit may be viewed by running the Chip MES Read command.

5.8.2 Options Load Setup

Use this command to load a setup file previously saved. To load a file, type the filename in the input box and press <Enter>.

5.8.3 Options Save Setup

The Options Save Setup command allows you to save your current configuration under any standard DOS filename of your choice. You may later restore the current settings by loading your setup file with the Options Load Setup command.

5.9 Trouble Shooting Communication Errors

Communication errors may result from a number of factors, including:

1. Failure to have the programming module turned on when starting the GAL program.

2. Improper installation of the interface card.

3. Improper installation of the data cable.

4. Selection of a port address on the interface card which clashes with existing I/O devices in the computer.

5. Having a device inserted in the socket on the programming module when starting the GAL program.

6. Computers with clock speeds higher than 25 MHz sometimes causes communication errors.

If an error exists when starting the GAL program, an intermediate menu appears with five options which may help alleviate the problem:

1. View Error Message. Select this option to get a full-screen error message with suggestions on how to resolve the communication problem.

2. Change Port Address. This option allows you to try a different port address. A list box appears with the addresses which are available. The address selected should match the address to the jumper position selected on the interface card. The default address is 280H.

3. Retry. Select this option after you have checked to see that the power to the programming module is on and that the cable connecting it to the interface card is properly installed.

4. Ignore. Select this option is you wish to ignore the communication problem. This will allow you to access the rest of the program so that you may load fuse map files from disk, edit them and save them without having the programming module or interface card attached or operative. Of course certain functions which require proper communications, such as Chip Read, Chip Program, and the balance of the Chip commands, will not operate under these conditions.

5. Quit. Return to DOS.

6 Bipolar PROM Programming

6.1 Introduction

The UNIPRO may be used to program a wide range of popular bipolar PROM devices. A menu-driven software program provides an easy interface for loading, editing, and downloading files to the device programmer.

Files on the release disk used for bipolar PROM programming include the following:

BPR.EXE	The program.
BPR.DAT	A configuration file which is transferred to the disk each time you quit the BPR program, and keeps a record of options selected. The file is read by the BPR program when it is re-started, so that previously selected options are restored.

The following checklist shows the steps required to run the UNIPRO programmer. Each step is explained in detail in this chapter:

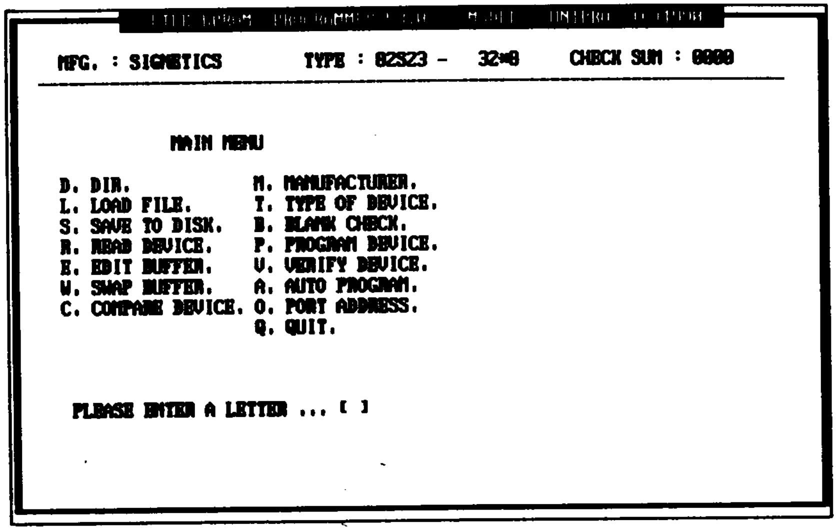
6.2 Programming Bipolar PROMs

To program bipolar PROMs, first be certain that the UNIPRO hardware is properly installed, and that the program is turned on. To start the program, type:

bpr <Enter>

6.3 The Main Screen

Upon executing the program, the main screen will appear, as shown in figure 6-1. (If an error occurs at this point, refer to the troubleshooting section below.)



```
[illegible]

MFG. : SIGNETICS        TYPE : 82S23 -   32*8     CHECK SUM : 0000


            MAIN MENU

D. DIR.               M. MANUFACTURER.
L. LOAD FILE.         T. TYPE OF DEVICE.
S. SAVE TO DISK.      B. BLANK CHECK.
R. READ DEVICE.       P. PROGRAM DEVICE.
E. EDIT BUFFER.       V. VERIFY DEVICE.
W. SWAP BUFFER.       A. AUTO PROGRAM.
C. COMPARE DEVICE.    O. PORT ADDRESS.
                      Q. QUIT.


 PLEASE ENTER A LETTER ... [ ]
```

Figure 6-1. The BPR program's main screen

6.4 Menu Options

Options are selected from the main menu by typing a key letter on the keyboard. Each option will be described below.

6.4.1 DIR

This option lists the files in the current subdirectory on disk. Wildcards like those used in DOS are available. Press <Esc> to return to the main menu.

6.4.2 LOAD FILE

Loads data from the disk into the memory buffer.

6.4.3 SAVE TO DISK

Saves data from the memory buffer onto the disk. To save the entire content of the memory buffer, please specify only the file name you want saved. Press <Esc> to return to the main menu.

6.4.4 READ DEVICE

Reads data from the device currently positioned on the UNIPRO device programming module. The read data is stored in the memory buffer. The chip's starting and ending address, and the buffer's starting address may be specified if needed. Press Y to choose the default mode, where the starting address is 0, or press C to change any parameters.

When there are two 4-bit devices to be read in, first read in the high four bit device, then run the SWAP BUFFER option. This causes the data to be located at the high four bit area in the memory buffer. Then read the low four bit device.

6.4.5 EDIT BUFFER

This option displays the contents of the memory buffer, and makes it available for editing. To edit, place the cursor in proper position using the cursor (arrow) keys. If the data is only four bits wide, then the high four bits will appear as 0. Press ESC to return to the main menu.

6.4.6 SWAP BUFFER

This option swaps the high four bits and low four bits of data in the memory buffer. The buffer's starting and ending addresses should be specified.

6.4.7 COMPARE DEVICE

Select this option to compare the device's contents in the socket to the memory buffer's contents. To compare less than the full range of the device, specify the address range by selecting the option C. Press Y to execute.

Discrepancies are displayed with the address and data information in the format:

Chip Address:Chip Data - Buffer Address:Buffer Data

While displaying any discrepancies, you may pause the display by typing CTRL-S, or terminate the display by typing ESC.

6.4.8 MANUFACTURER

In order to read or write a bipolar PROM, the manufacturer should be selected using this option.

6.4.9 TYPE OF DEVICE

The type of device is specified after the manufacturer has been selected. If there is not an exact device type to match yours, a similar device may be selected if the programming voltage and algorithm match with your device.

6.4.10 BLANK CHECK

This option checks to see if the contents of the bipolar PROM is blank or empty. An error message is displayed with address information of any data residing in the device.

6.4.11 PROGRAM DEVICE

This option downloads data from the memory buffer to the device on the programming module socket. Press Y to choose the default mode, where the contents of the buffer between address 0, and the last used address are written into the device.
Should the size of the buffer exceed the size of the device, only that portion of the data in the buffer that will fit into the device will be written there.
When programming eight bits of data into two pieces of four bit devices, the low four bits of data will be programmed first. Then perform the SWAP BUFFER option to relocate the high four bits of data into the low four bit area. Now the second device may be programmed.

6.4.12 VERIFY DEVICE

This option compares the contents of the bipolar PROM to that of the memory buffer. Verification is done automatically after programming. To verify less than the full range of the PROM, select the desired address with the C option.

6.4.13 AUTO PROGRAM

This option provides automatic execution in sequence of the Blank Check and the Program and Verify operations.

6.4.14 PORT ADDRESS

This option allows you to reset the port address for the UNIPRO programming module. The default-port address is 280. Consult the installation instructions in Chapter 3 regarding the port address and the jumper settings on the interface card.

6.4.15 QUIT

Exits the BPR program and returns you to DOS. Be sure to turn the power off (of the programming module) prior to exiting the BPR program and returning to DOS.

6.5 Trouble Shooting Communication Errors

Communication errors may result from a number of factors, including:

1. Failure to have the programming module turned on when starting the BPR program.

2. Improper installation of the interface card.

3. Improper installation of the data cable.

4. Selection of a port address on the interface card which clashes with existing I/O devices in the computer.

5. Insertion of a device in the socket of the programming module when starting the BPR program.

7 Logic IC and RAM Testing

7.1 Introduction

The UNIPRO may be used to test logic IC and RAM memory devices. A menu-driven software program provides an easy interface for loading files, editing test patterns, and downloading test files to the device programmer.

Files on the release disk used for logic IC and RAM memory testing include the following:

ICT.EXE	The program.
TTL.DAT	Contains the TTL and CMOS test patterns.

The following checklist shows the steps required to test the Logic IC and RAM's. Each step is explained in detail in this chapter.

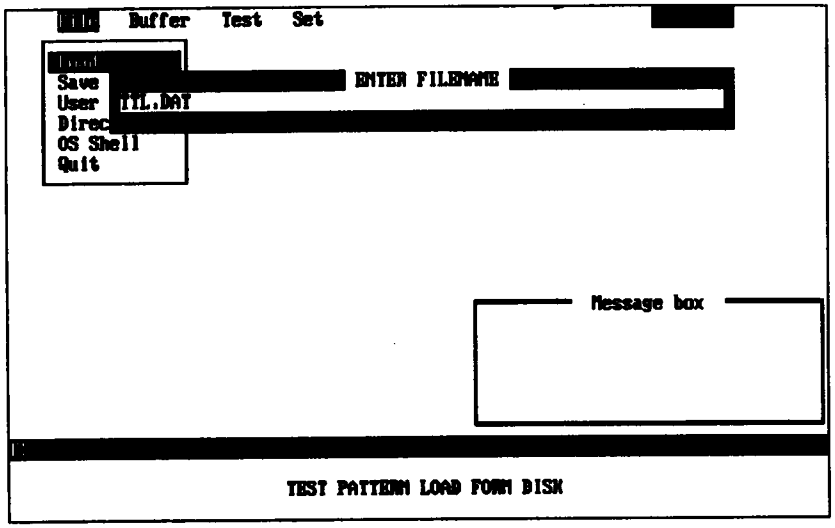
7.2 Testing Logic ICs and RAM Memories

To test logic ICs and RAM memories, be certain that the UNIPRO hardware is properly installed, and that the program is turned on. To start the program, type:

```
ict <Enter>
```

7.3 The Main Screen

Upon executing the program, the main screen will appear, as shown in figure 7-1. (If an error occurs at this point, refer to the troubleshooting section below.)



ENTER FILENAME

TTL.DAT

TEST PATTERN LOAD FORM DISK

Figure 7-1. The ICT program's main screen

7.4 Menu Options

Options are selected from the main menu by typing in a key letter on the keyboard. Each option will be described below.

7.4.1 DIR

This option lists the files in the current subdirectory on disk. Wildcards like those used in DOS are available. Press <Esc> to return to the main menu.

7.4.2 LOAD FILE

Loads test pattern data from the disk into the memory buffer.

> *To test logic ICs, be certain that a test pattern data file (TTL.DAT or a User Defined File) is loaded into be memory buffer first.*

7.4.3 EDIT PATTERN

This option is used to edit an existing test pattern in the user created test pattern file or TTL.DAT.

Following edit keys are used for editing existing test patterns for a selected device.
I : Specify Input
O : Specify Output
N : Power pins and outputs not tested
1 : Drive the input High, set the value to 1 (One)
0 : Drive the input Low, set the value to 0 (Zero)
C : Drive the input Low, High, Low (Clocked)
G : Specify Ground
V : Specify Vcc

7.4.4 Append Pattern

To test a new device that is not included in TTL.DAT, a new test pattern must be created. There are two ways to create new test patterns. The first is to use the Append Pattern option from the main menu. A new test file will be created in addition to the existing TTL.DAT file. All the user-created test patterns are stored in the new test file. The data format for the new test file is the same as the TTL.DAT file.

The alternate method is to create the test pattern file using a standard ASCII text editor or word processor. With this method, the new patterns may be added to TTL.DAT, or stored in a new test pattern file.

The data format is indicated by the example below:

```
"7400" 0
iioiiogoiioiiv
001001n100100n
101101n110110n
011011n101101n
110110n011011n
```

In the first line, the "7400" refers to a generic device type, and the 0 indicates it is not an open collector type. In the second line, the pin functions are described with the letter "i" for input, "o" for output, "g" for ground, "v" for VCC, and "c" for clock. Either upper or lower case text may be used. The maximum number of pins supported is 24. The subsequent lines are for test data, with 1, 0, N for no test, or C for clock being entered. There is no limit to the number of test lines.

Please refer to section 7.4.3 for detail explanations on editing keys.

7.4.5 SAVE TO DISK

Saves data from the memory buffer onto the disk. Specify the file name you want saved; the file name TTL.DAT may not be used. Press <Esc> to return to the main menu.

7.4.6 TTL & CMOS TEST

This option test TTL or CMOS logic devices according to the test pattern stored in TTL.DAT or in user-created files. When testing, enter the device type, such as 7400. Press **Enter** to begin the test. A message, indicating pass or fail, will appear.

7.4.7 Auto Find Device

This command finds out the device type of an unknown device. The program will search through either TTL.DAT or the user-created data file to find all matching patterns. The device type is displayed for all matching patterns ; a message "Device Not Found" is presented if no match is made.

7.4.8 MEMORY TEST

This option allows you to test static and dynamic RAM memory devices. When invoked, a list of memory devices is presented. Select a type and press **Enter** to perform the test.
For a static device, 55 and AA patterns are exercised. For a dynamic device, 1010----string and 0101----string are exercised.

7.4.9 PORT ADDRESS

This option allows you to reset the port address for the UNIPRO programming module. The port-default address is 280. Consult the installation instructions in Chapter 3 regarding the port address and the jumper settings on the interface card.

7.4.10 QUIT

Exits the ICT program and returns you to DOS. Be sure to turn the power off (on the programming module) prior to exiting the ICT program and returning to DOS.

7.5 Trouble Shooting Communication Errors

Communication errors may result from a number of factors, including:

1. Failure to have the programming module turned on when starting the ICT program.
2. Improper installation of the interface card.
3. Improper installation of the data cable.
4. Selection of a port address on the interface card which clashes with existing I/O devices in the computer.
5. Insertion of a device into the socket on the programming module when starting the ICT program.

8 LOPS Program

8.1 Introduction to LOPS

The Library-Operated Programming System (LOPS) may be used to program a wide range of Erasable Programmable Read Only Memory E(E)PROMs and single microcontrollers.

