SUPERPRO

Universal Programmer

User's Reference Guide Manual

XELTEK

Copyrights

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

ITALIA INSTRUMENTS srl
STRUMENTAZIONE
ASSISTENZA TECNICA
10136 TORINO   C.so IV Novembre 16
☎ (011) 31 80 712 r.a.

541911

SUPERPRO INTERFACE SOFTWARE

Version 1.0 includes programs for programming PAL's, GAL's, PROM's, and Single microcontrollers. Also this can test TTL & CMOS logic chips and RAM's.

The new features of these programs include support for JEDEC files created by many compilers and an entirely new user interface. The programs feature a pull-down menu interface which allows operation of functions from the menus, by typing in the key letter of each command. Typing the first character of the menu is convenient to select the menu of interest instead of highlighting and pressing the return key. (Thus it was inevitable to use the non overlapping middle character when the two menus have the same first characters. The on-screen prompt lines assist your selection of options.

Attached to these release notes are newly illustrated chapters for the SUPERPRO reference manual which fully document the operation of most of the programmable chips. This manual is consisted of three parts: Book I, Book II, and appendixes. Book I will explain the general informations such as system requirements, getting started, and etc. In Book II the detailed information about the each menu will be given. In the appendixes data format, customer support, customer and the method of self-maintenance will be described. So I suggest quick readers must read Book I, but refer to Book II as necessary.

In this manual, only regular version of SUPERPRO will be explained. The algorithm generator will approach the users in a different package. The algorithm generator has the power to update new devices.

Contents

BOOK II

BOOK I

This part will introduce the SUPERPRO, explain the requirements of system, and let you get started. In the Chapter 4, there are quick guide line about programming various kinds of chips. For quick readers, it is suggested to use the Chapter 4 as an itinerary line and Book II as a detailed map for programming.

1 INTRODUCTION

Congratulations on your purchase of the SUPERPRO universal programmer! To protect your investment in the SUPERPRO 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. What Is SUPERPRO?

The SUPERPRO is an affordable, reliable, and extremely fast universal device programmer. The SUPERPRO is designed to operate with the IBM-PC/XT/AT/386 series of computers and their compatibles. The SUPERPRO'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 SUPERPRO'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.

- Four layer PCB design for outstanding apperance and noise reduction.

The SUPERPRO interface software includes these features:

- The utmost capability of updating chips of SUPERPRO advanced version.

- 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 will be loaded, and are automatically converted to BIN format.

This document explains how to install and use your SUPERPRO 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.

2. Manual Organization

The SUPERPRO Reference Manual consists of three main parts: Book I, Book II, and Appendixes.

Book I : Chapters 1 - 4
An introduction to the SUPERPRO, including features, system requirements, set-up of the hardware and software, and brief instructions for programming various kinds of chips.

Book II : Chapters 1 - 7
Detailed operating instructions and menus for each of the software routines are explained.

Appendixes : A - D
Information on the JEDEC file format, HEX file format, XELTEK customer support, and self maintenance.

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:

    sp

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 selection inside a quotation mark. The sub levels will be indicated by an arrow. For example, referring to the Macro delay time inside the menu, Environment which is again the submenu of option will be as follows. Select "Option -> Environment -> Macro delay time". This tells the users to go to the menu of option and select the submenu, environment and finally select Macro delay time.

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 REQUIREMENTS

1. Hardware and Software

The minimum hardware requirements for using the SUPERPRO 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 SUPERPRO software operates in text mode and supports all graphics devices, color, or monochrome.

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 SUPERPRO 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.

3. SUPERPRO Package

Your SUPERPRO package contains the following:

- The SUPERPRO programming module.

- The half-slot parallel interface PCB card.

- A parallel interface connecting cable.

- The SUPERPRO Reference Manual.

- The SUPERPRO 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.

4. Files on the Release Diskette(s)

There will be one system diskette and one or more algorithms diskettes.

System diskette
        A:\install. bat
        A:\sp.exe
        A:\sphelp.hlp
        A:\lib*.lib
        A:\lib\ttl.lib

One of more algorithm diskette(s) contain(s) files with "lef" suffix.

3 GETTING STARTED

1. Checklist

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

- Make backup copies of the SUPERPRO software diskettes.

- Install the SUPERPRO software in a directory on the hard disk.

- Install the SUPERPRO 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 SUPERPRO software routines to load, edit, and download files to the programmable device.

2. Creating Backup Copies

We strongly recommend that you make backup copies of the SUPERPRO 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 SUPERPRO 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 SUPERPRO 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 SUPERPRO diskettes in a cool, dry place, away from magnetic fields and direct sunlight.

3. Installing Hardware

The SUPERPRO 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 SUPERPRO 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:

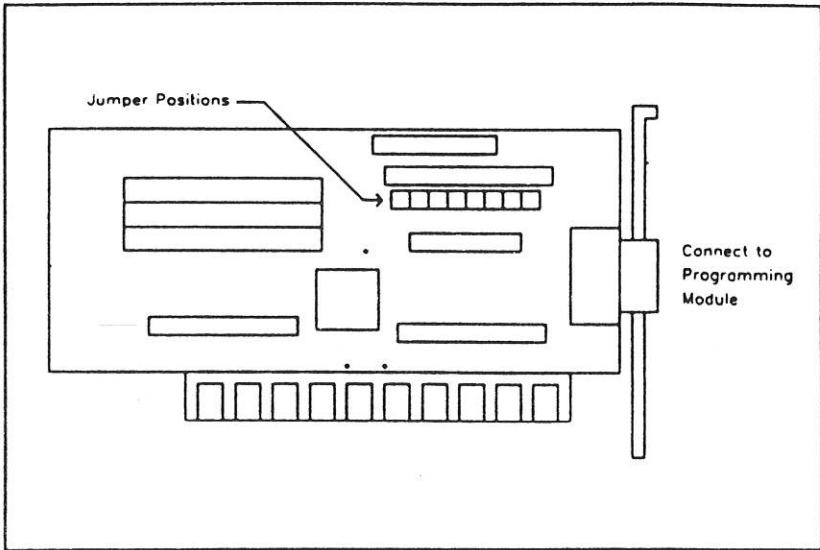| Jumper Position	I/O Address (Hex)
#1	200-21F
#2	220-23F
#3	240-25F
#4	260-27F
#5	280-29F (Default)
#6	2A0-2BF
#7	2C0-2Df
#8	2E0-2FF

Figure 3.1 SUPERPRO parallel interface card

If you changed the jumper position on the interface card, you may have to change the port address information within SUPERPRO software routine. When you first run the 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. SUPERPRO routine includes a command to reset the port address, which should resolve the problem. To change the port address, ignoring the communication error message press <return> and select "Option -> Interface port". Then select the desired address.

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!

4. Installing 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>

Insert the system diskette and type **install**. After the creation of BIN, LIB, and ALGO, system disk will be copied in the directory BIN, and Library will be copied under LIB. Then it will ask for the next diskette to be inserted until there is no diskette left.

4.1. Setting DOS Path

By setting the DOS PATH variable (described in detail in the DOS Manual), you can run the SUPERPRO software from any directory. You do not have to be in the \SUPERPRO directory to execute the program.

To set the DOS path, assign the pathname \SUPERPRO 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:\SUPERPRO

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.

5. Running Software

In general, you should always turn ON the SUPERPRO programming module *before* starting any of the software routines, to avoid the communication error. The regular version will not include **ag** and **ig** software. The advanced version will be sold separately for the users who want to update devices by themselves.

To program all kinds of devices in the advanced version of SUPERPRO, type :

sp (abbreviation of SUPERPRO)

To update devices in the advanced version of SUPERPRO, type :

ag (abbreviation of Algorithm Generator)

To manage the database in the advanced version of SUPERPRO, type :

lg (abbreviation of Library Generator)

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 SUPERPRO 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 SUPERPRO software routine to the setting of the interface card.