A menu-driven software program provides an easy interface for loading, editing, and downloading HEX files to the device programmer. Files on the release disk used for E(E)PROM and single microcontroller programming are:

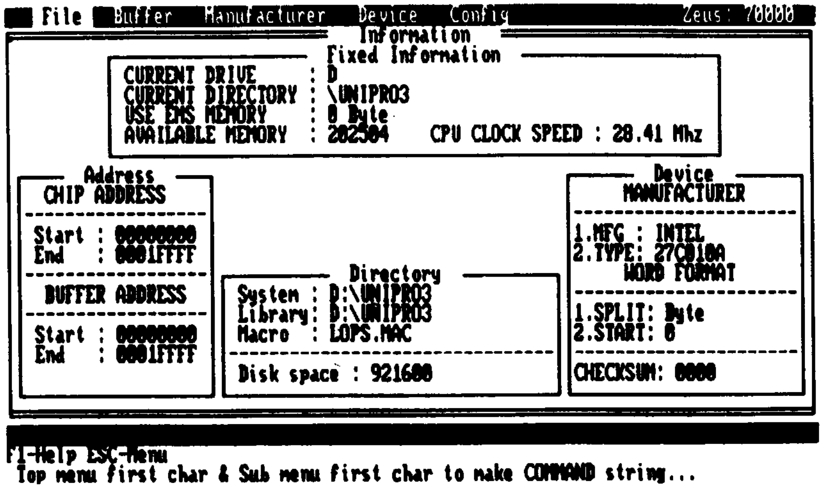
LOPS.EXE	Program
LOPS.HLP	Help file
CORP.INX	Manufacturer index file
CORP.DTA	Manufacturer data file
EPRS.INX	Device indexfile
EPRS.DTA	Device data file

8.2 The Main Screen

Upon executing the program, the main screen will appear, as shown in figure 8-1. (If an error occurs at this point, refer to the troubleshooting section below.) The user interface is the pull-down menu variety, with options either being selected from the menus or by typing in a command's key letter on the keyboard. We'll briefly describe the information provided on the main screen.

8.2.1 The Main Menu

Across the top of the screen is the menu line, which presents the main menu options, File, Buffer, Manufacturer, Device, Library, and Config. Use the cursor (arrow) keys on your keyboard to move the cursor from one menu to another. Then press <Enter> to select that menu. A pull-down menu will appear with sub-menu options. The main menu line situated at the top of screen contains selctions which lead to the submenus.



```
                    Information
                 Fixed Information
CURRENT DRIVE     : D
CURRENT DIRECTORY : \UNIPRO3
USE EMS MEMORY    : 0 Byte
AVAILABLE MEMORY  : 202504    CPU CLOCK SPEED : 28.41 Mhz

    Address                                        Device
  CHIP ADDRESS                                  MANUFACTURER
-----------------                        ----------------------
Start : 00000000                         1.MFG : INTEL
End   : 0001FFFF                         2.TYPE: 27C010A
-----------------          Directory          WORD FORMAT
 BUFFER ADDRESS     System : D:\UNIPRO3  ----------------------
-----------------   Library: D:\UNIPRO3  1.SPLIT: Byte
Start : 00000000    Macro  : LOPS.MAC    2.START: 0
End   : 0001FFFF    ---------------------  ----------------------
                    Disk space : 921600   CHECKSUM: 0000

F1-Help ESC-Menu
 Top menu first char & Sub menu first char to make COMMAND string...
```

Figure 8-1. The LOPS program's main screen.

8.2.2 The Fixed Information Box

In the top center of the screen is the Fixed Information Box. It displays the current drive, directory, EMS memory used and availability, and the CPU clock speed.

8.2.3 The Directory Box

In the lower center of the screen is the Directory box. It displays the directory location of the system, the library, and the name of the macro file currently being used. It also displays the total disk space available.

8.2.4 The Address Box

On the left side of the screen is the Address Box. It displays the address locations of both the chip and buffer.

8.2.5 The Device Box

On the right side of the screen is the Device box. Displayed in the Device box includes:

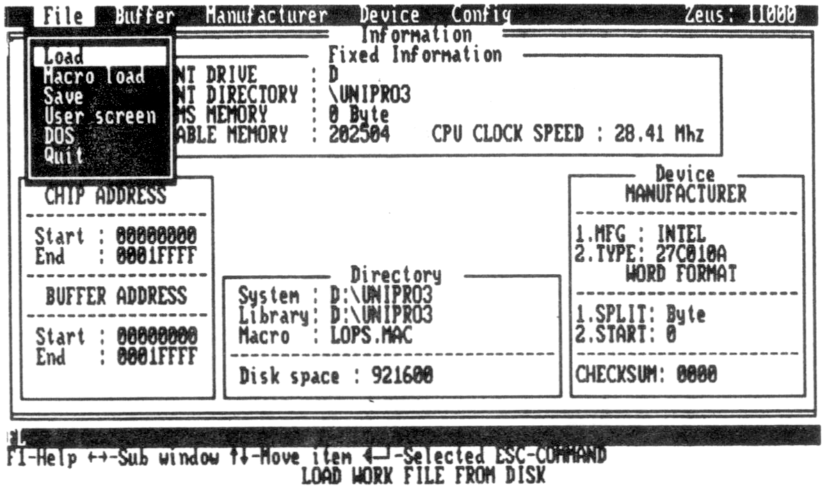
MANUFACTURER	Manufacturer of the device currently selected for programming.
DEVICE TYPE	The part name of the device currently selected for programming.
WORD FORMAT	The split and start location are both displayed.
CHECKSUM	The sum of the data in the memory buffer or chip.

8.2.6 The Prompt Line

Across the bottom of the screen is the prompt line; it provides a brief explanation of how to operate the program.

8.3 The File Menu

The File Menu contains six commands for loading and saving files, loading macro files, temporarily exiting the program to DOS, and quitting the program.

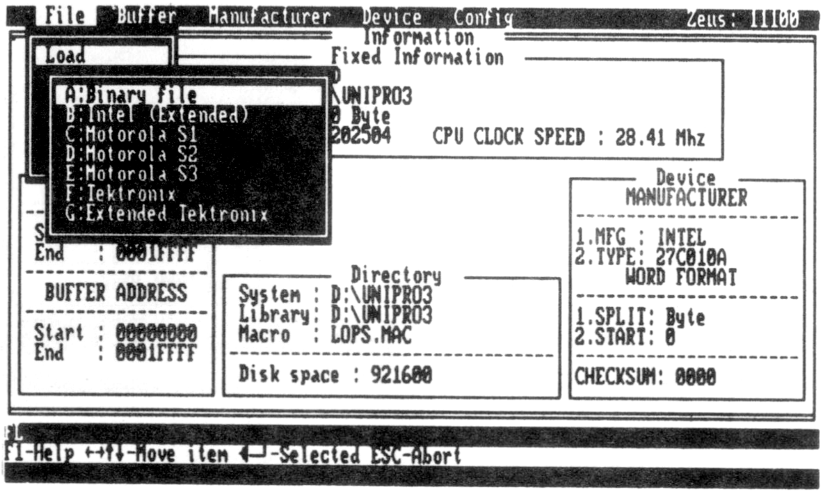


```
Information
Fixed Information
NT DRIVE      : D
NT DIRECTORY  : \UNIPRO3
MS MEMORY     : 0 Byte
ABLE MEMORY   : 202504     CPU CLOCK SPEED : 28.41 Mhz

CHIP ADDRESS
Start  : 00000000
End    : 0001FFFF
BUFFER ADDRESS
Start  : 00000000
End    : 0001FFFF

Directory
System  : D:\UNIPRO3
Library : D:\UNIPRO3
Macro   : LOPS.MAC
Disk space : 921600

Device
MANUFACTURER
1.MFG : INTEL
2.TYPE: 27C010A
    WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000

LOAD WORK FILE FROM DISK
```

Figure 8-2. The File menu.

8.3.1 File Load

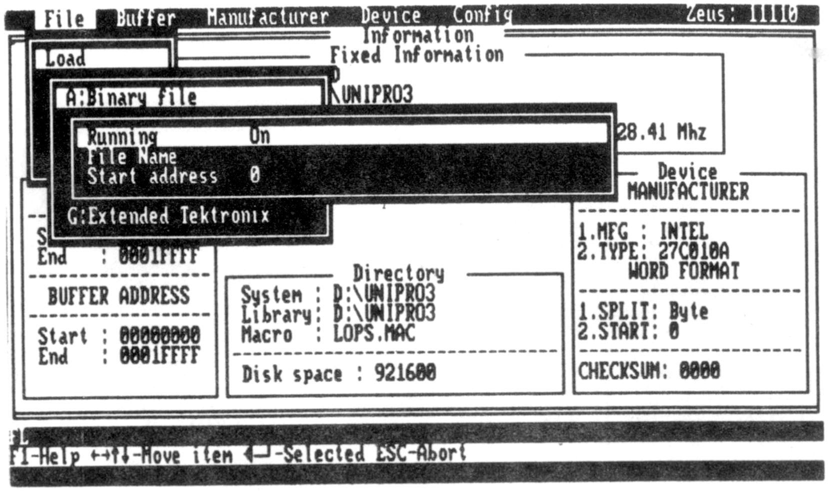
This command loads the data file from the disk or diskette into the buffer. Submenus are displayed in the Figure 8-3 below.



```
Information
Fixed Information
D
\UNIPRO3
0 Byte
202504     CPU CLOCK SPEED : 28.41 Mhz

S
End    : 0001FFFF
BUFFER ADDRESS
Start  : 00000000
End    : 0001FFFF

Directory
System  : D:\UNIPRO3
Library : D:\UNIPRO3
Macro   : LOPS.MAC
Disk space : 921600

Device
MANUFACTURER
1.MFG : INTEL
2.TYPE: 27C010A
    WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000
```

Figure 8-3. The File Load submenu.

Binary file: Reads the binary file into the memory. Multiple loading is possible when the menu CONFIG ENVIRONMENT MULTIPLE LOAD is set on. Submenu of the binary file load is divided by 3 menu as shown in figure 8-4.



```
 File  Buffer  Manufacturer  Device  Config                  Zeus: 11110
                          Information
 Load                  Fixed Information
   A:Binary file          D
                          \UNIPRO3
     Running           On                                28.41 Mhz
     File Name
     Start address     0                              Device
                                                   MANUFACTURER
   G:Extended Tektronix                          ---------------------
 --                                              1.MFG : INTEL
 S                                               2.TYPE: 27C010A
 End     : 0001FFFF                                  WORD FORMAT
 -----------------         Directory             ---------------------
  BUFFER ADDRESS     System : D:\UNIPRO3         1.SPLIT: Byte
 -----------------   Library: D:\UNIPRO3         2.START: 0
 Start   : 00000000  Macro  : LOPS.MAC           ---------------------
 End     : 0001FFFF  -------------------------   CHECKSUM: 0000
                     Disk space : 921600

F1-Help ←→↑↓-Move item ↵-Selected ESC-Abort
```

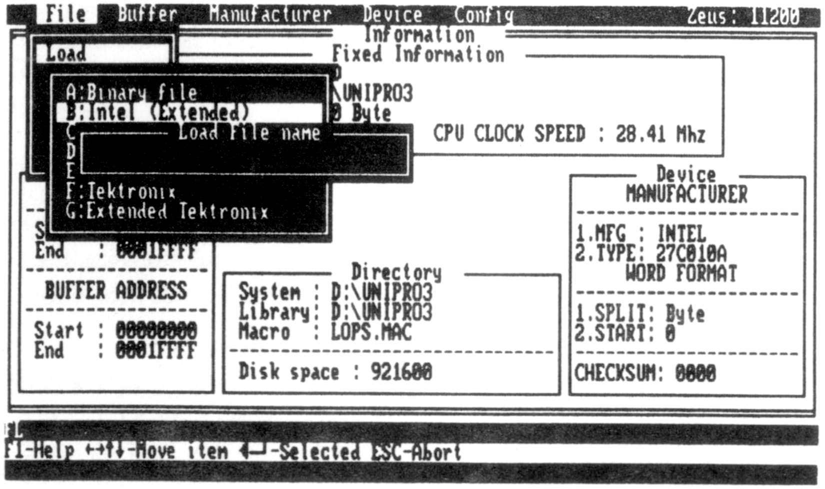
Figure 8-4. The Bînary File Load submenu.

Running : Actually excutes the command of loading files into the buffer. The other menus are set first, and this menu is slected for actual execution.

File name : Indicates file name which will be loaded into memory on the disk. If you leave the box blank or type * and <enter> on **running** then it will display all the files in the directory specied.

Start address : This menu is used to indicate the beginning of the address to be stored. The address doesn't mean the movement of the memory buffer; it only indicates the beginning address of a retrieved file to be stored in.

The other six File Load menu options perform to load hex-formatted file. The hex file is differentiated by its formats. It should be loaded using the right menu according to the file format.



```
 File  Buffer  Manufacturer  Device  Config                  Zeus: 11200
                          Information
 Load                  Fixed Information
   A:Binary file          D
   B:Intel (Extended)     \UNIPRO3
   C                      0 Byte
   D  ---- Load File name ----     CPU CLOCK SPEED : 28.41 Mhz
   E
   F:Tektronix                                      Device
   G:Extended Tektronix                          MANUFACTURER
 --                                              ---------------------
 S                                               1.MFG : INTEL
 End     : 0001FFFF                              2.TYPE: 27C010A
 -----------------         Directory                 WORD FORMAT
  BUFFER ADDRESS     System : D:\UNIPRO3         ---------------------
 -----------------   Library: D:\UNIPRO3         1.SPLIT: Byte
 Start   : 00000000  Macro  : LOPS.MAC           2.START: 0
 End     : 0001FFFF  -------------------------   ---------------------
                     Disk space : 921600         CHECKSUM: 0000

F1-Help ←→↑↓-Move item ↵-Selected ESC-Abort
```

Figure 8-5. The Hex File Load submenu

- Intel (Extended)
- Motorola S1
- Motorola S2
- Motorla S3
- Tektronix
- Extended Tektronix

8.3.2 File Macro Load

This menu loads macro file. If the system finds macro file, it loads into the buffer and reset macro table to notify standard macro. Details of the macro will be discussed in Chapter 9.4.2.

8.3.3 File Save

The meaning of save is to record the content of the buffer into the disk or diskette under the name specified by the user. **save** contains six submenus as below.

Binary :
Records the original content of the current buffer into the disk or diskette.

Intel (extended) :
Records the contents of a current memory buffer with intel extended hex file format.

Motorola S1 :
Records the contents of a current memory buffer with Motorola S1 formats.

Motorola S2 :
Records the contents of a current memory buffer with Motorola S2 hex file format.

Motorola S3 :
Records the contents of a current memory buffer with Motorola S3 hex file format.

Tektronix (extended) :
Records the contents of a current memory buffer with Tektronix (extended) hex file format.

Running
Actually excutes the command of saving files into the buffer. The other menus are set first, and this menu is slected for actual execution.

File name
Indicates file name which will be saved into the disk or diskette. If you leave the box blank or type * and <enter> on **running** then it will display all the files in the directory specied.

Start address
This menu is used to indicate the beginning of the address to be stored. It only indicates the beginning address of a file to be stored in.

Buffer size
Indicates the size of the file to be saved. For example, if start address is 0, and save size is FFF, then the contents (0 --FFF) is stored into the disk by the file name specified by an user.

8.3.4 File User Screen (Directory)

Use this command to view the screen, which is saved just before the software is invoked. Press <Enter> to view the previous screen, and hit any key to return to the LOPS program.

8.3.5 File DOS

The File DOS command temporarily exits you to DOS. All standard DOS commands are then available. You should refrain from executing any terminate-and-stay-resident (TSR) programs when in the OS shell, since they may interfere with the GAL program's operation when you return. Be aware that the DOS PRINT command is a TSR.