4 PROGRAMMING

This chapter will be devoted in explaining the menu selection for the various kinds of devices. Thus, a general way of selecting menus for programming will be discussed without much details. For the details in question please refer to any relevant Chapter of Book II which explains the menus in details.

1. TTL & CMOS (RAM)

The SUPERPRO can test logic IC and RAM memory devices and vector testing. A menu-driven software program provides an easy interface for loading files, editing test patterns, and downloading test files to the device. TTL.LIB Contains the TTL and CMOS test patterns. The following checklist shows the steps required to test the Logic IC's and RAM's. To test logic ICs and RAM memories, be certain that the SUPERPRO hardware is properly installed, and that the programmer is turned on.

A. Set up the interface card into the EMS (Expansion Memory Slot) and connect the programmer to the computer by the cable provided. Turn on the computer and invoke the software by typing "SP". Mount the device of interest on the 40 pin DIP socket of SUPERPRO. For details on installation, please refer to Chapter 3.

B. From the screen, select the "Test->TTL & CMOS Test" and press the return key. This menu can also be selected by pressing "Ctrl-F4."

C. Enter the device name in the box provided. If the device is not found in the library of TTL.LIB, an error message will be displayed.

D. Upon pressing <return>, test result will be displayed.

For RAM testing, follow the procedure exactly, but in B select "Test -
>Memory test."

2. PROM (Bipolar, CMOS, NMOS..) and Single Microcontroller

A. Set up the interface card into the EMS (Expansion Memory Slot) and connect the programmer to the computer by the cable provided. Turn on the computer and invoke the software by typing "SP". Mount the device of interest. For details on installation, please refer to Chapter 3.

B. From the main screen, go to "Manufacturer->Default device format." and select "ROM" in the submenu.

C. Go to "Manufacturer->Manufacturer" and select by typing the name of manufacturer or <return>, to display all the manufacturers. When all the manufacturers are displayed, highlight the right manufacturer and press the return key to select.

D. Go to "Manufacturer->Type" and select the part name of a chip by either highlighting or typing.

E. Go to "File->Load" and load a HEX file.

F. Go to "Buffer->Edit" and check the data file.

G. Go to "Device->Function select", a big subscreen is displayed. Highlight a function desired and execute. To move between blocks in the screen, press Alt and the capitalized character of the head title of the block.

3. PAL & GAL

A. Set up the interface card into the EMS (Expansion Memory Slot) and connect the programmer to the computer by the cable provided. Turn on the computer and invoke the software by

typing "SP". Mount the device of interest. For the details of installing, please refer to Ch 3.

B. From the main screen, go to "Manufacturer->Default device format." and select "PLD" in the submenu.

C. Go to "Manufacturer->Manufacturer" and select by typing the name of manufacturer or just return to display all the manufacturer. When all the manufacturer displays highlight the manufacturer and press return key to select.

D. Go to "Manufacturer->Type" and select the name of a chip by either highlighting or typing.

E. Go to "File->Load" and load a JEDEC fuse map file.

F. Go to "Buffer->Edit" and check the data file.

G. Go to "Device->Function select", a big subscreen is displayed. Highlight a function desired and execute. To move between blocks in the screen press Alt and the capitalized character of the head title of the block.

H. For Vector Testing, go to "Test->Vector test", and perform vector testing.

BOOK II

This describes all the menus and their submenus. This is consisted of seven Chapters which are the root level menus seen in the main screen. The left most menu "File" is Chapter one and the right most menu "Option" is Chapter seven.

1 FILE

This menu deals with file management and system interfacing.

Figure 1.1 File Menu

1. Load

Primarily, files can be categorized into two kinds, one is JEDEC fuse map file format and the other is HEX file format. This menu will deal with both. In both cases wildcard characters such as * and ? will be accepted. If you use wildcards the range of files specified will be displayed, then highlight the portion and press return key for loading.

The reserved key "F3" will bring out this menu from any menu and at any time.

1.1. Loading a JEDEC file

This loads a JEDEC file from disk to the buffer if "Manufacturer->Default Device Format -> Fuse" is selected. A box for the name of the JEDEC file to be loaded will be displayed. Error message will be displayed for non JEDEC fuse map files.
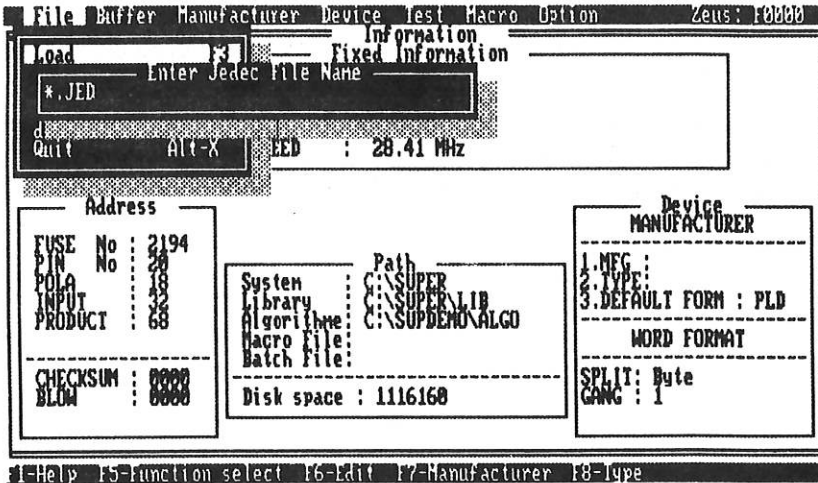

Figure 1.2 Menu for JEDEC file

1.2. Loading a HEX file

This loads a HEX file from disk to the buffer. The format selected has to be the same format as the HEX file. If not, loading of the HEX file will not be successful. The following formats are supported by the programmer:

Sub Menus ;

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

The reserved key "F3" will bring out this menu from any menu and at any time.

A. Binary File : Multiple Loading

Loads the binary data into the buffer from the disk. Multiple loading is possible if "Config->Environment->Multiple Load" menu is set "on". In this mode submenus will be displayed, and by specifying the start addresses it is possible to load many files in the buffer. Multiple loading is possible with HEX files since Superpro can save any file into binary format after loading HEX files into the software.

    Sub menus ;

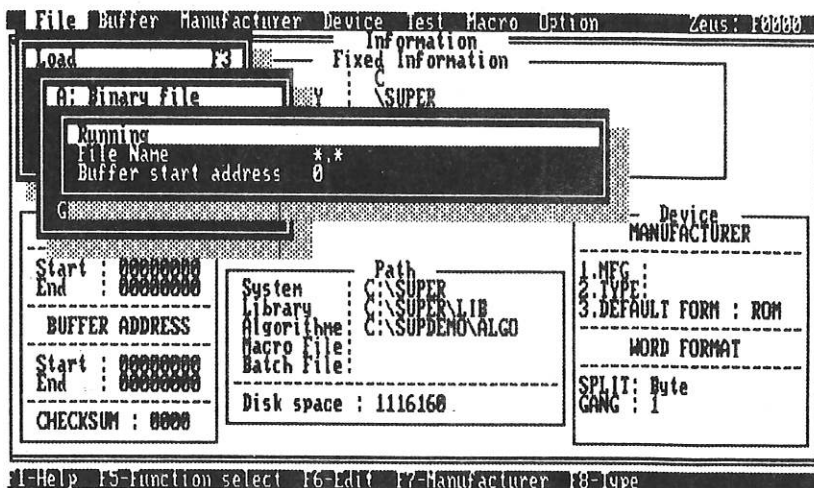|	
Running	: Initiates the execution
File name	: Takes a file name to save
Start address	: Start address of the buffer

Figure 1.3 Multiple Loading set "ON"

If "Config->Environment->Multiple Load" · menu is set "off",
Submenus will not be displayed. Only a box for a filename will be
displayed as in the figure below.

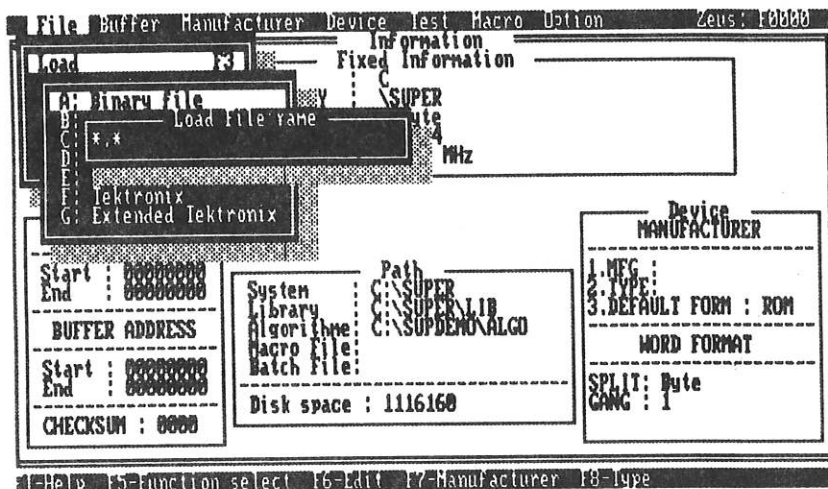Figure 1.4 Multiple Loading Set "OFF"

B. Intel (Extended) HEX file load

C. Motorola S1 HEX file load

D. Motorola S2 HEX file load

E. Motorola S3 HEX file load

F. Tektronix HEX file load

G. Tektronix Extended HEX file load

2. Save

Saves the file in the buffer which has been created, loaded from disk, modified, or read from a chip onto a specified disk. Being similar to the menu "File ->Load", this is used for two kinds of savings: JEDEC fuse maps and HEX files.

The reserved key "F2" will bring out this menu from any menu and at any time.

2.1. Saving JEDEC fuse map file

If PLD is chosen in the menu of "Manufacturer->Default Device Format" a box for the name of the JEDEC file to be saved will show up. If there is a file existing overwrite protection message will be displayed.

Figure 1.5 JEDEC file saving

2.2. Saving HEX file

If ROM is chosen a submenu which has different HEX file formats will
show up as below.

Sub menus ;

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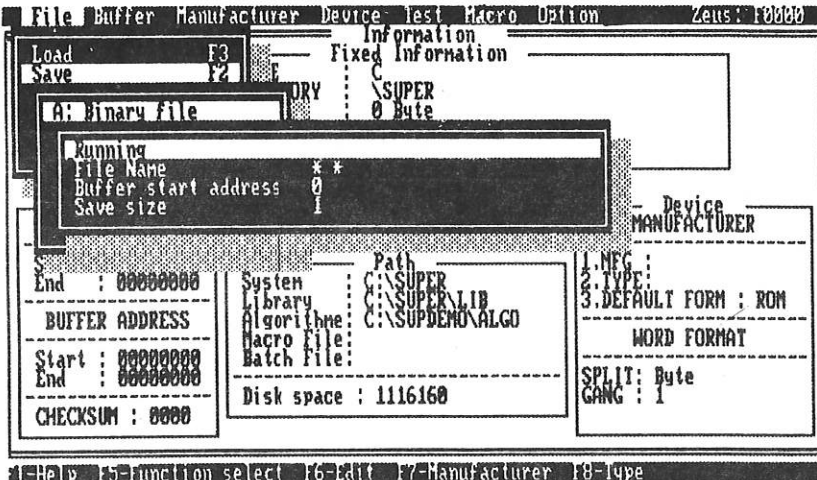F   : Tektronix
G   : Extended Tektroni

Figure 1.6 HEX file saving

All seven menus have the same submenus and the usage is identical.
They are explained below.

Sub menus ;

Running   : Initiates the execution
File name   : File name to be saved
Start address   : Buffer address to start

Buffer size   : No. of bytes to be saved

CAUTION: When specifying the buffer address, the number of bytes to be saved is one byte greater than the difference between the last address and the first address. For example, to save the content from 0th byte to 7th byte, the buffer size should be 8.

3. Directory

If a directory is specified, the files in that directory will be displayed.

4. User Screen

It displays the screen just before you enter the software. Typing any key will return to the software.

The reserved key "Alt-F5" will bring out this menu from any menu and at any time.

5. DOS/SHELL

This allows the user to switch easily between the SUPERPRO software and DOS. To switch to SUPERPRO software from DOS, type "exit".

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

6. Quit

Returns back to the operation system. The reserved key "Alt-X" will bring out this menu from any menu and at any time closing the program and returning the cursor on DOS.

The reserved key "Alt-X" will bring out this menu from any menu and at any time.

2 BUFFER

The buffer contains the data from the JEDEC file, HEX file or a master chip.

1. Edit

Views and edits the data loaded in the buffer.

Sub menus ;

Edit
Checksum
Fill
Vector Table
External
Special Area

The reserved key "F6" will bring out this menu from any menu and at any time.

1.1. HEX file buffer

If "Manufacturer->Default Device Format->ROM" is selected, The buffer for HEX data will display.

Arrow	:	Left,right,up or down
Ctrl-N	:	Moves the cursor to the address you specified
Ctrl-E	:	Scroll up one line
Ctrl-Z	:	Scroll down one line
PgUp	:	Page down
PgDn	:	Page up
Home	:	To the beginning of a line
End	:	To the end of a line
Ctrl-Home	:	To the beginning of a screen
Ctrl-End	:	To the end of a screen
Ctrl-PgUp	:	To the beginning of the buffer

Ctrl-PgDn	:	To the end of the buffer
TAB	:	Switches back and forth between ASCII and HEX fields
ESC	:	Exits out of the buffer

1.2. JEDEC fuse map buffer

If "Manufacturer->Default Device Format->PLD" is selected, the fuse map will be displayed. Editor keys are same as those for the HEX editor buffer with some differences listed below.

| 1 | Logic high |
| 0 | Logic low |

2. Check Sum

If "Manufacturer->Default Device Format->ROM" is selected, the four least significant HEX digits from the sum of all the data will show up to accommodate quick comparisons.

The reserved key "Alt-F9" will bring out this menu from any menu and at any time.

Submenus ;

Running	: Initiates the execution
Buffer start address	: Start address
Buffer end address	: End address

If "Manufacturer->Default Device Format->PLD" is selected, a box for the check sum and the blow count will be displayed the lower left corner.

The reserved key "Alt-F9" will bring out this menu from any menu and at any time.

Figure 2.1 Checksum

3. Fill

When "Manufacturer->Default Device Format->ROM" is selected, this fills the desired value in HEX on the specified portion of the buffer.

The reserved key "Alt-F10" will bring out this menu from any menu and at any time.