To return to the GAL program from the OS shell, type:

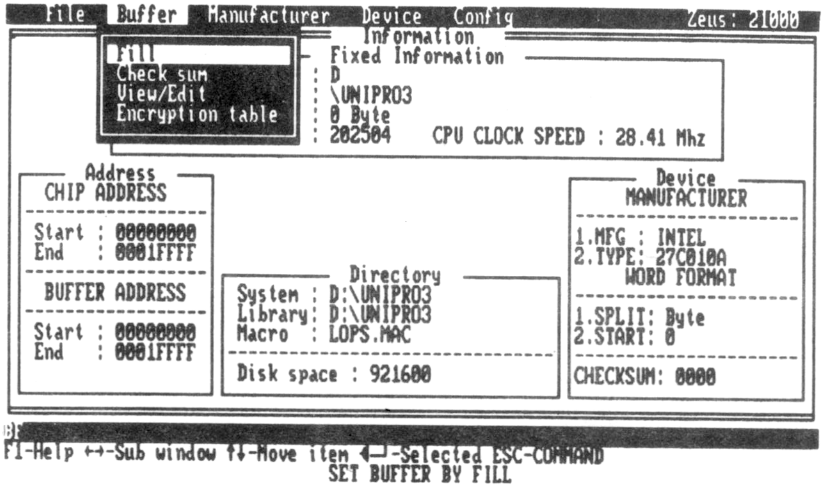
```
exit  <Enter>
```

8.3.6 File Quit

This command quits the program and returns you to DOS.

8.4 The Buffer Menu

The Buffer performs the functions that are related to data area and encryption.

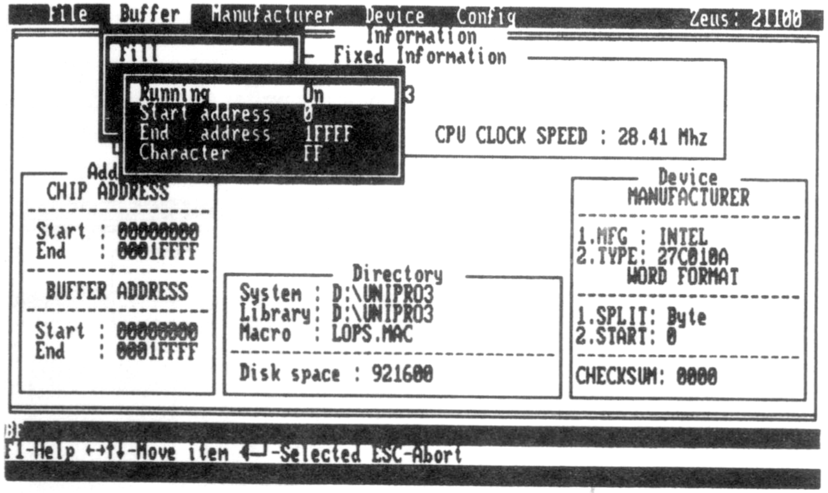


```
Information
Fixed Information
: D
: \UNIPRO3
: 0 Byte
: 202504     CPU CLOCK SPEED : 28.41 Mhz

Address
CHIP ADDRESS
Start  : 00000000
End    : 0001FFFF
BUFFER ADDRESS
Start  : 00000000
End    : 0001FFFF

Directory
System : D:\UNIPRO3
Library: D:\UNIPRO3
Macro  : LOPS.MAC
Disk space : 921600

Device
MANUFACTURER
1.MFG : INTEL
2.TYPE: 27C010A
   WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000

F1-Help ←→-Sub window ↑↓-Move item ←┘-Selected ESC-COMMAND
                    SET BUFFER BY FILL
```

Figure 8-6. The Buffer Menu

8.4.1 Buffer Fill

This command fills the specified area of the buffer with the hex value which is given by the user.



```
Running          On
Start address    0
End   address    1FFFF
Character        FF

Information
Fixed Information
             CPU CLOCK SPEED : 28.41 Mhz

CHIP ADDRESS
Start  : 00000000
End    : 0001FFFF
BUFFER ADDRESS
Start  : 00000000
End    : 0001FFFF

Directory
System : D:\UNIPRO3
Library: D:\UNIPRO3
Macro  : LOPS.MAC
Disk space : 921600

Device
MANUFACTURER
1.MFG : INTEL
2.TYPE: 27C010A
   WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000

F1-Help ←→↑↓-Move item ←┘-Selected ESC-Abort
```

Figure 8-7. The Buffer Fill menu

Running
Actually excutes the command of filling data into the buffer. The other menus are set first, and this menu is slected for actual execution.

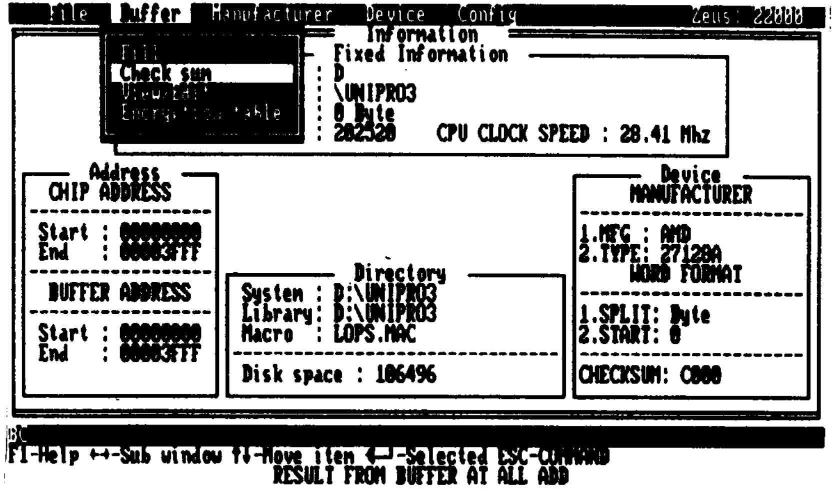
Start Address
Indicates the starting address to fill the memory buffer. If it is bigger than the size of memory buffer, the max size of buffer is set automatically.

End address
Indicates the ending address. If the starting address is bigger than ending address, then starting address is settled by one less than ending address.

Character
Receives the data value by hex for filling the buffer. The possible value is one byte (the byte for hex), and if there is more than one byte, only the ending byte will be effective.

8.4.2 Buffer Check Sum

Check sum menu adds the contents of memory buffer (starting address -- ending address), and shows the value of the least two significant bytes.



```
                    Information
                    Fixed Information
                    : D
                    : \UNIPRO3
                    : 0 Byte
                    : 282520    CPU CLOCK SPEED : 28.41 Mhz

     Address                                           Device
  CHIP ADDRESS                                      MANUFACTURER
 ----------------                               --------------------
 Start : 00000000                               1.MFG : AMD
 End   : 00003FFF                               2.TYPE: 27128A
 ----------------          Directory                WORD FORMAT
  BUFFER ADDRESS     System : D:\UNIPRO3        --------------------
 ----------------    Library: D:\UNIPRO3        1.SPLIT: Byte
 Start : 00000000    Macro  : LOPS.MAC          2.START: 0
 End   : 00003FFF    ----------------------     --------------------
                     Disk space : 106496        CHECKSUM: C000

F1-Help ←→-Sub window ↑↓-Move item ↵-Selected ESC-COMMAND
                    RESULT FROM BUFFER AT ALL ADD
```

Figure 8-8. The Buffer Checksum menu

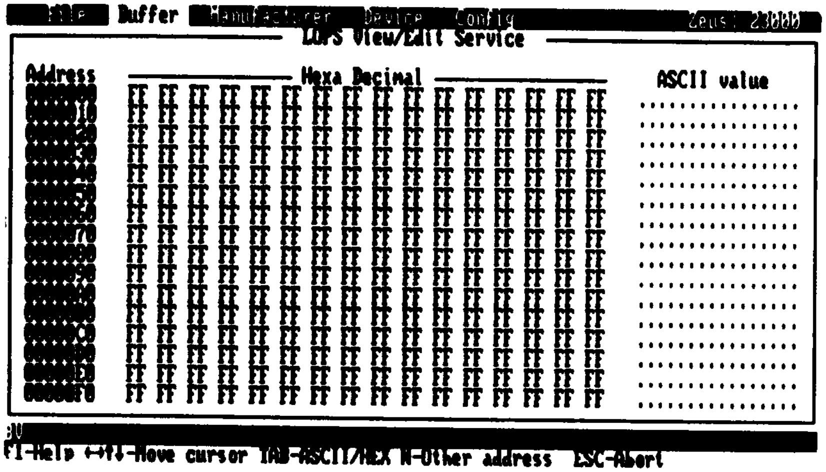
Running :
Actually excutes the command of **checksum**. The other menus are set first, and this menu is slected for actual execution.

Start Address :
Indicates the starting address to fill the memory buffer. If it is bigger than the size of memory buffer, the max size of buffer is set automatically.

End address :
Indicates the ending address. If the starting address is bigger than ending address, then starting address is settled by one less than ending address

8.4.3 Buffer View/Edit

After reading the data of the chip or loading file from a disk into buffer, this command views or edits the data in the memory buffer. The content of the memory buffer is displayed in hex and ASCII value formats.



```
                         LOPS View/Edit Service
Address   -------------- Hexa Decimal --------------         ASCII value
00000000  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
00000010  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
00000020  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
00000030  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
00000040  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
00000050  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
00000060  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
00000070  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
00000080  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
00000090  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
000000A0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
000000B0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
000000C0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
000000D0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
000000E0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................
000000F0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF    ................

F1-Help ←→↑↓-Move cursor TAB-ASCII/HEX N-Other address  ESC-Abort
```

Figure 8-9. The Buffer View/Edit screen.

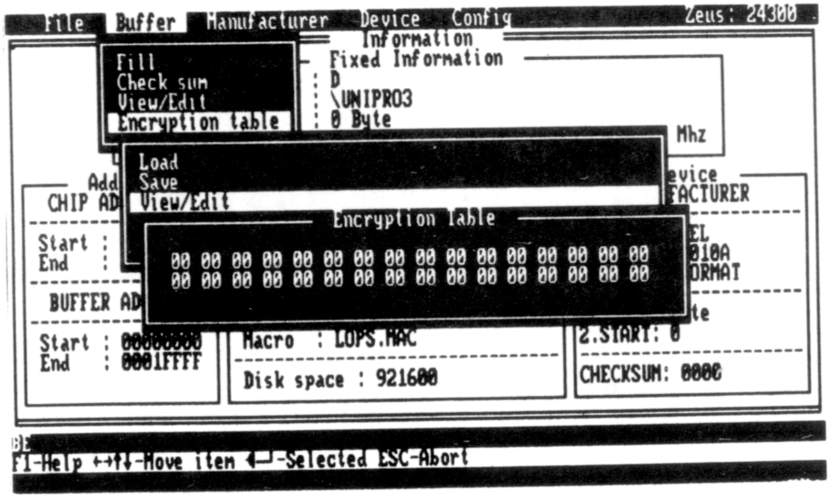
Keys for retrieving memory buffer

Left	: cursor to left
Righ	: cursor to right
Up	: cursor upward
Down	: cursor down
Ctrl-Z	: one line down
Ctrl-E	: one line up

TAB	: travels HEX and ASCII section
ESC	: back to menu
Pgup	: page up
Pgdn	: page down
Home	: to the beginning of line
End	: to the end of line
Ctrl-Pgup	: to the beginning of the entire buffer
Ctrl-Pgdn	: to the end of the entire buffer
Ctrl-Home	: to the beginning of a screen
Ctrl-End	: to the end of a screen
N or n	: moves the cursor to the address you specified.

8.4.4 Buffer Encryption Table

This menu is used for only limited chips. It loads, saves, and retrieves the encryption table, and also sets the size of the table and the mode. The chips which have the encryption tables are: 87C51, 87C52, 8751BH, 87C52BH, and 87C751.



```
: D
: \UNIPRO3
: 0 Byte
Mhz

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

CHIP AD
Start :
End   :
BUFFER AD
Start : 00000000
End   : 0001FFFF

Macro : LOPS.MAC
Disk space : 921600

evice
FACTURER
EL
010A
ORMAT
te
2.START: 0
CHECKSUM: 0000
```

Figure 8-10. The Buffer Encryption Table menu

Load
Looads the file from disk to encryption table.

Save
Saves the file of the encryption table to the disk.

View/Edit
The contents of encryption table is converted as a 32 hex bytes. Move the cursor with the arrow keys, and input the desired HEX values. Regardless of the table's size, it has a 32 byte output.

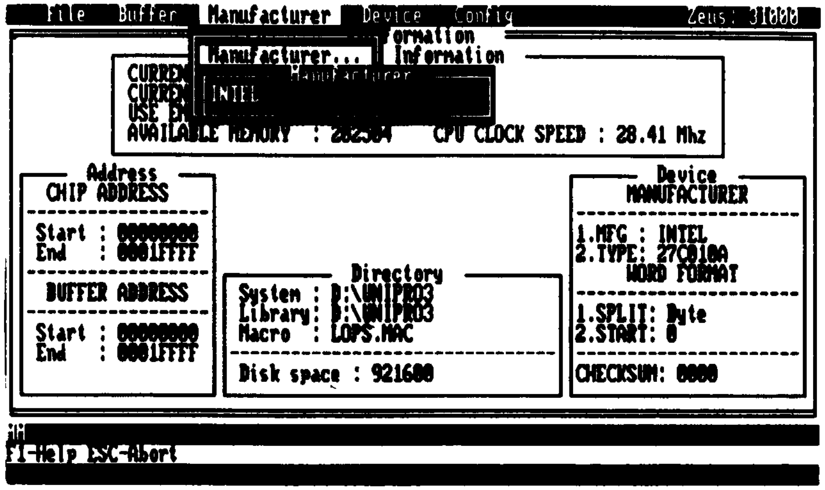
Encryption mode
When the encryption mode is on, it reads the exclusive NORed value between the encryption table and the data of the chip.

Table size
The maximum size of the encryption table is a 32 byte; a 20 byte for hex. Furthermore, the encryption table only contains 16 byte and 32 byte; the input has to be set as 10 or 20 (1 byte) by hex when selecting the size of table.

8.5 The Manufacturer Menu

The user can define the chip to be used, select the manufacturer and type of the chip, and modify the flow of the chip in the submenu "Customization."

The manufacturer menu includes three commands used for selecting a device to be programmed and for the customization of the device.



```
CURREN
CURREN
USE EN
AVAILABLE MEMORY   : 282504     CPU CLOCK SPEED : 28.41 Mhz

     Address                              Device
  CHIP ADDRESS                         MANUFACTURER
-----------------                 ---------------------
Start : 00000000                  1.MFG : INTEL
End   : 0001FFFF                  2.TYPE: 27C010A
-----------------                        WORD FORMAT
 BUFFER ADDRESS        Directory  ---------------------
-----------------  System : D:\UNIPRO3  1.SPLIT: Byte
Start : 00000000   Library: D:\UNIPRO3  2.START: 0
End   : 0001FFFF   Macro  : LOPS.MAC  ---------------------
                   Disk space : 921600  CHECKSUM: 0000

F1-Help ESC-Abort
```

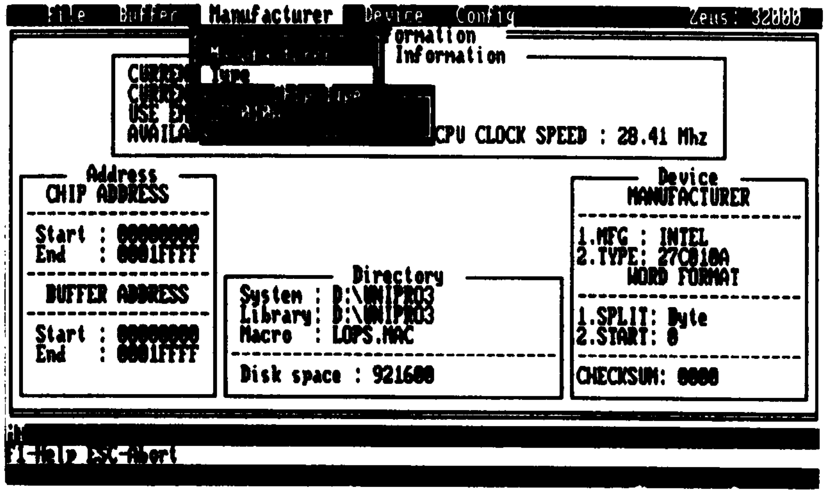
Figure 8-11. The Manufacturer menu.

8.5.1 Manufacture Manufacture

The manufacturer command is used to select the manufacturer of device to be programmed. This command selects the type of device to be programmed.

8.5.2 Manufacturer type

Select the chip type.

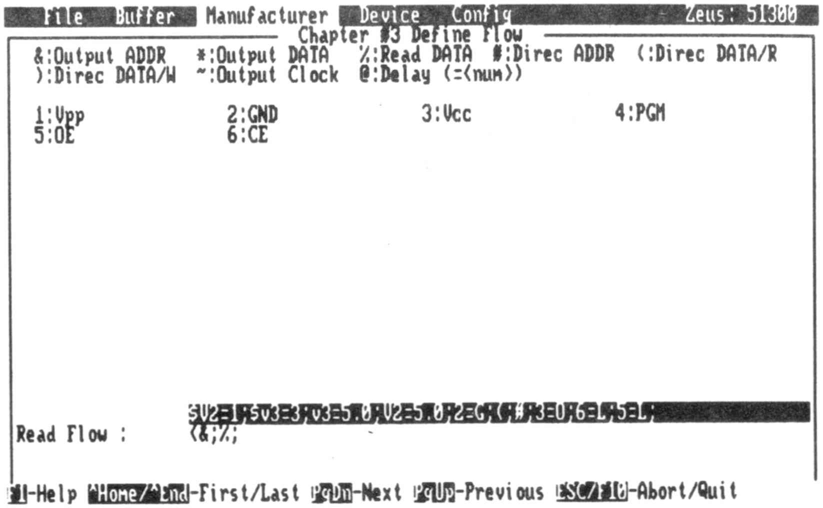


```
CURREN
CURREN
USE EN
AVAILA                          CPU CLOCK SPEED : 28.41 Mhz

     Address                              Device
  CHIP ADDRESS                         MANUFACTURER
-----------------                 ---------------------
Start : 00000000                  1.MFG : INTEL
End   : 0001FFFF                  2.TYPE: 27C010A
-----------------                        WORD FORMAT
 BUFFER ADDRESS        Directory  ---------------------
-----------------  System : D:\UNIPRO3  1.SPLIT: Byte
Start : 00000000   Library: D:\UNIPRO3  2.START: 0
End   : 0001FFFF   Macro  : LOPS.MAC  ---------------------
                   Disk space : 921600  CHECKSUM: 0000

F1-Help ESC-Abort
```

Figure8-12. Manufacturer Type submenu

8.5.3 Manufacture Customization

This command can modify the contents of read, program, encryption, and security flow after defining the manufacture and type. As you see in Figure 8-13, it modifies the needed part of the existing flow.

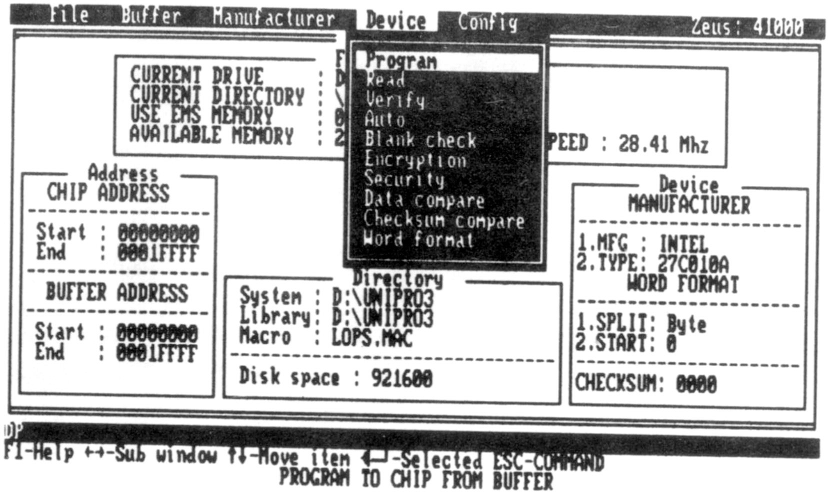


```
                          Chapter #3 Define Flow
 &:Output ADDR   *:Output DATA   %:Read DATA  #:Direc ADDR  (:Direc DATA/R
 ):Direc DATA/W  ~:Output Clock  @:Delay (=<num>)

 1:Vpp            2:GND            3:Vcc            4:PGM
 5:OE             6:CE


                 $V2=1;$V3=3;V3=5.0;V2=5.0;2=G;(;#;3=0;6=L;5=L;
Read Flow :      <&;%;

F1-Help ^Home/^End-First/Last PgDn-Next PgUp-Previous ESC/F10-Abort/Quit
```

Figure 8-13. The Manufacture Customization screen.

8.6 The Device Menu

The device menu contains commands which cause the software to interrelate with the programming module. Included are commands to read the contents of a previously programmed device into the memory buffer, to program a device with the data in the memory buffer, to verify the programming of a device, to perform a blank check, to check the encryption and security bits, to compare the data and checksum, and to set the word format.

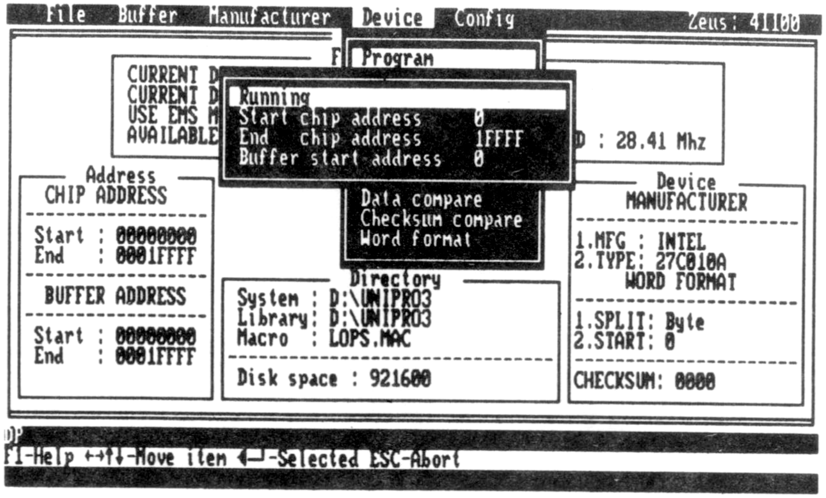


```
 CURRENT DRIVE        : D
 CURRENT DIRECTORY    : \
 USE EMS MEMORY       : 0
 AVAILABLE MEMORY     : 2                 PEED : 28.41 Mhz

    Address                                        Device
 CHIP ADDRESS                                   MANUFACTURER
 ----------------                          ----------------------
 Start : 00000000                          1.MFG : INTEL
 End   : 0001FFFF                          2.TYPE: 27C010A
 ----------------                               WORD FORMAT
 BUFFER ADDRESS             Directory      ----------------------
 ----------------   System : D:\UNIPRO3    1.SPLIT: Byte
 Start : 00000000   Library: D:\UNIPRO3    2.START: 0
 End   : 0001FFFF   Macro  : LOPS.MAC      ----------------------
                    --------------------   CHECKSUM: 0000
                    Disk space : 921600
DP
F1-Help ↔-Sub window ↑↓-Move item ↵-Selected ESC-COMMAND
                    PROGRAM TO CHIP FROM BUFFER
```

Figure 8-14. The Device menu

8.6.1 Device Program

This command downloads the data from the memory buffer to the programming module which in turn programs the device placed in its socket. After programming, the device program command automatically performs a verification.



```
Running
Start chip address      0
End   chip address      1FFFF
Buffer start address    0

CURRENT D
CURRENT D
USE EMS M
AVAILABLE
D : 28.41 Mhz

Address
CHIP ADDRESS
Start : 00000000
End   : 0001FFFF
BUFFER ADDRESS
Start : 00000000
End   : 0001FFFF

Directory
System : D:\UNIPRO3
Library: D:\UNIPRO3
Macro  : LOPS.MAC
Disk space : 921600

Device
MANUFACTURER
1.MFG : INTEL
2.TYPE: 27C010A
   WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000

F1-Help ←→↑↓-Move item ◄┘-Selected ESC-Abort
```

Figure 8-15. The Device Program menu

Running :
Programming and verifying the bytes, it stops executing if an error occur, and display the memory buffer's and data's addresses. The user can easily notice the current processing with the increment of current address, and messages for successful completion. Press the <ESC> key to return to the menu.

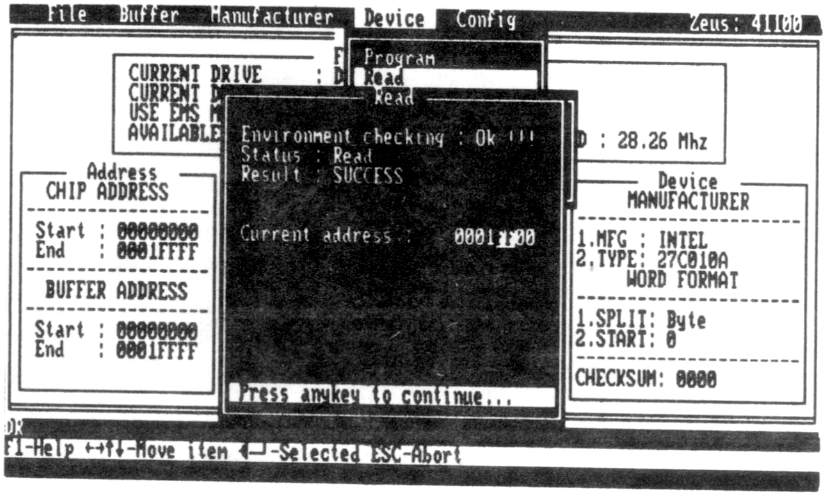
Start chip address :
Defines the starting address. The address is defined by one less hex value than the ending address.

End chip address :
Defines the ending address. The address is defined as the maximum address if it is bigger than the maximum address.

Buffer start address :
Defines start address of the memory buffer to be programmed.

8.6.2 Device Read

The Device Read command copies the contents of the device which is currently located on the programming module into the memory buffer. Once read from the chip, the copied data may be edited, saved to disk or used to program devices.

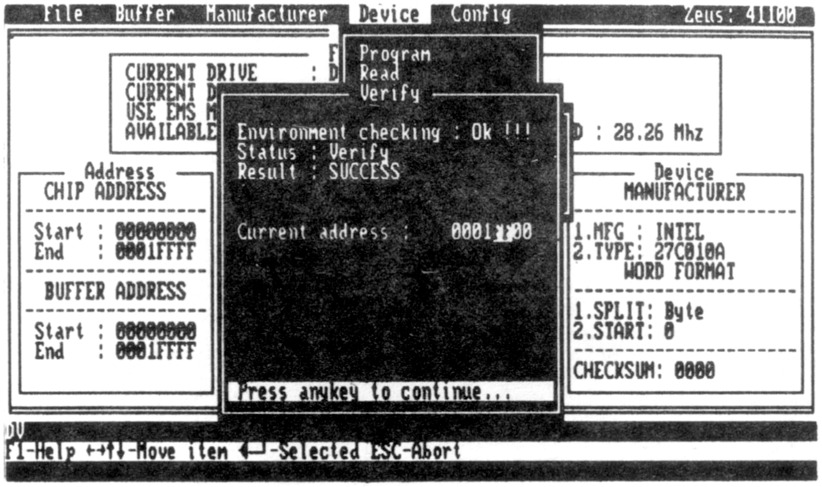


```
Read
Environment checking : Ok !!!
Status : Read
Result : SUCCESS

Current address :   0001[illegible]00

Press anykey to continue...

CURRENT DRIVE     : D
CURRENT D
USE EMS M
AVAILABLE
D : 28.26 Mhz

Address
CHIP ADDRESS
Start : 00000000
End   : 0001FFFF
BUFFER ADDRESS
Start : 00000000
End   : 0001FFFF

Device
MANUFACTURER
1.MFG : INTEL
2.TYPE: 27C010A
   WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000

F1-Help ←→↑↓-Move item ◄┘-Selected ESC-Abort
```

Figure 8-16. The Device Read menu

8.6.3 Device Verify

The Device Verify command allows you to verify that the contents of a device currently located on the programming module match the data in the memory buffer. While the Device Program command verifies at the time of programming, with the Device Verify command you may perform verification at any time. Should an error occur, a message will be displayed with the address which was programmed incorrectly.


File
Buffer
Manufacturer
Device
Config
Zeus: 41100
Program
Read
Verify
Press anykey to continue...
F1-Help ↔↑↓-Move item ↵-Selected ESC-Abort

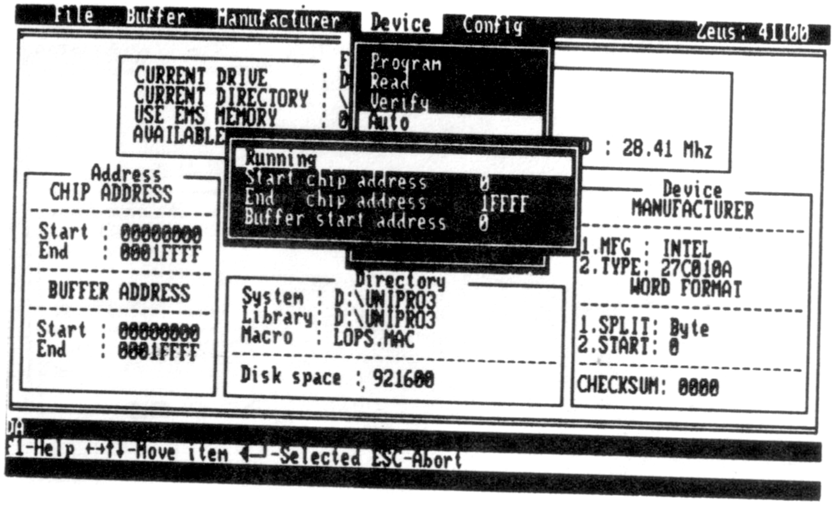
```
CURRENT DRIVE     : D
CURRENT D
USE EMS M
AVAILABLE
                                          D : 28.26 Mhz
Environment checking : Ok !!!
Status : Verify
Result : SUCCESS

Current address :     0001??00

Address
CHIP ADDRESS
Start : 00000000
End   : 0001FFFF
BUFFER ADDRESS
Start : 00000000
End   : 0001FFFF

Device
MANUFACTURER
1.MFG : INTEL
2.TYPE: 27C010A
     WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000
DV
```

Figure 8-17. The Device Verify menu.