Sub menus ;

Running on	: Initiates the execution
Buffer Start Address	: Filling starts
Buffer End Address	: Filling ends
Fill Data	: Data to be filled

If "Manufacturer->Default Device Format->PLD" is selected, the content of the buffer will be filled with blank characters.
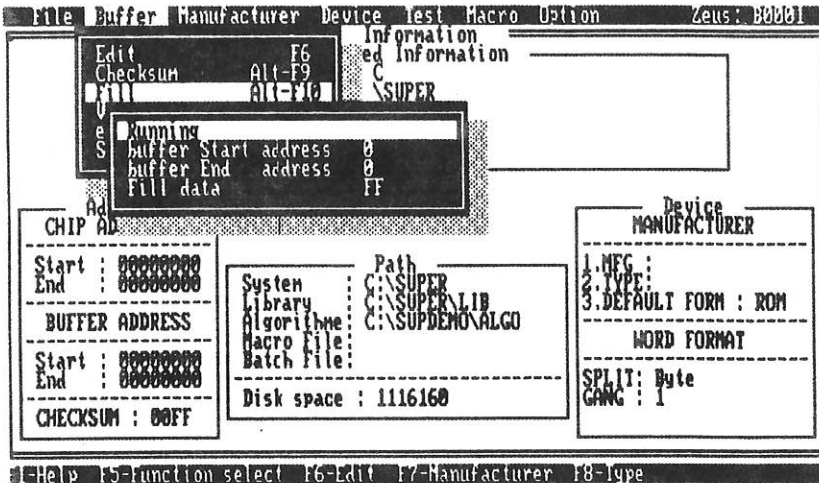
Figure 2.2 Fill menu

4. Vector Table

This menu only applies when "Manufacturer->Default Device Format->PLD" is selected, This loads the test vector file from JEDEC files into the buffer automatically if there is a vector test file in the JEDEC file. Press T to perform Vector testing.

The reserved key "Alt-F8" will bring out this menu from any menu and at any time.

Figure 2.3 Vector Table

Editor Keys are as followed.

Z	: High impedance state
X	: Don't care state
N	: Vcc and Ground, and not tested output pins
H	: Output Logic High (VOH)
L	: Output Logic Low (VOL)
C	: Clock pin
1	: Input Logic High (VIH)
0	: Input Logic Low (VIL)
Arrow	: Left,right,up or down
Ctrl-N	: Moves the cursor to the required address

Ctrl-E	: Scroll up one line
Ctrl-Z	: Scroll down one line
PgUp	: Page down
PgDn	: Page up
Home	: Go to the beginning of a line
End	: Go to the end of a line
Ctrl-Home	: Go to the beginning of a screen
Ctrl-End	: Go to the end of a screen
Ctrl-PgUp	: Go to the beginning of the buffer
Ctrl-PgDn	: Go to the end of the buffer
TAB	: Toggles between ASCII and HEX fields
ESC	: Exits buffer
T	: Activates vector test
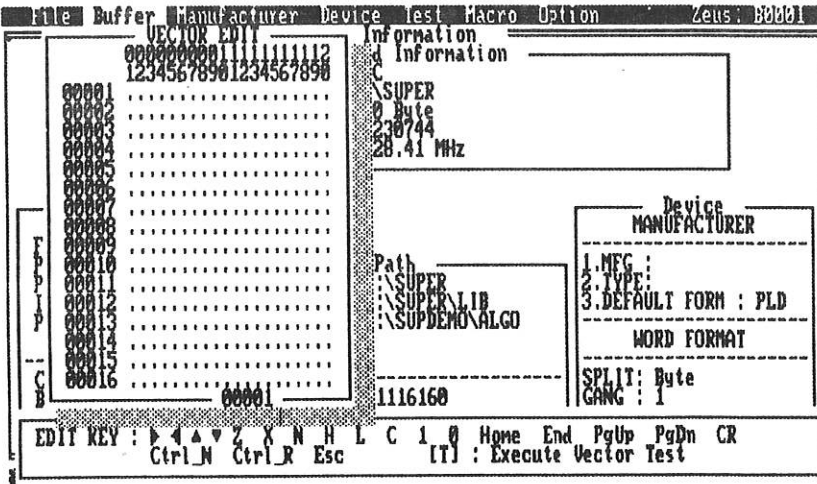
Figure 2.4 Vector Edit Buffer

5. External

This applies to only GAL and Single chips. For GAL this is used for
MES and UES read. For microcontrollers, it is used for encryption
tables.

*The reserved key "Alt-F7" will bring out this menu from any menu and at
any time.*
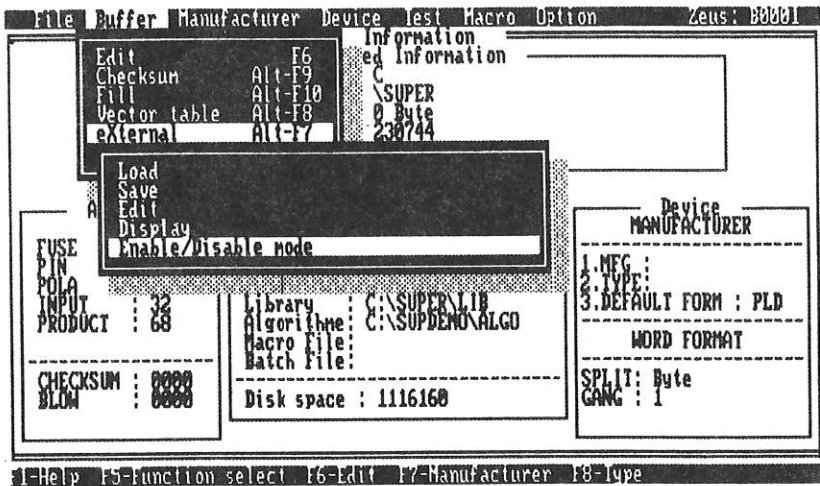
Figure 2.5 External

5.1 MES Edit

MES (Manufacturer Electronic Signature) is the device information
defined by the manufacturer. The MES contains the following
information:

Cycle counter   No. of programmings done to the device.

Algorithm revision   Provides the necessary parameters for
                       programming

Master Bit	1 means master, and 0 means non master. When programming or erasing a master device it displays a warning message.
Device code	8-bit code assigned to a device. (For example, RAL18H4 [code = 27])
Manufacturer's ID	ID for an individual manufacturer.

5.2 Encryption Table (for single chip)

This menu deals with encryption table management.

Figure 2.6 Encryption Table

A. Load : Loads a file into the encryption table.

B. Save : Saves the content of the encryption table in disk.

C. Edit : Views and edit the table. Encryption table can be edited by arrow key, home, and End key. The size of the default table is 32 bytes. Only HEX values will be accepted.

D. Encryption table on/off : If this mode is on, the data read will be Exclusive-NORed with the data in the encryption table. The outcome will be displayed in the buffer.

E.   Encryption table size : Normally, the size of the encryption table is 32 byte, but it differs a little depending on the device. Only HEX values will be accepted.

6. UES Edit (Special Area)

UES (User Electronics Signature) EDIT is used to edit the information in the UES information area. UES editing is also possible in the fuse map buffer. The UES is 64 bit long and can contain information like socket location, program pattern revision, design revision, and etc.

3 MANUFACTURER

This menu is used to choose the IC manufacturer and the device type. Also, this menu will set up the environment for the chip type selected.

Sub menus ;

Manufacturer
Type
Device format

The reserved key "F7" will bring out this menu from any menu and at any time.

1. Manufacturer

This defines the IC manufacturer. To see the list of all the companies, type *, highlight the entry and press the carriage return. Or, specify the name of the company and return.

The reserved key "F7" will bring out this menu from any menu and at any time.

2. Type

After the selection of the manufacturer, the device type is selected. If users return, not typing any characters, a group of devices will be displayed. If PLD is selected in "Manufacturer->Device Default Format", all the PLD's will show from the manufacturer selected, and if ROM then ROM's will show up. However if ALL is chosen then all the devices will be displayed.

The reserved key "F8" will bring out this menu from any menu and at any time.

3. Default Device Format

This menu sets the mode for PLD, ROM and ALL. Depending on the
mode, different submenus will be displayed.

*The reserved key "F9" will bring out this menu from any menu and at any
time.*

Figure 3.1 Default Device Format

4 DEVICE

This menu deals with the management or manipulation of the device. There are three submenus, Function Select, Word Format, and Gang Select.