8.6.4 Device Auto

The Device Auto command is a time-saver. it automatically runs the following commands in sequence

. Device Blank Check
. Device Program
. Device Verify

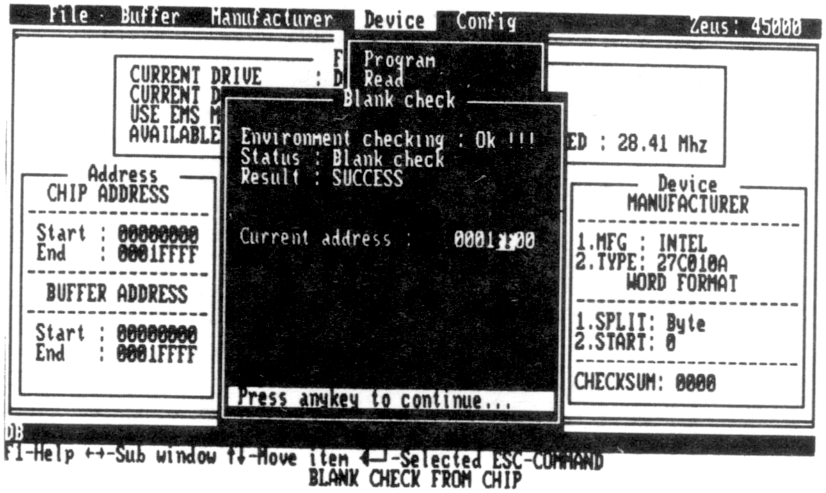


```
CURRENT DRIVE     : D
CURRENT DIRECTORY : \
USE EMS MEMORY    : 0
AVAILABLE
                                          D : 28.41 Mhz
Running
Start  chip address     0
End    chip address     1FFFF
Buffer start address    0

Address
CHIP ADDRESS
Start : 00000000
End   : 0001FFFF
BUFFER ADDRESS
Start : 00000000
End   : 0001FFFF

Directory
System : D:\UNIPRO3
Library: D:\UNIPRO3
Macro  : LOPS.MAC
Disk space :, 921600

Device
MANUFACTURER
1.MFG : INTEL
2.TYPE: 27C010A
     WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000
DA
```

Figure 8-18. The Device Auto Program scree

8.6.5 Device Blank Check

This command checks if the device currently located on the programming module is blank. Should any data be contained in the device, an error message is displayed with the first address that is not blank (has been programmed). The data value of a blank chip is defined at CONFIG ENVIRONMENT BLANK CHARACTER menu. In general, blank characters are defined as "FF" by hex, but for the series of 8748, they are "00".

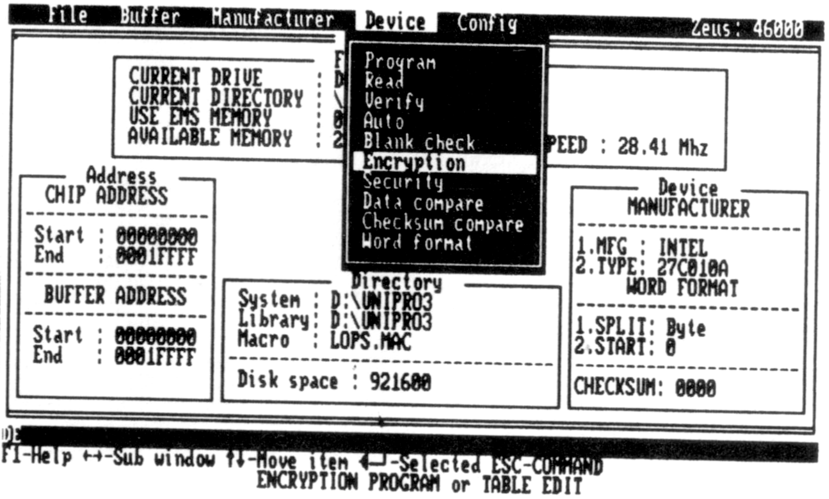


```
CURRENT DRIVE     : D
CURRENT D
USE EMS M
AVAILABLE

Environment checking : Ok !!!
Status : Blank check
Result : SUCCESS

Current address :    0001[illegible]00

Press anykey to continue...

ED : 28.41 Mhz

CHIP ADDRESS
Start : 00000000
End   : 0001FFFF
BUFFER ADDRESS
Start : 00000000
End   : 0001FFFF

MANUFACTURER
1.MFG : INTEL
2.TYPE: 27C010A
   WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000

F1-Help ↔-Sub window ↑↓-Move item ↵-Selected ESC-COMMAND
                BLANK CHECK FROM CHIP
```

Figure 8-19. The Device Blank Check screen.

8.6.6 Device Encryption

The Device Encryption command is used only with some of the single-micro controller chips such as the 87C51, 8752BH, 87C52, and the 875BH. This menu writes the encryption table on the device. After the encryption table has been written, the data will be irrecognizable because the result is "exclusive NORed" between the encryption table and the chip data. Unless there is data in the encryption, this menu will not be activated.

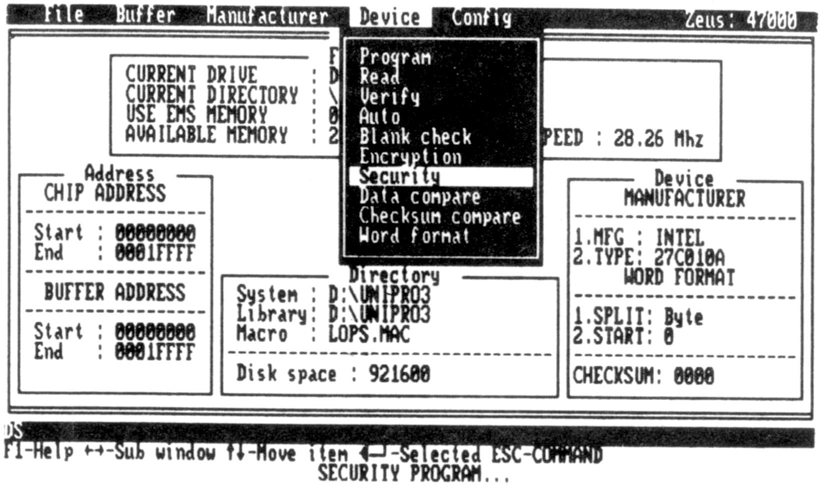


```
CURRENT DRIVE     : D
CURRENT DIRECTORY : \
USE EMS MEMORY    : 0
AVAILABLE MEMORY  : 2

PEED : 28.41 Mhz

CHIP ADDRESS
Start : 00000000
End   : 0001FFFF
BUFFER ADDRESS
Start : 00000000
End   : 0001FFFF

System : D:\UNIPRO3
Library: D:\UNIPRO3
Macro  : LOPS.MAC
Disk space : 921600

MANUFACTURER
1.MFG : INTEL
2.TYPE: 27C010A
   WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000

F1-Help ↔-Sub window ↑↓-Move item ↵-Selected ESC-COMMAND
          ENCRYPTION PROGRAM or TABLE EDIT
```

Figure 8-20. The Device Encryption screen

8.6.7 Device Security

The Device Security sets the security bit on the 8751 (Single chip microcontroller) for the 8751 (single chip). After execution of programming the security bit, it is impossible to read the original contents of the device; therefore, the chip's contents is read as blank. Also, you can't program the device after the Device Security function has been performed.



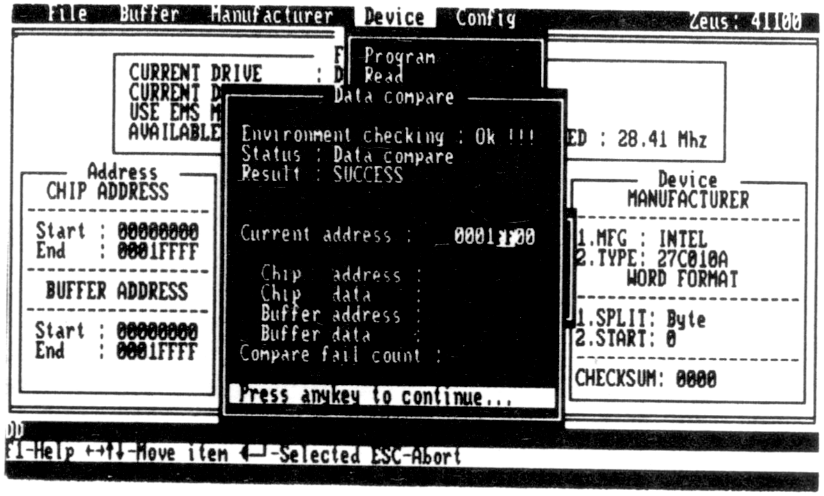
```
CURRENT DRIVE       : D
CURRENT DIRECTORY   : \
USE EMS MEMORY      : 0
AVAILABLE MEMORY    : 2
PEED : 28.26 Mhz

Address
CHIP ADDRESS
Start : 00000000
End   : 0001FFFF
BUFFER ADDRESS
Start : 00000000
End   : 0001FFFF

Directory
System : D:\UNIPRO3
Library: D:\UNIPRO3
Macro  : LOPS.MAC
Disk space : 921600

Device
MANUFACTURER
1.MFG : INTEL
2.TYPE: 27C010A
WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000

DS
F1-Help ←→-Sub window ↑↓-Move item ←┘-Selected ESC-COMMAND
                    SECURITY PROGRAM...
```

Figure 8-21. The Device Security screen

8.6.8 Device Data Compare

The Device Data Compare command allows you to compare the contents of a device and the data in the memory buffer. A file ("Lopsfail.faf") will be created and contained all the discrepancies.

NOTE: to stop in the middle of the execution, press <ESC>. "User's break" will be displayed.



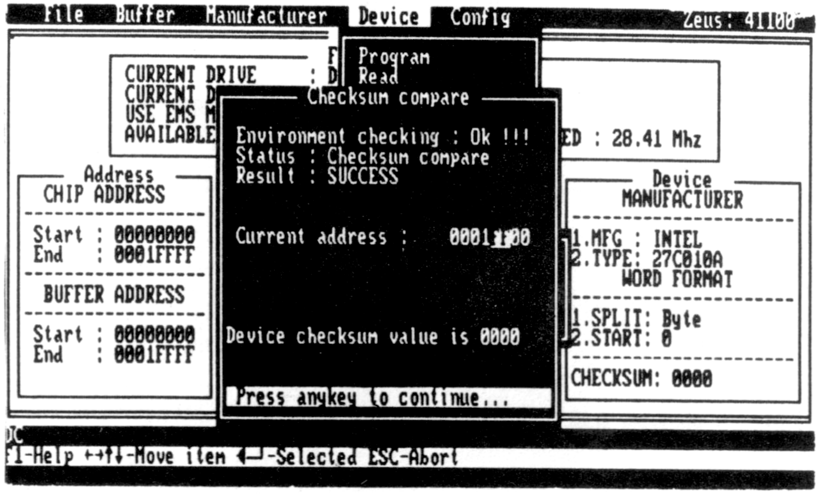
```
CURRENT DRIVE       : D
CURRENT D
USE EMS M
AVAILABLE
ED : 28.41 Mhz

Data compare
Environment checking : Ok !!!
Status : Data compare
Result : SUCCESS

Current address :      0001[illegible]00

  Chip   address :
  Chip   data    :
  Buffer address :
  Buffer data    :
Compare fail count :

Press anykey to continue...

Address
CHIP ADDRESS
Start : 00000000
End   : 0001FFFF
BUFFER ADDRESS
Start : 00000000
End   : 0001FFFF

Device
MANUFACTURER
1.MFG : INTEL
2.TYPE: 27C010A
WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000

DD
F1-Help ←→↑↓-Move item ←┘-Selected ESC-Abort
```

Figure 8-22. The Device Data Compare screen

The only differences between "verify" and "Data Compare" are the file creation and nonstopping comparison in the Data Compare menu.

8.6.9 Device Checksum Compare

The Device Checksum Compare command displays last 2 bytes of the checksum after adding all data bytes in a device. This will not modify or change the content of the current memory buffer.



```
CURRENT DRIVE      : D
CURRENT D
USE EMS M
AVAILABLE
                                   ED : 28.41 Mhz

Checksum compare
Environment checking : Ok !!!
Status : Checksum compare
Result : SUCCESS

Current address :     0001_00

Device checksum value is 0000

Press anykey to continue...

Address
CHIP ADDRESS
Start : 00000000
End   : 0001FFFF
BUFFER ADDRESS
Start : 00000000
End   : 0001FFFF

Device
MANUFACTURER
1.MFG : INTEL
2.TYPE: 27C010A
  WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000
```

Figure 8-23. The Device Checksum Compare screen

8.6.10 Device Word Format

The Device Word Format command is only available for (E)EPROM. It defines the word format : byte, word, and double word.

BYTE : Defines the word format as a unit of 8 bits.

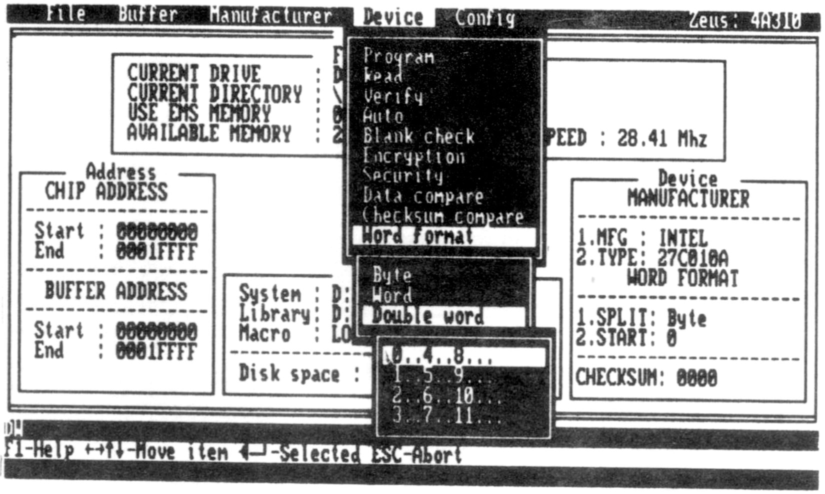
WORD : Defines the word format as a unit of 16 bits. When programming ROM, a unit of 8 bits is programmed. Thus, the sequence of programming "word" should be defined; even or odd will be selected.

> **Even** : The functions read, program, and verify will be executed only on the even address.

> **Odd** : The functions read, program, and verify will be executed only on the odd address.

DOUBLE WORD : Defines word format as a unit of 32 bits. With the same reason described in "B. WORD", the sequence of programming "DOUBLE WORD" will be defined below:

. 0..4..8..
. 1..5..9..
. 2..6..10..
. 3..7..11..



```
CURRENT DRIVE     : D
CURRENT DIRECTORY : \
USE EMS MEMORY    : 0
AVAILABLE MEMORY  : 2
                                   PEED : 28.41 Mhz

Address
CHIP ADDRESS
Start : 00000000
End   : 0001FFFF
BUFFER ADDRESS
Start : 00000000
End   : 0001FFFF

System : D:
Library: D:
Macro  : LO
Disk space :

Device
MANUFACTURER
1.MFG : INTEL
2.TYPE: 27C010A
  WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000
```

Figure 8-24. The Device Word Format menu

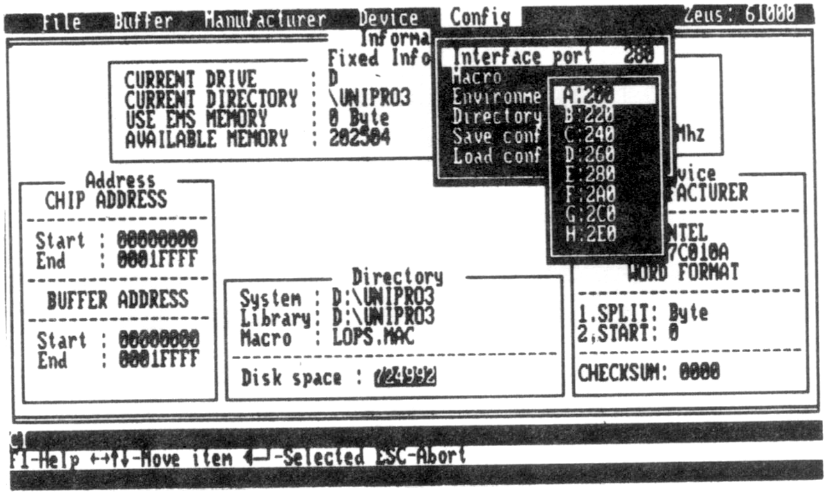
8.7 The Config Menu

The Config menu includes commands for interface port and directory selection and for Macro file loading.