1. Function Select

This menu will display a screen for all the pertinent information for programmings. There are five fields such as functions, message, address, repeat, and environment. Each field will be explained below.

Figure 4.1 Function Select

A.   Function field

Users can perform all the possible functions for implementing devices such as programming, reading, verifying, blank-check, data comparison, automatic programming, encryption and security programming.

B.   Repeat field

Repeat, Delay, and Count will be recorded.

C.   Message field

Error message or programming result will be displayed.

D.   Address field

The informations about the chip and buffer address, current address, and the gang number selected will be displayed.

E.   Environment field

The information such as check sum, the number of gang selected, the way of byte splitting for programming, the name of a manufacturer and the part name of the device selected will be displayed.

Note: There are only three fields which are necessary to be accessed. They are fUnctions, Repeat, and addrEss. To go into the each section users need to press Alt and the capitalized character of the section title wanted. For example to change the start address of the buffer for programming users have to go into the address field and type over the start address field. To go into the address field users must press Alt-E.

A. fUntions

1. Program

This downloads the data in the buffer onto the chip. The size of the buffer is 64k bytes ranging from 0 ~ 0XFFFF. "Verify" function will be performed after programming. If there is an error the error message with the address where the error occurred will be displayed. Any other result will be displayed in the message section.
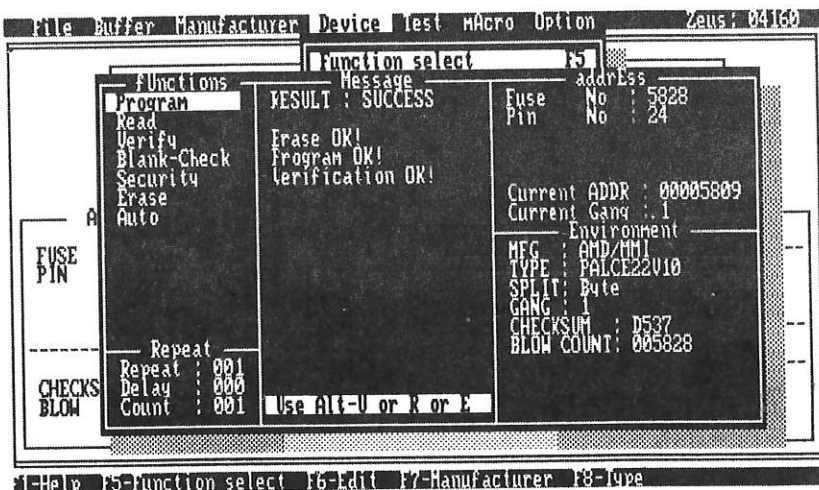
Figure 4.2 Program (PAL/GAL)

In the "addRess" section the address will be increased while the chip
is being programmed or verified. The start address and the end
address of the buffer can be corrected in the addrEss section.
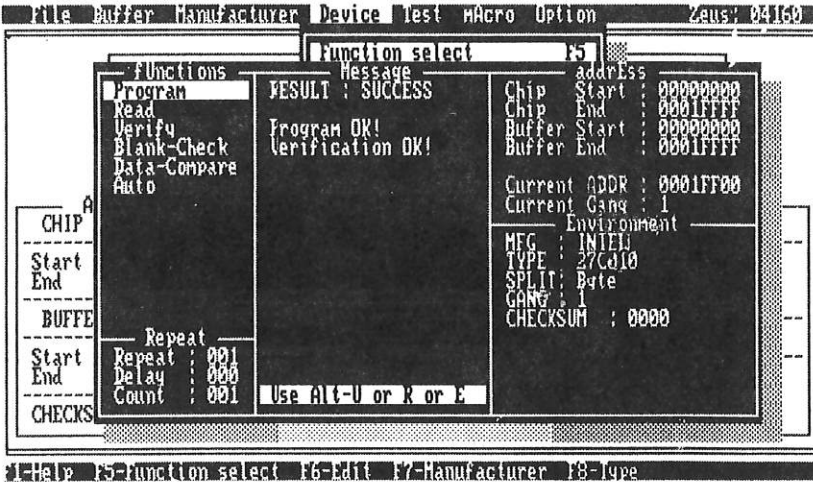
Figure 4.3 Program (ROM/Single)

2. Read

This reads the content of a chip into the buffer. After reading is done, in the address section, the checksum of the data will be displayed. If the chip is a PAL or GAL the blow count will be shown also.

When a GAL is programmed the device should match the manufacturer and the part name selected by the software. Otherwise, an error message will be displayed and the chip will not be programmed. If the security fuses are blown in a PAL or a GAL, the data read from the chip will be all 1's or 0's regardless of what the content is.
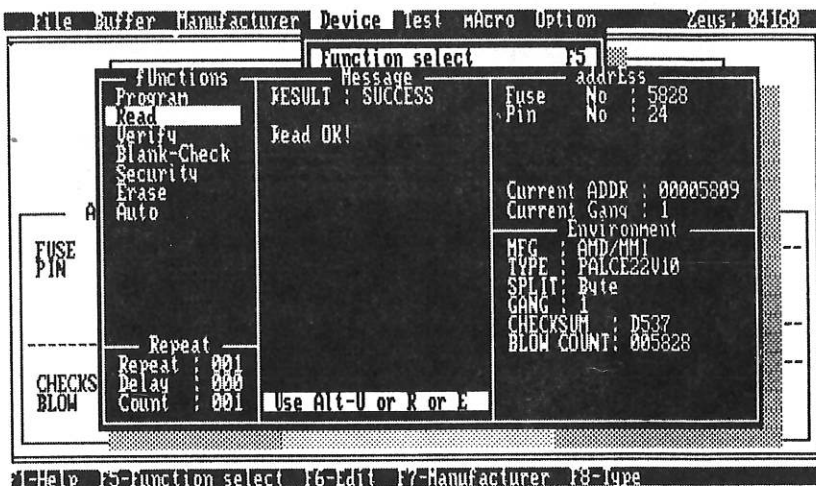
Figure 4.4 Read Menu (PLD)

If the chip is Rom or Single chip the data between the start address and the end address will be read into the buffer. The address being programmed will be displayed and the message will be displayed in the section of "message"
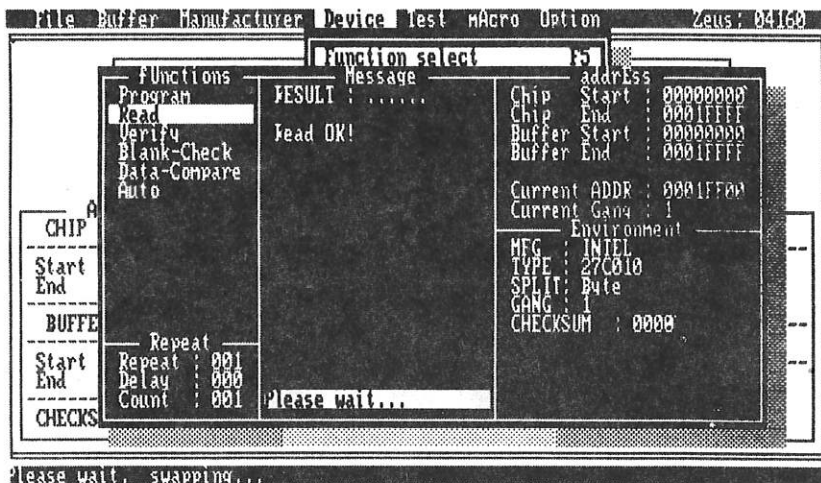
Figure 4.5 Read (ROM / Single)