8.7.1 Config Interface Port

The Config Interface Port command is used to change the port address for the programmer interface card which is installed in your computer. Normally, the port address need not be changed unless the jumper setting on the interface card is changed.

If error occurs, refer to section 3.3 for "Installation and Jumper Position."

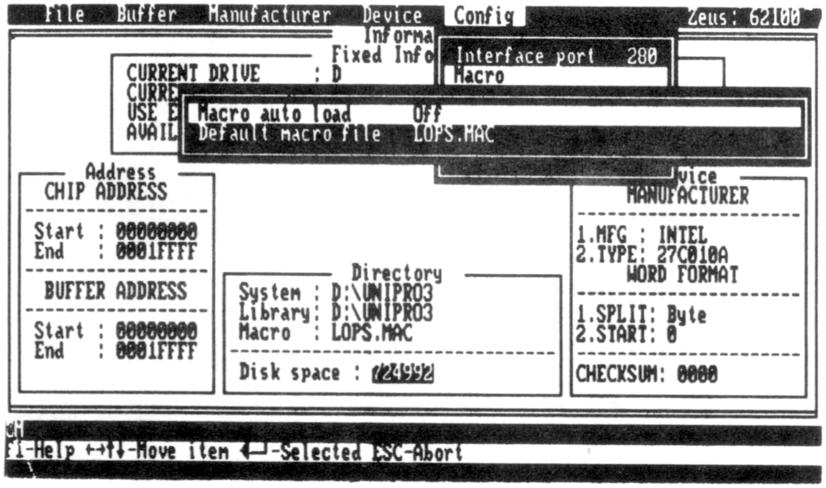


```
CURRENT DRIVE     : D
CURRENT DIRECTORY : \UNIPRO3
USE EMS MEMORY    : 0 Byte
AVAILABLE MEMORY  : 202504

CHIP ADDRESS
Start : 00000000
End   : 0001FFFF
BUFFER ADDRESS
Start : 00000000
End   : 0001FFFF

System : D:\UNIPRO3
Library: D:\UNIPRO3
Macro  : LOPS.MAC
Disk space : 724992

NTEL
7C010A
WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000
```

Figure 8-25. The Config Interface Port menu

8.7.2 Config Macro

The Config Macro command specifies the default macro file name, and loads it automatically. There are two submenus: Macro auto and Default macro file.

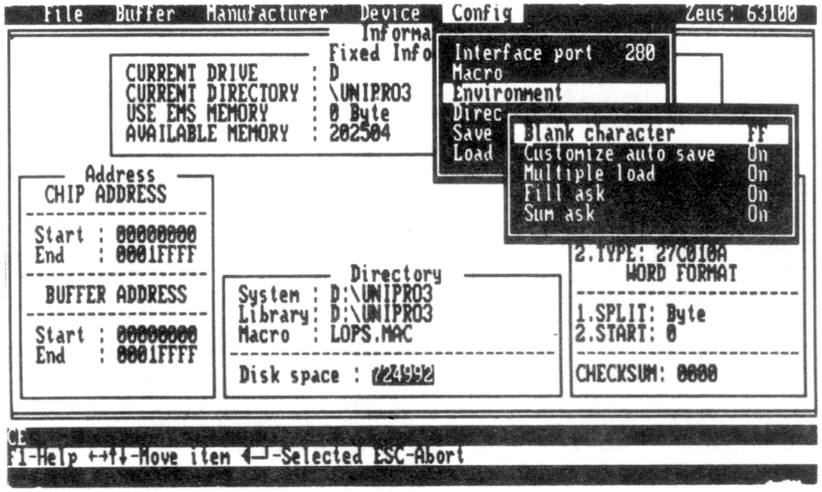


```
CURRENT DRIVE : D
CURRE
USE E
AVAIL

Macro auto load      Off
Default macro file   LOPS.MAC

CHIP ADDRESS
Start : 00000000
End   : 0001FFFF
BUFFER ADDRESS
Start : 00000000
End   : 0001FFFF

System : D:\UNIPRO3
Library: D:\UNIPRO3
Macro  : LOPS.MAC
Disk space : 724992

1.MFG : INTEL
2.TYPE: 27C010A
   WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000
```

Figure 8-26. The Device Macro menu

8.7.3 Config Environment

The Config Environment command sets up the enviornment for automatic save, multiple load, blank character setting, buffer filling, and Checksum generation.



```
Fixed Info
CURRENT DRIVE        : D
CURRENT DIRECTORY    : \UNIPRO3
USE EMS MEMORY       : 0 Byte
AVAILABLE MEMORY     : 202504

Address
CHIP ADDRESS
Start : 00000000
End   : 0001FFFF
BUFFER ADDRESS
Start : 00000000
End   : 0001FFFF

Directory
System : D:\UNIPRO3
Library: D:\UNIPRO3
Macro  : LOPS.MAC
Disk space : 724992

2.TYPE: 27C010A
   WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000
```

Figure 8-27. The Config Environment menu

8.7.3.1 Blank Character

Defines the blank character. In general, the blank character is FF in hex file formats, but is 00 in Single Chip Microcontrollers (8748 series).

8.7.3.2 Customize Auto Save

If this command is on, it automatically saves the data as soon as it gets out of the program flow in customization.

8.7.3.3 Multiple Load

This command is used in loading multiple binary files. When this mode is on, multiple file loading is possible.

8.7.3.4 Fill Ask

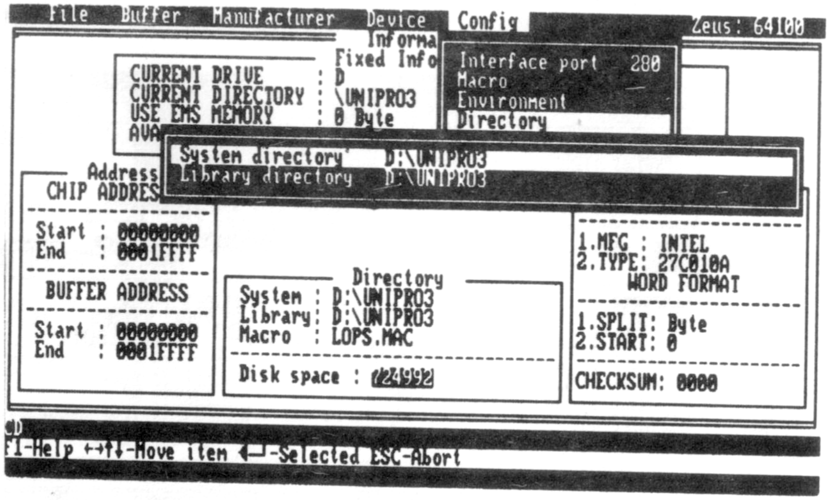
If this mode is on, partial filling is possible, and the submenu of "Fill" will be shown for the portion of the buffer to be filled.

8.7.3.5 Sum Ask

If the Sum Ask mode is on, partial checksum is possible, and the submenu of checksum will be shown.

8.7.4 Config Directory

This command shows directories of the system files and library files. You can change directories after typing <Enter> on the current directory.



```
System directory    D:\UNIPRO3
Library directory   D:\UNIPRO3

Fixed Info
CURRENT DRIVE        : D
CURRENT DIRECTORY    : \UNIPRO3
USE EMS MEMORY       : 0 Byte

Address
CHIP ADDRESS
Start : 00000000
End   : 0001FFFF
BUFFER ADDRESS
Start : 00000000
End   : 0001FFFF

Directory
System : D:\UNIPRO3
Library: D:\UNIPRO3
Macro  : LOPS.MAC
Disk space : 724992

1.MFG : INTEL
2.TYPE: 27C010A
   WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: 0000
```

Figure 8-28. The Config Directory menu

System directory
Shows where the executable file of LOPS is located. The user can change the system directory.

Library directory

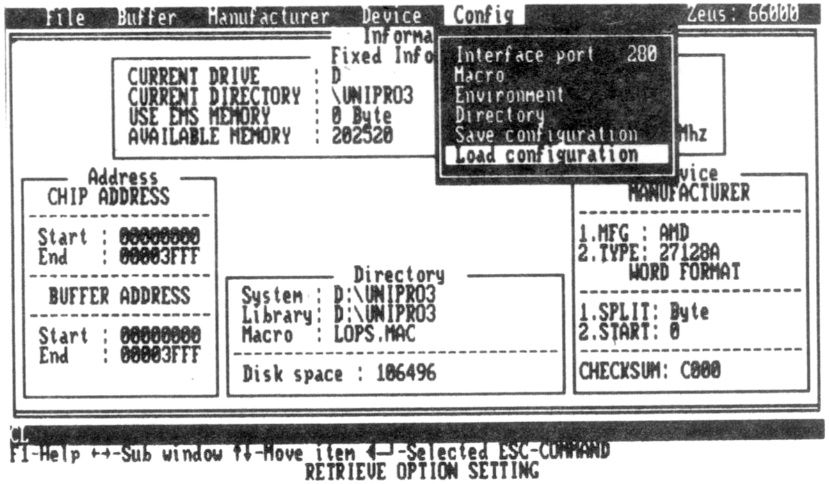
Shows where the library is located; it is also changable.

8.7.5 Config Save Configuration

This command saves all the setup, and writes them into a file called "LOPSCONF.CFG". You can load the saved setup file back for frequent usages.

8.7.6 Config Load Configuration

The Config Load Configuration retrieves the setup from the file (LOPS.CFG) saved by **Save Configuration** command.



```
CURRENT DRIVE      : D
CURRENT DIRECTORY  : \UNIPRO3
USE EMS MEMORY     : 0 Byte
AVAILABLE MEMORY   : 202520

CHIP ADDRESS
Start  : 00000000
End    : 00003FFF

BUFFER ADDRESS
Start  : 00000000
End    : 00003FFF

System : D:\UNIPRO3
Library: D:\UNIPRO3
Macro  : LOPS.MAC
Disk space : 106496

MANUFACTURER
1.MFG : AMD
2.TYPE: 27128A
WORD FORMAT
1.SPLIT: Byte
2.START: 0
CHECKSUM: C000

F1-Help ←→-Sub window ↑↓-Move item ←┘-Selected ESC-COMMAND
RETRIEVE OPTION SETTING
```

Figure 8-29. The Config Load configuration screen

9 LOPS Commands

9.1 Introduction

This chapter will be devoted to the explanation of commands (ch 9.2), and reserved words for the creation of macro or batch files (ch 9.3).

9.2 Commands-Syntax and Explanation

This section describes each command used to add devices in the library and describes the syntax. The available commands include the followings:

1. &
2. $V
3. #
4.)
5. (
6. ~
7. *
8. D=data
9. %
10. !

11. <

12. R

13. W

14. @

15. P=address

16. Pin assignment

17. IR, IP, IS, IE

Note : *There are pin numbers. Those pin numbers are assigned numbers by the system. These pin numbers are very confusing, because we use the pin number to symbolize Vcc, Vpp, GND, OE, CE, etc. So be sure not to confuse the symbols of major pins with the physical pins of a chip.*

1. &

This command produces the current address to the chip.

2. $V(1,2,3)=pin number

There are three voltage sources in the programmer. They are defined as V1, V2, and V3. This reserved word only indicates connection, but doesn't turn on the voltage sources. V3 ranges from 5V to 10V, and V1 and V2 range from 5V to 25V. Generally V3 will be used for Vcc, V1, and V2 will be used for programming chips.

Examples:
If the pin numbers are defined as below.
1: VPP 2: GND 3:Vcc

Flow:
```
$V2=1;$V3=3;V2=12.5;V3=5;
```

Explanation:
```
$V2=1     ; Voltage source V2 will be connected to Vpp.
$V3=3     ; Voltage source V3 will be connected Vcc.
V2=12.5   ; Sets the voltage of V2 equal to 12.5V
```

```
V3=5      ; sets the voltage of V3 equal to 5V.
```

3.

This command sets the port of the programmer which is connected to address pins to output.

4.)

This command sets the port of the programmer which is connected to data pins to output.

5. (

This command sets the port of the programmer which is connected to data pins to input.

6. ~

This command connects clock to the chip. The hardware of the Unipro has 2 MHz clock generator. As soon as this command is issued the clock generator of Unipro will be connected to the designated clock pin.

7. *

This command writes the content of the memory buffer into the device.

8. D = Data

This command will be used when the user wants to program the special data which is not in the buffer.

Example
```
D=CA;
```

9. %

This command will read the chip's data of the current address into the buffer.

10. !

This command will output the address of the current page.

11. <

This indicates the beginning of the loop. This is like a "if" or "for" loop. The code from this point to the end of the flow will be repeated. No end mark will be indicated.

12. R Repeating times { }

This command will repeat the code inside of the brackets. Also the command can control the number of the repetitions.

13. W pin number [Initial value, Delay time, Final value]

This important command is used to control the duration of the (programming) pulse. This will generate the logic high or low voltage initially, and hold the value for the designated amount time (in microsecond), and generate the final value. The maximum delay time is 65535 usec, and the minimum delay time depends on the CPU clock speed. On a XT computer of 4.77Mhz, the minimum delay time is 10usec.

NOTE: After this command, no language separator (;) will be required.

14. @

This command also delays the time. But this is not limited to a certain pin. The default unit of the delay time is micro seconds. The delay time ranges 0 to 65535. This can be use more than once to achieve the delay which is greater than the 65535 micro seconds.

Example: @=100; Delays for 100 us.

15. P = Address pin number

P defines the pin for paging among the address pins.

16. Pin number=value

This command designates the status of the voltage of the system pins.

H : Logic high.
L : Logic low.
O : On.
F : Off.
G : Connects to ground.

Example:

PIN ASSIGNMENTS
1:Vcc 2:Vpp 3:CE

Flow:

$V3=1;V3=5;1=O;
The voltage source V3 will be connected to Vcc and turned to 5V.

17. IR, IP, IS, IE

These commands will bring in the flow of Read, Program, Security, and Encryption wave code to patch it in the current position.

9.3 Reserved Words for macros and bath files

9.3.1 Introduction

This section will explain the reserved words that are the basic elements of macro and batch files, which will help users automate the programming process.

9.3.2 Macro

Format:

```
macro   key to be assigned
        reserved word 1
        reserved word 2
              :
              :
        reserved word n
        ENDM
```

macro: Declares the beginning of macro.

key: A sequence of commands will be assigned to one of the 10 function keys (F1....) and their combinations with <Ctrl> or <Shift> or <Alt>. Thus total 40 keys are available for macro files.

syntax: One reserved word per line.

reserved word: This will be explained in 9.3.4.

ENDM: Indicates the end of a macro.

EX)

```
macro   F5
        FL A F TEST.EXE R
        MM INTEL 27128
        DP R -N10
        ENDM
```

This example will assign F5 to load file "test.exe", choose Intel 27128, and program 10 times.

9.3.3 Batch Files

These files are little different to macro files. There is no limit for the number of reserved words, and there is no need to assign keys. Batch files will be written up and named with ".bat" at the end. One reserved word per line is allowed. After saving, if the file name is "test.bat", type in DOS as follows:

```
LOPS -Btest.bat
(B means batch)
```

As soon as LOPS is invoked, the reserved words will be performed sequentially.