3. Verify

This compares the content of the buffer and the content of the chip. If an error occurs, it displays the error message and the address where it failed. If the chip is a ROM of Single chip, it verifies between the start address and the end address. The address being verified will be shown in the screen, being increased.
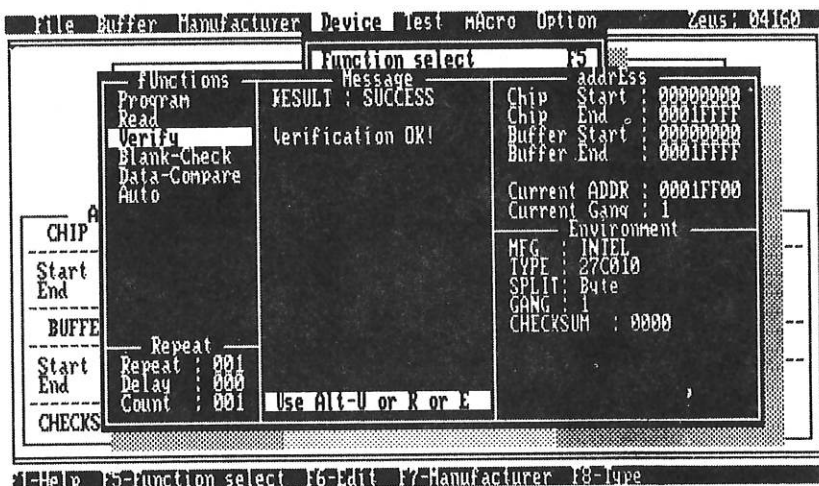
Figure 4.6 Verification

4. Blank check

If reads the content of the data and compares it with the blank
characters. if the chip is not blank it will display the discrepancy with
the address. If the chip is a ROM or Single chip partial blank check is
possible by Indicating the start and end address.
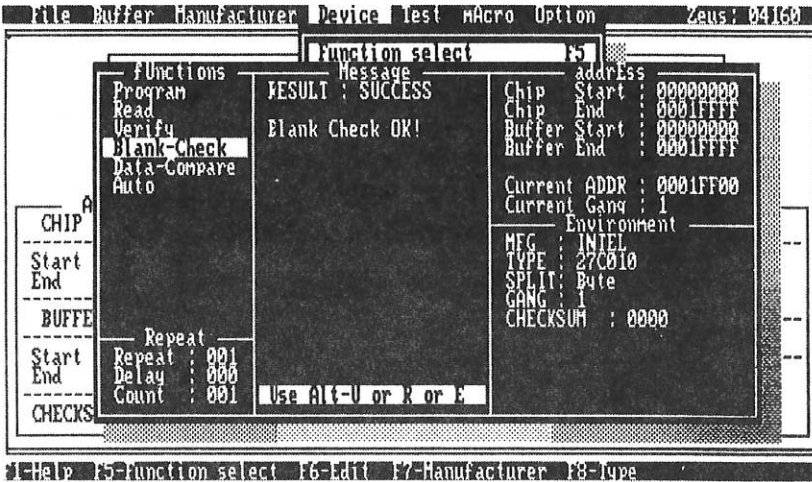
Figure 4.7 Blank Check

5. Data Compare

This menu only applies to the ROM's and single micro-controllers. This is the same as "Verify" menu except that this will generate the file which will contain all the differences between the data of the chip and the buffer. After the execution of the menu of "Data Compare" the file name, which is the name of a device selected, with the suffix of "cmp" will be created in a current directory.

For example, if AMD 27256 has been selected in the softwre, the file created will be 27256. cmp. The file called 27267. cmp can be viewed in a regular editor and contain all the differences between the data of the chip and buffer. Not like the "verify" it will not stop in the first difference it encounters, but it will continue checking.
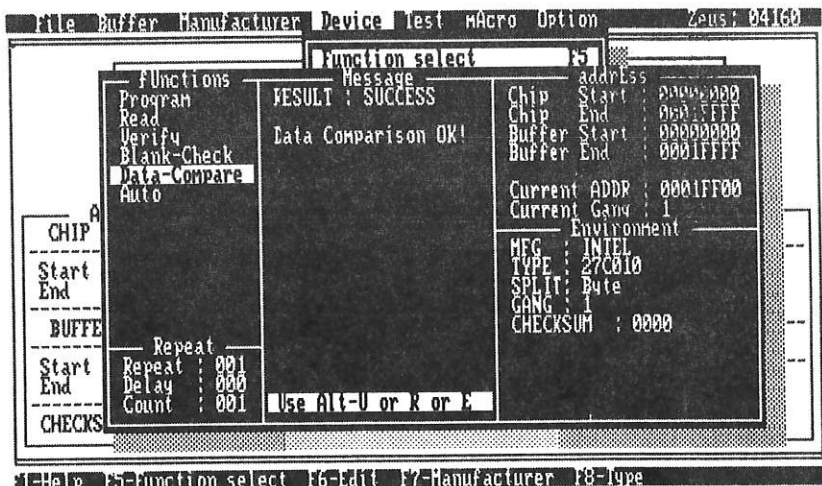
Figure 4.8 Data Compare

6. Auto

This will execute many menus in a sequence. If the chip is a PAL or Gal this will execute Erase, blank check, program, verify and security.

If any of the menu is interrupted by an error, the next step will not be executed. If the chip is a ROM or Single chip this will execute Blank check, Program, and Verify. For the series of 87 Single micro-controllers encrypting is possible too.
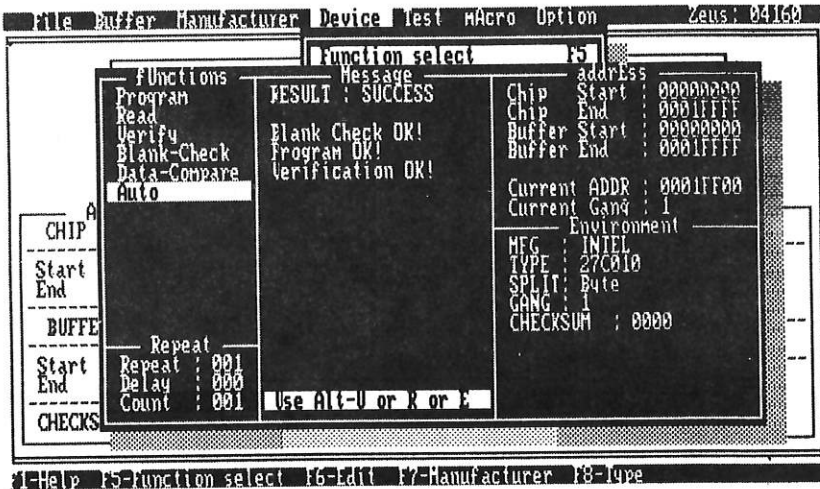
Figure 4.9 Auto

7. Security

If the security bit is programmed the chip will be recognized as a
blank chip. Also, programming will not be possible. The security
function will apply only to PAL's and GAL's.
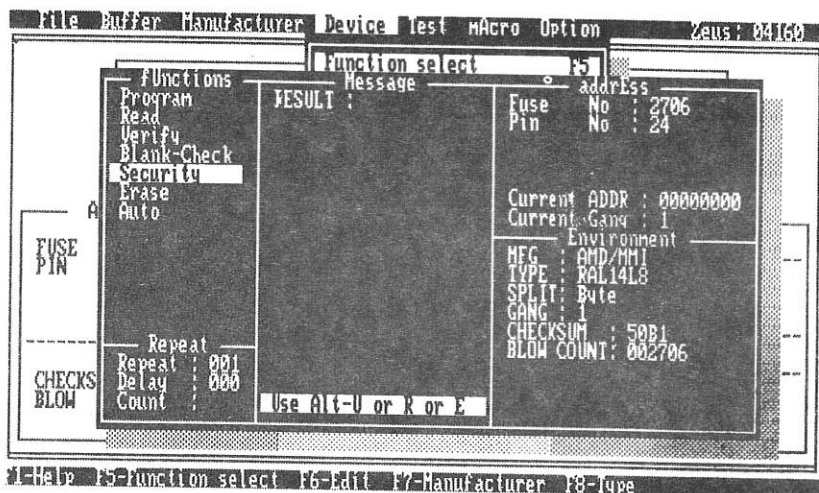
Figure 4.10 Security

8. Encryption program

This only applies to single chips. This will program the content of the
encryption table onto a chip. The content of the encryption table can
be loaded, saved and edited.