EX)

```
c:>copy con test.bat
FL A F TEST.BAT
MM INTEL 27128A
DP R -N10
^Z
C:>LOPS -BTEST.BAT
```

This example will assign F5 to load file "test.exe", choose Intel 27128, and program 10 times.

9.3.4 Reserved Words

Generally, "-N" means the number of repetitions, "R" means execute, and "-B" means batch files. This will be repeated further in this manual.

1. File load

Loads the data to be programmed in the current buffer. You can choose the format desired. Formats include Intel (Extended) HEX, Motorola (S1, S2, S3,), and Tektronix (Extended) HEX.

SUBMENUS

A: Binary file B: Intel (Extended)
C: Motorola S1 D: Motorola S2
E: Motorola S3 F: Tektronix
G: Extended Tektronix

Command format:

```
FL A [F/S] [R]
```

FL [B/C/D/E/F/G] [filename]

2. File Load A (Binary)

Loads the binary data into the buffer from the disk. Multiple loading is possible if the Config-Environment-Multiple Load menu is set "on".

Sub menus:
File name
Start address
Running

Command format:
FL A F [file name] [R]
FL A S [start address] [R]

3. File Load A (binary) Running

Loads the file after all specifications have been made. If this is written at the end of any command format, the command will be executed; otherwise, it will only be displayed.

4. File Load A (Binary) File name

Indicates the file name which takes wildcard characters. In case you use wildcard characters, all the files that meet the specification will show up. To proceed, select a file.

Command format:
FL A F [file name]

5. File Load A (Binary) Start address

When you use binary files, you need to indicate the start- address when referring to different files.

Command format:
FL A S [start address]

6. File Load B (Intel HEX, Extended)

Loads Intel (Extended) HEX-file data from the disk to the buffer.

Command format:
FL B [filename]

7. File Load C (Motorola S1)

Loads Motorola S1 HEX-file data from the disk to the buffer.

Command format:
FL C [filename]

8. File Load D (Motorola S2)

Loads Motorola S2 HEX-file data from the disk to the buffer.

Command format:
FL D [filename]

9. File Load E (Motorola S3)

Loads Motorola S3 HEX-file data from the disk to the buffer.

Command format:
FL E [filename]

10. File Load F (Tektronix HEX)

Loads Tektronix HEX-file data from the disk to the buffer.

Command format:
FL F [filename]

11. File Load G (Tektronix Extended HEX)

Loads Tektronix Extended HEX-file data from the disk to the buffer.

Command format:
FL G [filename]

12. FILE MACRO

Loads the macro file that will be memorized by the software. Micro combines menu commands into one, and assigns it to a function key.

13. File Save

Saves the file to the buffer which has been created, modified, or read from a chip onto a specified disk.

SUBMENUS

A: Binary file B: Intel (Extended)
C: Motorola S1 D: Motorola S2
E: Motorola S3 F: Tektronix
G: Extended Tektronix

Command format
FL [A/B/C/D/E/F/G] [F/S/B] [R]

14. File Save A (Binary)

Saves binary data from the buffer to the disk.

Sub menus:
Running
File name
Start address
Buffer size

Command format:
FS A [F/S/B] [file name/start address/ address/buffer size] [R]

15. File Save A (Binary) File name

Specify the file name

Command format:
FS A F [file name]

16. File Save A (Binary) Start address

In this section "start address" is specified.

Command format:
FS A S [start address]

17. File Save A (Binary) Buffer size

In this section "buffer size" is specified.

Command format:
FS A B [buffer size]

18. File Save B (Intel, Intel Extended)

Saves the data file of the Intel (Extended) HEX file from the buffer to the disk.

Sub menus:
Running
File name
Start address
Buffer size

Command format:
FS B[F/S/B] [file name/start address/buffer size] [R]

19. File Save C (Motorola S1)

Saves the data file of the Motorola S1 HEX-file from the buffer to the disk.

Submenu:
Running
File name
Start address
Buffer size

Command format:
[F/S/B] [file name/start address/buffer size] [R]

20. File Save D (Motorola S2)

Saves the data file of the Motorola S2 file from the buffer to the disk.

Sub menus:

Running
File name
Start address
Buffer size

Command format:

FS D [F/S/B] [file name/start address/buffer size] [R]

21. File Save E (Motorola S2)

Saves the data file of the Tektronix Extended HEX file from the buffer to the disk.

Sub menus:

Running
File name
Start address
Buffer size

Command format:

FS E [F/S/B] [file name/start address/buffer size] [R]

22. Save F (Tektronix)

Saves the data file of the Tektronix Extended HEX file from the buffer to the disk.

Sub menus:

Running
File name
Start address
Buffer size

Command format:

FS F [F/S/B] [file name/start address
/buffer size] [R]

23. File Save G (Tektronix Extended)

Saves the data file of the Tektronix Extended HEX file from the buffer to the disk.

Sub menus:

Running
File name
Start address
Buffer size

Command format:

FS G [F/S/B] [file name/start address/buffer size] [R]

24. File User Screen

This command will be used to review the screen before you enter the programmer's menu.

25. File DOS

This command will save you the trouble from exiting the software when you want to execute DOS commands. To return to the menu after executing some commands, type "exit."

Caution: If you invoke other software, be sure to close the software before typing "exit" (to return to the programmer's menu).

27. File QUIT

Returns to the operation system.

Command format:

FQ

28. Buffer FILL

Fills the desired value in HEX on the specified portion of the buffer.

Sub-menus:
Running On
Start Address
End Address
Character

Command format:
BF S [start address]
BF [S/E/C] [start address/end address/character] [r]

29. Buffer Fill Running

Starts executing the command. If this buffer is written at the end of any command format, the command will be executed; otherwise, it will only be displayed.

30. Buffer Fill Start

The address of the beginning is specified for filling.

Command format:
BF S [start address]

31. Buffer Fill End

The address of the end is specified for filling.

Command format:
BF E [end address]

32. Buffer Fill Char

The HEX value to be filled is specified. The range of HEX value is from 0 to FF.

Command format:
BF C [character]

33. Buffer Check Sum

Sums all the values on the specified portion of the buffer and displays the two least significant HEX digits to accommodate easy comparisons.

submenus:
Running
File name
Start address

Command format:
BC [S,E] [start address/end address] [r]

34. Buffer Check Sum

Starts executing the command. If "R" is written at the end of any command format, the command will be executed; otherwise, it will only be displayed.

35. Buffer Check Start

The start-address is specified for "check sum."

Command format:
BC S [start address]

36. Buffer Check End

The address of the end is specified for "check sum."

Command format:
BC E [end address]

38. Buffer Encryption Table

In this menu, the data in the chip is read and "exclusive nor" is performed and displayed when the encryption mode is on. It has all the same functions as the main buffer.

Submenu:

Load
Save
View/Edit
Encryption
Table size

Command format:

```
BE [L/S/E/T] [file name/file name/on or off/table size]
```

39. Buffer Encryption Load

Loads the encryption file name.

Command format:

```
BE L [file name to load]
```

40. Buffer Encryption Save

Saves the encryption file in the disk.

Command format:

```
BE S [file name to save]
```

41. Buffer Encryption View/Edit

The encryption table will be viewed and edited by pressing the arrow keys. HEX values will be used, and regardless of the size of the table, 32 bytes will be the output.

42. Buffer Encryption Mode

This will activate or deactivate the encryption mode. If the mode is on, the data is read, and ex-nor is performed with the encryption table. Then, the result is displayed in the buffer. When programming other than 87C51, 87C252, 8752BH, and the 8751BH, please set the encryption mode to "off."

Command format:

```
BE E [on or off]
```

43. Buffer Encryption Table

The maximum size of the encryption table is 32 bytes (20 in HEX). There are only 2 types in table sizes: 16-bytes or 32-bytes. Thus, when you specify only two values in HEX, 10 and 20 will be allowed.

Command format:

```
BE T [10 or 20]
```

44. Manufacturer Manufacture

Defines the manufacturer. To see the list of all the companies, type "*", and press the carriage return key; or, specify the name of a company and press <Return>.

Command format:

```
MM [manufacture name]
```

NOTE: After a user selects a manufacturer, the software will select the chip type automatically.

45. Manufacturer Type

After the selection of the manufacturer, the device type is selected.

Command format:

```
MT  [type name]
```

46. Manufacturer Customization

Edits the Read, Program, Encryption, Security Flow. For example, the customization will let you change Vpp, Vcc, the programming pulse width, and all the chip control pins. The syntax will be explained in a later Chapter. Some languages are printed and explained in the upper section of the screen.

47. Device Program

Programs the data of the buffer into the chip mounted on the socket. Choose "Running" and press <Enter>. The entire chip will be programmed, but it is only possible to program part of the device.

Submenus:
Running
Start chip address
End chip address
Buffer start address

Command format:

P [S/E/B] [chip start address, chip end address, buffer start address]

48. Device Program Running

Start executing. If "R" is written at the end of any command format, the command will be executed; otherwise, it will only be displayed.

49. Device Program Start

Specifies the beginning of the address.

Command format:

DP S [chip start-address]

50. Device Program End

Specifies the end of the chip-address.

Command format:

DP E [chip end address]

51. Device Program Buffer

Specifies the beginning of the buffer.

Command format:

DP B [buffer start address]

52. Device Read

This function reads in the chip's data mounted on the socket, reads it into the buffer, and displays the "check-sum." A partial reading of the chip is possible by specifying the portion of the address to be read. To re- serve the current memory of the buffer, you can change the beginning of the buffer address.

Sub-menus:
Running
Start Chip Address
End Chip Address
Buffer Start Address

Command format:

DR [S/E/B] [chip start address, chip end address, buffer start address]

53. Device Verify

Compare the buffer and the content of the chip; it displays the difference in the screen, and quits when an error occurs.

Sub-menus:
Running
Start Chip Address
End Chip Address
Buffer Start Address

Command format:

DV [S/E/B] [chip start address, chip end address, buffer start address]

54. Device Auto

Performs the entire programming sequence automatically; Blank Check, Program, and Verify will be performed, respectively. If needed, Encryption and Security will be executed.

Sub-menus:
Running
Start Chip Address
End Chip Address
Buffer Start Address

Command format:

DA [S/E/B] [chip start address, chip end address, buffer start address]

56. Device Blank

Checks whether the chip is blank or not; the FF (or 00 for 87 series single chip microcontrollers) in the data equals a blank chip.

Command format:

DB

57. Device Encryption

Used only in some of single chips, such as the 87C51, 8752BH, 87C52, and the 8751BH micro-controllers. This menu writes the encryption table on the chip. After this menu is performed, the read data will be Ex-NORed with the encryption table.

Command format:

DE

58. Device Security

Secures the chip from being copied. After security is executed, the chip will be read as blank, and will not be reprogrammed.

Command format:

DS

59. Device Data Compare

Compares the data of the chip and the buffer and writes the discrepancies into the file "lopsfail.faf." To view the difference, exit the program, and find "lopsfail.faf" in the same directory. In the file, the data will be displayed in the following format:

Device Address Device Data Buffer Address Buffer Data

Note: You can press <Esc> to stop while comparing.

Sub-menus:

- Running
- Start Chip Address
- End Chip Address
- Buffer Start Address

Command format:

DB [S/E/B] [chip start address/chip end adress/buffer start address]

60. Device Check-sum Compare

All the chip's data will be read and summed. The last two bytes will be displayed without disturbing the data of the current buffer.

Sub-menus:

- Running
- Start Chip Address
- End Chip Address
- Buffer Start Address

Command format:

DC [S/E/B] [chip start address/chip end address/buffer start address]

61. Device Word format

This menu is used to select 8, 16, and 32-data byte modes to read and program.

Sub-menus:

- Byte
- Word
- Double Word

Command format:

DW B
DW W [E/O]
DW [0/1/2/3]

62. Device Word-format Byte

This menu is selected when an 8-bit EPROM is programmed.

Command format:

DW B

63. Device Word Format Word Even/Odd

In this mode, two-8 bit EPROMs can be programmed by selecting EVEN or ODD. This device programs the chip buffer's even-address or odd-address.

Command format:

EVEN - DW W E
ODD - WW O

64. Config Interface

Selects the port-address to match the jumper setting in the interface card. The normal jumper position is port 5, and its equivalent HEX value is 280. If the port settings for the interface and the software are not matching, "Communication Error" will be displayed.

Submenus:

a: 200
b: 220
c: 240
d: 260
e: 280
f: 2A0
g: 2C0
h: 2E0

Command format:

CI [A/B/C/D/E/F/G/H]

65. Config Macro

Deals with macro files which can be written and used for automation.

Submenus:

Macro auto load
Default macro file

Command format:

CM [A/D] [ON or OFF/macro file name]

66. Config Macro Macro-auto-save

With any default macro file defined, you can selecting the Macro Auto Load to "ON", and the same configuration will allow the use of this macro file when running LOPS.

Command format:

CM A [ON or OFF]

67. Config Macro Default-macro-file

Specifies the default macro file. When this defined and macro auto-load is "ON," a user can use this defined macro file whenever running LOPS.

Command format:

CM D [file name]

68. Config Environment

Sets up the environment for automatic saving, multiple loading, and the default blank characters.

Sub-menus:

Blank Characters
Customize Auto Save
Multiple Load
Fill Ask
Sum Ask

Command format:

CE [B] [character]
CE [C/M/F/S] [ON or OFF]

69. Config Environment Blank

Sets the standard characters for comparison. In general, this should be set "FF", but for the 8748 series, this should be set for "00". Only "00" and "FF" will be allowed.

Command format:

CE [B] [character]

70. Config Environment Customize

If any modified program that is saved automatically without asking the program to save, then this mode is on.

Command format:

CE C [ON or OFF]

71. Config Environment Multiple

This mode should be on whenever a user deals with binary files; it will make enable the loading of multiple binary files.

Command format:

CE M [ON or OFF]

72. Config Environment Fill-ask

With this mode, partial filling is possible. With the mode on, the sub-menu for "Fill" in the "Buffer" section will be displayed. No sub-menu will be displayed if the mode is "OFF."

Command format:

CE [B] [character]
CE F [ON or OFF]

73. Config Environment Sum-ask

With this mode, partial check sum is possible. With the mode on, the sub-menu for "Check sum" in the "Buffer" section will be displayed; otherwise, there will be no sub-menu displayed.

Command format:

CE S [ON or OFF]

74. Config Directory

Specifies the directory where the LOPS files and libraries are able to be executed.

Submenu:

System directory
Library directory

Command format:

CD [S/L] [system directory/library directory]

75. Config Directory System

Specifies the directory where the LOPS files are to be executed.

Command format:

CD S [system directory]

76. Config Directory File

Specifies the directories that contain the "Library Directory."

Command format:

CD L [system directory/library directory]

77. Config Save-configuration

Saves the manufacturer, the chip, port-address, and the set-up of all the "config" menu in the file "LOPSCONF.CFG."

Command format:

CS

78. Config Load-configuration

Loads the file which has been saved for the specific set-up, and reconstructs the set-up for the continued task.

Command format:

CL [file name]

From this point on, until the end of the section, the library's menu will be explained. The library's menu is for the advanced version; for the regular version, these menus will not be available.

79. Library Make

Defines the manufacturer and the name of a chip, and assigns the pins. Also, it will let the user to define the programmer's algorithms (Read Flow, Program Flow, Encryption Flow, and Security Flow). After the algorithms are saved, the library make will show up in the specified or created manufacturer's menu.

Command format:

LM [name of manufacture]

80. Library Update

Updates the existing programmer's algorithms and voltages.

Command format:

LU [name of manufacture]

81. Library Delete

Deletes the existing file for the chip which is non-existent. Please be cautious not to erase any valuable file.

Command format:

LD [name of manufacture] [type of chip]

82. Library Copy

Instead of using "Make" to prepare a file from scratch, this menu is used to create new program files in case a target file is similar to the file of the existing chip. Library Copy copies a file from a manufacturer's specific chip to a different file, under a different chip name.

Command format:

LC [source manufacture] [type]
[target manufacturer] [type]

A The JEDEC File Format

A.1 Introduction

XELTEK's UNIPRO software and hardware are fully compatible with the JEDEC file format. The JEDEC file format is a standard data format between data preparation systems, and programmable logic device programmers. The standard was developed to prevent the proliferation of data transfer formats that occurred with microprocessor development systems. The focus of the standard is on field programmable devices and their support tools. It is not intended for other types of semi-custom logic devices, or other types of fabrication or testing equipment.

The format was developed by the Solid State Products Engineering Council, and is published by the Electronics Industries Association.

The 18-page JEDEC document includes sections on special notations and definitions, transmission protocol, data fields, comment and definition fields, device programming fields, device testing fields, programmer/tester options, and examples.

To obtain a copy of the JEDEC file format, contact:

Electronic Industries Association
Engineering Department
2001 Eye Street, N.W.
Washington, D.C. 20006

The fee as of 1986 was US$11.00.

B XELTEK Customer Support

B.1 Introduction

This appendix contains Customer Support information, including the Software Registration Card, the XELTEK Customer Support department for technical assistance, information on the License Agreement, the limited one-year warranty, the 30-day money-back warranty, and the one-year software service with fee.

B.2 Registration

Complete the card and return it to XELTEK, to become eligible for the following:

. Customer support, warranty service, and technical assistance.

. Notification of any new products or special offer.

Registration is particularly important if the program was purchased from a dealer or distributor, or through your purchasing department. Why not take a moment right now to complete the card?

B.3 Calling Customer Support

XELTEK software has been designed to require a minimum of technical support. The programs come with a comprehensive, indexed Reference Manual. If you cannot find the answer in the manual, you can turn to your dealer, distributor, or to XELTEK.

XELTEK provides free telephone technical assistance during normal business hours (8:00 am to 5:00 pm, Pacific Time). **Have your invoice or purchase number ready when calling, as we cannot answer your questions without it.** To help us serve you better, please review the following check list prior to placing your call:

(408)524-1929

Pre-Call Check List

1. If you're having difficulty understanding the program, have you studied the manual? The Reference Section of the manual explains the program and design language details in even greater depth.
2. If there seems to be a problem with the software, can you reproduce it?
3. If the program has displayed an error message, please write down the message.
4. When calling in for technical assistance, have your invoice number ready.
5. Be familiar with the hardware configuration you're using. We may need to know the brand/model of your computer, the total amount of memory available when starting up the software, your graphics adapter, your printer brand/model, and version of the disk operating system (DOS).
6. If possible, **please call our Customer Support department while you are at your computer.** Be prepared to repeat the sequence of steps leading up to the problem.

B.4 License Agreement

The purchaser is granted a non-exclusive license to use the program on a single computer, subject to the terms and restrictions set forth in this Agreement. This license is not a sale. The copyright of the programs and Reference Manual remain the property of XELTEK.

You may:

- Copy the programs for back-up purposes ONLY in support of their use on a single computer.
- Transfer the program from computer to computer, provided that it is not operated on more than one computer at a time.
- Transfer the program and license to another party if the other party agrees to accept the terms and conditions of this Agreement. If you transfer the program, you must at the same time either transfer all copies to the same party or destroy any copies not transferred.

You may not:

- Use this product in a computer system or network which allows the program to be operated by more than one user at a time.
- Use, modify, copy or transfer the Reference Manual or other documentation or any copy thereof except as expressly provided in this Agreement.
- Reverse engineer, decompile or disassemble any program module or security device.

This license is effective until terminated. You may terminate it by destroying or returning the program, reference manual, and all copies thereof. This license will terminate if you fail to comply with any term or condition of this Agreement.

B.5 Limited Warranty

XELTEK warrants that its products will be free from defects in workmanship and materials, and shall conform to specifications current at the time of shipment, for a period of one year from date of shipment. During the one year period, any defective software or hardware products will be repaired or replaced, at Seller's option, on a return to factory basis. (For details, see **Return/Replacement Procedure** below.)

This warranty applies only to products properly installed and operated within specified environmental conditions.

XELTEK's responsibility under this warranty does not apply to (a) any products which have been repaired, worked upon or altered by any person not duly authorized in writing by XELTEK; or (b) any product which has been subject to misuse, negligence or accident or whose serial number has been altered, defaced or removed; or (c) any faults induced into the program by physical damage to the diskette or to the corruption of the program by electronic or electrical interference.

All efforts have been made to verify the proper operation of the software and the accuracy of the Reference Manual. XELTEK does not warrant that the operation of the program will be uninterrupted or totally error free.

New features and enhancements to the existing programs may be added, and verified program faults or necessary amendments will be rectified through the issue of periodic software revisions, notice of which will be mailed to **buyers of software updates service only**. Failure to return the completed Software Registration Card with correct mailing address will exclude the purchaser from automatically receiving notification of revisions to the program. Software uptades and revisions will be subject to a fee.

XELTEK does not accept liability for any damages, including loss of profit or savings, other incidental or consequential damages arising out of the use, or the misuse or inability to use the program.

THIS WARRANTY TO REPAIR OR REPLACE PRODUCTS NOT CONFORMING THEREOF, IS SPECIFICALLY IN LIEU OF ALL OTHER WARRANTIES EXPRESSED OR IMPLIED INCLUDING, WITHOUT LIMITATION, THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE. NO AGENT OR REPRESENTATIVE OF XELTEK HAS ANY AUTHORITY TO BIND XELTEK TO ANY AFFIRMATION, REPRESENTATION, OR WARRANTY CONCERNING PRODUCTS MADE BY XELTEK; AND, XELTEK WILL HAVE NO LIABILITY WHATSOEVER FOR ANY DAMAGE, LOSS, COST, OR EXPENSE (WHETHER DIRECT, SPECIAL OR CONSEQUENTIAL) SUFFERED OR INCURRED BY BUYER IF PRODUCTS FAIL TO CONFORM WITH XELTEK'S WARRANTY HEREIN.

B.6 30-Day Money-Back Guarantee

XELTEK product is sold with a 30-day money-back guarantee. If you are not completely satisfied with the package, return it to the place of original purchase within 30 days of receipt for a complete refund of your purchase price. See **Return/Replacement Procedure** below for instructions on how to return the software.

B.7 Return/Replacement Procedure

Under its Warranty, XELTEK agrees to replace defective diskettes, documentation or units for up to one year from the time of delivery. The replacement materials will be supplied at no charge, providing they are verified to be defective through no fault of the user. A reasonable cost for materials and return freight will be charged to replace materials defective through the fault of the user, for which the one year warranty period has expired.

Under its Money-Back Guarantee, XELTEK agrees to accept for return, and refund its software and hardware products for up to thirty (30) days from the time of delivery.

The Buyer must obtain prior approval and a Return Material Authorization (RMA) number from XELTEK before returning any product to XELTEK. The entire package, including software, hardware, documentation, discount coupons, and any other items supplied, must be returned intact. This guarantee will not be honored for packages that are not returned complete and intact.

When you mail a unit for repair or refund, please be sure to have the following information included with the product: the RMA number, the invoice and purchase order of copy, and the description of the problem. This will expedite the process of repairing and refunding. Also, have your RMA number visibly written outside of the box.

Transportation and insurance charges to XELTEK's facility shall be paid by the buyer. Transportation and insurance charges from XELTEK's facility to buyer's facility of equipment repaired or replaced will be paid by XELTEK. XELTEK will ship the above repaired items via a method similar to the one that Buyer used to return the defective equipment.

C HEX FILE FORMAT

C.1. INTEL Formats

A. INTEL 1 : 16 bit address field format, for files 64K bytes in length or less.

DATA RECORD

Byte 1	Header=colon(:)
2..3	The number of data bytes in hex notation.
4..5	High byte of the record load address
6..7	Low byte of the record load address
8..9	Record type, must be "00"
10..x	The data bytes in hex notation: x=(number of data bytes-1)*2+11.
x+1..x+2	Checksum in hex notation.
x+3..x+4	Carriage return,line feed

END RECORD

Byte 1	Header = colon(:)
2..3	The byte count, must be "00".
4..7	Transfer-address (usually "0000").
8..9	Record type, must be "01".
10..11	Checksum in hex notation.
12..13	Carriage return, line feed.

The checksum are the two complements of the 8-bit sum with out the carry of the byte count,and the transfer address. It will be "FF" if above format is followed.

B. INTEL 2 : MCS 86 format, using a 20-bit address for files larger than 64K bytes.

DATA RECORD

Byte 1	Header = colon(:)
2..3	The byte
4..5	High byte of the record load address.
6..7	Low byte of the record load address
8..9	Record Type, must be "00".
10..x	The data bytes in hex notation: x=(number of data bytes-1)*2+11.
x+1..x+2	Checksum in hex notation
x+3..x+4	Carriage return, line feed

EXTENDED ADDRESS RECORD

Byte 1	Header = colon(:)
2..3	The byte count, must be "02".
4..7	Load address, must be "0000".
8..9	Record type, must be "02".
10..11	High byte of the offset address
12..13	Low byte of the offset address
14..15	Checksum in hex notation.
16..17	Carriage return, line feed

The checksum are the two's complement of the 8-bit sum without carry of the byte count,offset address, and the record type.

START ADDRESS RECORD

Byte 1	Header = colon(:)
2..3	The byte count, must be "04".
4..7	Load address, must be "0000".
8..9	Record type, must be "03".
10..13	8086 CS value.
14..17	8086 IP value.
18..19	Checksum in hex notation.
20.21	Carriage return, line feed.

C.2. Motorola Formats

A. Mortorola S1 : this is the 16-bit address field format, for files 64K bytes in length of less.

DATA RECORD

Byte 1..2	Header = "S1".
3..4	Byte count = number of data bytes + 3, in hex notation.
5..6	High byte of the record-load address.
7..8	Low byte of the record-load address.
9..x	The data bytes in hex notation: x =(number of data bytes1)*2+10.
x+1..x+2	Checksum in hex notattion.
x+3..x+4	Carriage return, line feed

END RECORD

Byte 1..2	Header = "S9".
3..4	Carriage return, line feed.

B. Mortorola S2: This is the 24 extended address field format, for files greater than 64 Kbytes in length.

DATA RECORD

Byte 1..2	Header = "S2" (24 bit address field)
3..4	Byte count = the number of data bytes+4, in hex notation.
5..6	High byte of the 24 bit record-load address.
7..8	Middle byte of the 24 bit record-load address.
9..10	Low byte of the 24 bit record load-address.
11..x	The data bytes in hex notation: x =(number of data bytes-1)*2+12.
x+1..x+2	Checksum in hex notation.
x+3..x+4	Carriage return, line feed.

The byte count in data records includes the load-address and the checksum.

END RECORD

Byte 1..2	Header = "S9".
3..4	Carriage return, line feed.

The checksum for any record is the one's complement of the 8-bit sum, without carry of the byte count, the load address, and the data bytes.

COMMENT OR SIGN-ON RECORD

Byte 1	Header = "S0".
2..x+1	Message up to x characters.
x+2..x+3	Carriage return, line feed.

C. Mortorola S3 : This is the 32 extended address field format, for files greater than 64 Kbytes in length.

DATA RECORD

Byte 1..2	Header = "S3". (32-bit address field).
3..4	Byte count = the number of data bytes+5, in hex notation.
5..6	High byte of the 32-bit record load address.
7..10	Middle byte of the 32-bit record load address.
11..12	Low byte of the 32-bit record load address.
13..x	The data bytes in hex notation: x=(number of data bytes-1)*2+14.
x+1..x+2	Checksum in hex notation
x+3..x+4	Carriage return, line feed.

The byte count in data records includes the load-address and the checksum.

END RECORD

Byte 1..2	Header = "S9".
3..4	Carriage return, line feed.

The checksum for any record is the one's complement of the 8-bit sum, without carry of the byte count, the load address, and the data bytes.

COMMENT OR SIGN-ON RECORD

Byte 1	Header = "S0".
2..x+1	Message up to x characters.
x+2..x+3	Carriage Return, line feed.

C.3. Tektronix Format

A. Tektronix Format

DATA RECORD

Byte 1	Header = slash (/).
2..3	High byte of the record-load address.
4..5	Low byte of the record-load address.
6..7	The number of data bytes in hex notation.
8..9	The first checksum in hex notation; it is the sum of the load-address and the number of data bytes.
10..x	The data bytes in hex notation: x=(number of data bytes-1)*2+11.
x+1..x+2	The second checksum in hex notation; it is the 8-bit sum modulo 256 of the data bytes.
x+3	Carriage Return.

END RECORD

Byte 1	Header = slash (/).
2..5	Transfer address.
6..7	The byte count, must be "00".
8..9	Checksum in hex notation; it is the sum of the transfer-address and the byte count.
10	Carriage Return

ABORT BLOCK

Byte 1	Header = slash (/).
2	Slash (/).
3..x+2	Message up to 69 characters:

	x = number of characters.
x+3	Carriage Return.

All checksums are used in the Textronix format are computed as the 8-bit sum module 256 of the 4-bit hex digits that make up the bytes. That is, the checksum is the sum of the upper and lower nibbles of each byte.

B. Extended Tektronix Format

DATA RECORD

Byte 1	Header = "%"
2..3	Block length. Number of characters in the record, minus the "%"
4	Record type.Data record = "6"
5..6	Sumcheck. A 2-digit hex sum module 256 of all the values in the record except the "%" and the sumcheck itself
7..X	Load address. Determines where the object code will be located. A variable length number that may contain upto 17 characters. The first number determines the address length, with a zero 9 signifying a length of 16.
X+1	Object code, 2 characters per byte

END RECORD

Byte1	Header = "%"
2..3	Block length. Number of characters in the record minus the "%"
4	Record type. End record = "8"
5..6	Sumcheck. A 2-digit hex sum modele 256 of all the values in the record except the "%" and the sumcheck itself.

D SELF MAINTENANCE KIT

COMMUNICATION & TRANSISTOR TEST FOR UNIPRO

Optional Self Maintenance Kit for UNIPRO is available for users who want to maintain UNIPRO by themselves. It includes self testing software and a testor which you can plug into the 40-pin software of UNIPRO.

Note: This test only works with the default-port address, the fifth jumper, in the interface card.

D.1 Testing

1. Set up your Unipro.

2. Execute the testing software and select option 1. This will check the communication of the PCB.

3. Next, mount the "Unitestor", which is the array of 330 ohm resistors, and select option number two of the testing software. If you don't have the "Unitestor", anyone can make one. The lowest left pin will be connected with all other pins in the 40 pin dip socket through one 330 ohm resistor in each connection.

D.2 Reparing

For example, if the results read "TRQ 2 ERROR",

1. Flip the Unipro cover open

2. You will see rows and columns of the transistor (looks like semi-circle-cylinders standing).

3. Locate the Q2 and replace it (There are only two kinds of transistors: 2N 4401 or 4403).

4. Test Unipro again following procedure A.

These procedures will solve 65% of problems on UNIPRO. If you have further questions, please contact the division of technical support at (408) 745-7974.