Once the data of encryption is written the data in the main buffer will
be Exclusive NORed with the data in the encryption table. If there is
an error, an error message will be displayed.
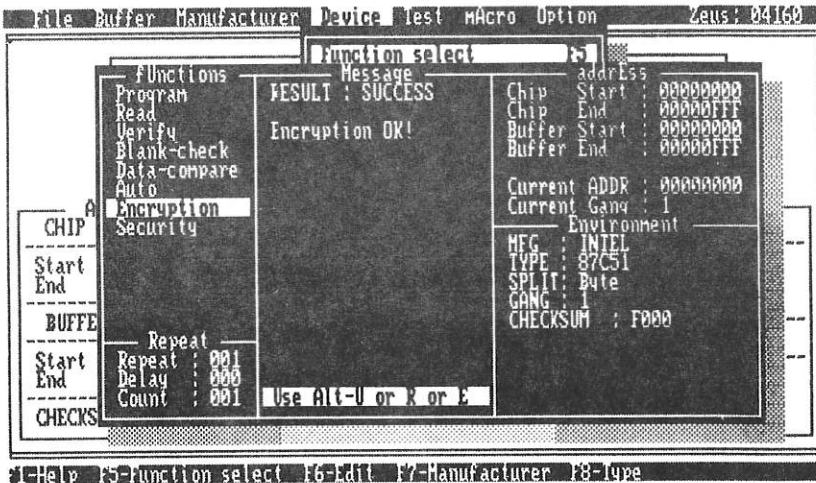
Figure 4.11 Encryption

9. MES Read

In GAL, Manufacturer's Electronic Signature contains the information
for the chip. The data other than the main data will be assigned to
the external buffer, and can be viewed through the menu, "Buffer ->
External Edit".

B. Repeat

This determines the number of executions for a function to be repeated.
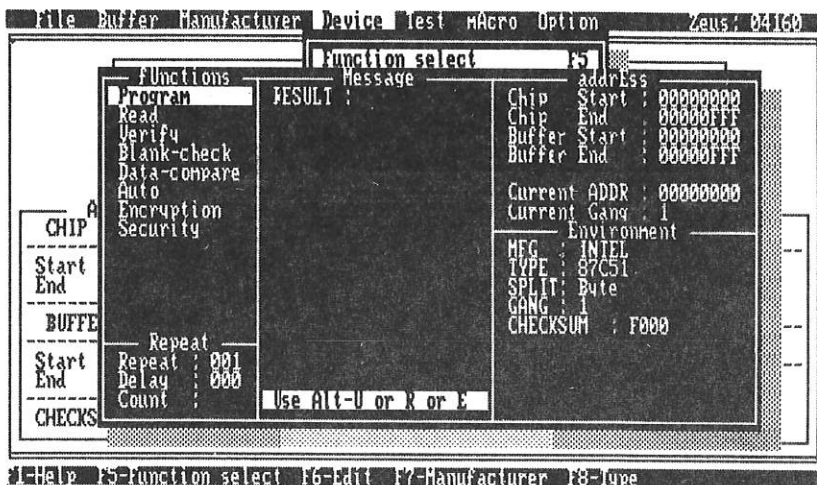
Figure 4.12 Repeat

1. Repeat

Users can determine the number of repetitions.

2. Delay

When users repeatedly use "Function Select" to program the same kind of devices many times this sets the time for pause between executions. In the pause a chip can be replaced.

3. Count

The count of chips programmed will be displayed

C. Message

The message for success or failure will appear.

D. Address
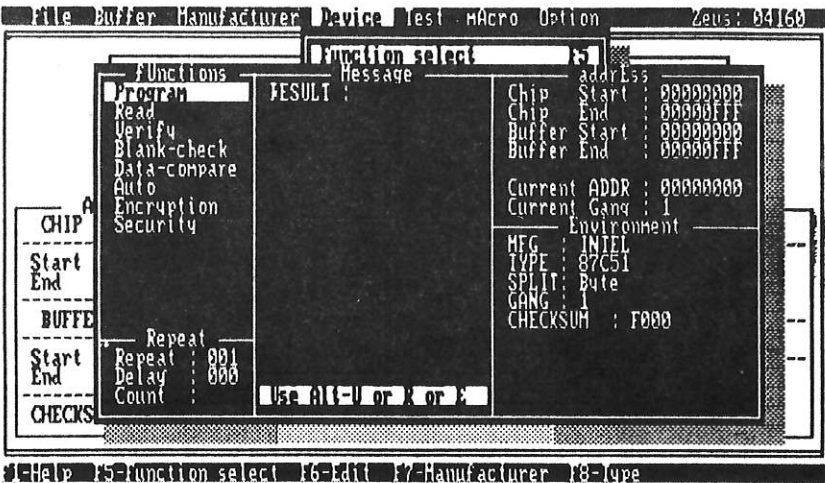
Figure 4.13 Address

1. GAL

In programming GAL, this will show the number of fuses and pins of the chip in question, and it will show the current address which is being read, verified, compared, and etc. Also, the number of gangs selected will be shown.

2. ROM

In programming ROM's or Single microcontrollers, the entries in the address box will be displayed as explained below.

2.1 Chip Start

The address in the chip where programming will start will be designated.

2.2 Chip End

The end address where programming will stop will be designated. Hexa decimal values will be input. The address bigger than the last address of a chip will not be accepted.

2.3 Buffer Start Address

The Start address of the buffer to be programmed will be designated.

2.4 Buffer End Address

The last address of the buffer to be programmed will be designated.

The address increment and the number of gangs will be shown as in programming GAL's.

E.   Environment

The information such as check sum, the number of gang selected, the way of byte splitting for programming, the name of a manufacturer and the part name of the device selected will be displayed.

2. Gang & Word Format

This menu determines the number of gangs and the type of word format. The word format command will configure the way of retrieving the data in the current buffer. The number following the capital G in the menu is the number of gangs selected. There is an optional four socket adapter for purchase. This is based on the assumption that users use the four socket adapter. The four socket adapter will program each gang serially.

Figure 4.14 Gang & Word Format

A. G1-Byte

This is for one gang and programs a byte (8 bits) at a time.

B. G1-Even Word

It is used for one gang and processes two bytes (16 bits) at a time. Since "Even" is indicated it will program the even bytes. The definition of the even byte will be explained as below.

(Example)

Given:

Address of Buffer	Data of Buffer
00	01
01	23
02	45
03	67
04	89
05	AB
06	CD

Result after programming :

Address of Chip	Data of Chip
00	01
01	45
02	89
03	CD
.	.
.	.
.	.

C. G1-Odd word

It is used for one gang and process 2 bytes (16 bits) at a time. Since "Odd" is selected it will program the odd bytes. The example is as follows.

(Example)

Given:

Address of Buffer	Data of Buffer
00	01
01	23
02	45
03	67
04	89
05	AB
06	CD
07	EF

Result:

Address of Chip	Data of Chip
00	23
01	67
02	AB
03	EF
.	.
.	.
.	.

D. G1-Double word 0

This is used for one gang and processes 4 bytes (32 bits) at a time. This will program the data of the buffer in the addresses of 0th, 4th, 8th, ... and so on.

(Example)

Given:

Address of Buffer	Data of Buffer
00	01
01	23
02	45
03	67
04	89
05	AB
06	CD
07	EF
08	FE
09	DC
0A	BA
0B	98
0C	76

Result:

Address of Chip	Data of Chip
00	01
01	89
02	FE
03	76
.	.

E. G1-Double word 1

This is used for one gang and processes 4 bytes (32 bits) at a time. This will program the data of the buffer in the addresses of 1st, 5th, 9th, ... and so on.

(Example)

Given:

Address of Buffer	Data of Buffer
00	01
01	23
02	45
03	67
04	89
05	AB
06	CD
07	EF
08	FE
09	DC
0A	BA
0B	98
0C	76

Result:

Address of Chip	Data of Chip
00	23
01	AB
02	DC
.	.
.	.

F. G1- Double word 2

This is used for one gang and processes 4 bytes (32 bits) at a time. This will program the data of the buffer in the addresses of 2nd, 6th, 10th, ... and so on.

(Example)

Given:

Address of Buffer	Data of Buffer
00	01
01	23
02	45
03	67
04	89
05	AB
06	CD
07	EF
08	FE
09	DC
0A	BA
0B	98
0C	76

Result:

Address of Chip	Data of Chip
00	45
01	CD
02	BA
.	.
.	.

G. G1-Double word 3

This is used for one gang and processes 4 bytes (32 bits) at a time. This will program the data of the buffer in the addresses of 3rd, 7th, 11th, ... and so on.

(Example)

Given:

Address of Buffer	Data of Buffer
00	01
01	23
02	45
03	67
04	89
05	AB
06	CD
07	EF
08	FE
09	DC
0A	BA
0B	98
0C	76

Result:

Address of Chip	Data of Chip
00	67
01	EF
02	98
.	.

H. G2-Byte

This is used for two gangs and processes one byte (8 bits) at a time. The same data from the buffer will be programmed for the two gangs. When users read in this mode only the first gang will be read and the data read will be loaded onto the buffer. But every other functions will be executed in both gangs.

I. G2-Word

This is for two gangs and processes two bytes (16 bits) at a time. The even data will be programmed in the first gang, and the odd data will be programmed into the second gang.

(Example)

Given:

Address of Buffer	Data of Buffer
00	01
01	23
02	45
03	67
04	89
05	AB
06	CD
07	EF

Result:

Address of Chip	Data of Gang 1	Gang 2
00	01	23
01	45	67
02	89	AB
03	CD	EF

J. G3-Byte

This is for three gangs and processes one byte (8 bits) at a time. All the functions except "read" will be applied to each gang. When users read in this mode only the chip in the first gang will be read and loaded into the buffer. All the three gangs will be programmed with the same data.

K. G4-Byte

This is for four gangs and processes one byte (8 bits) at a time. All the functions except "read" will be applied to each gang. When users read in this mode only the chip in the first gang will be read and loaded into the buffer. All the four gangs will be programmed with the same data.

L. G4-Word

This is used for four gangs and processes two bytes (16 bits) at a time. the even bytes will be programmed onto the first and the third gangs, and the odd bytes will be programmed onto the second and fourth gangs.

All the functions except "read" will be applied to each gang. When users read in this mode only one chip in the first gang out of four gangs will be read and loaded into the buffer. As the result of programing the first and the third gang will receive the same data, and the second and the fourth gang will receive the same data.

(Example)

Given:

Address of Buffer	Data of Buffer
00	01
01	23
02	45
03	67
04	89
05	AB
06	CD
07	EF

Result:

Address of Chip	Data of Chips In Gang 1	Gang 2	Gang 3	Gang 4
00	01	23	01	23
01	45	67	45	67
02	89	AB	89	AB
03	CD	EF	CD	EF

M. G4-Double word

This is used for four gangs and processes 4 bytes (32 bits) at a time. The first byte will be programmed in the first gang, second byte in the second gang, third byte in the third gang, and fourth in the fourth.

(Example)

Given:

Address of Buffer	Data of Buffer
00	01
01	23
02	45
03	67
04	89
05	AB
06	CD
07	EF
08	FE
09	DC
0A	BA
0B	98
0C	76
0D	54

Result:

Address of Chip	Data of Chips in Gang1	Gang2	Gang3	Gang4
u0	01	23	45	67
01	89	AB	CD	EF
02	FE	DC	BA	98
03	76	54	.	.

5 TEST

The SUPERPRO can test logic IC and RAM memory devices and vector testing. A menu-driven software program provides an easy interface for loading files, editing test patterns, and downloading test files to the device. TTL.LIB 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. To test logic ICs and RAM memories, be certain that the SUPERPRO hardware is properly installed, and that the programmer is turned on.

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

Figure 5.1 Test

1. Edit Pattern

This option is used to edit a test pattern in the library of pattern files, TTL. LIB. By specifying the directory or highlighting and pressing <return> a buffer with the test pattern selected will show-up in the upper left corner of the screen. The Following edit keys are used for editing existing test pattern for the device selected;

The reserved key "Ctrl-F1" will bring out this menu from any menu and any time.

V	: VCC pins
G	: GND (Ground) pins
X	: Used in power pins, Output values will not be tested (Vcc, Gnd)
H	: Output Logic High (Voh)
L	: Output Logic Low (Vol)
C	: Clock pins
1	: Input Logic High (Vih)
0	: Input Logic Low (Vil)
Arrow	: Left,right,up or down
Ctrl-N	: Moves the cursor to the required address
PgUp	: Page Up
PgDn	: Page Down
Home	: Go to the beginning of the line
End	: Go to the end of the line
Ctrl-Home	: Go to the beginning of a screen
Ctrl-End	: Go to the end of a screen
Ctrl-PgUp	: Go to the beginning of the buffer
Ctrl-PgDn	: Go to the end of the buffer
TAB	: Toggles between ASCII and HEX fields
<Return>	: first position of next line
ESC	: Exits buffer

2. Append Pattern

To test a new device that is not included in the TTL.LIB, 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 menu. A new test file will be created in addition to the existing TTL.LIB files. 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.LIB files.

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 the TTL.LIB, or stored in a new test pattern file.

Please refer to the section "Edit Pattern" for detailed explanations on editing keys.

The reserved key "Ctrl-F2" will bring out this menu from any menu and at any time.

3. Delete Pattern

This option allows the users to delete a pattern file from the library of test pattern files, "TTL.LIB". Either type the name of a device in the box provided or <return>, to display all the patterns and select by highlighting an entry.

The reserved key "Ctri-F3" will bring out this menu from any menu and at any time.

4. TTL & CMOS Test

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

The reserved key "Ctrl-F4" will bring out this menu from any menu and at any time.

5.Auto Find Device

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

The reserved key "Ctrl-F5" will bring out this menu from any menu and at any time.

6. Memory Test

This option tests static and dynamic RAM memory devices. A list of memory devices is displayed then select a type and press **Enter** to perform the test.

The reserved key "Ctrl-F6" will bring out this menu from any menu and any time.

7. Vector Test

This performs the vector testing for GAL's and PAL's. If error occurs the message will display. A vector file will be loaded automatically if it is included in a JEDEC fuse map file.

The reserved key "Ctrl-F7" will bring out this menu from any menu and at any time.

6 MACRO

This makes a macro file and a batch file for automation. A macro file would be created by the sequence of menu selections between the BEGIN and END period of the menu. The menu selected and executed will be memorized by the program and assigned to a special function key specified by the user prior to the beginning of the macro making. After the macro file creation, by pressing the function key users can repeat the memorized sequence. After a macro file is loaded, the file name is displayed in the lower middle corner of the main screen. To stop the execution of a macro file press Ctrl-Break.

Figure 6.1 Macro

1. Begin

In Superpro making a macro file is done by actual key code pressed during the period which is partitioned by this menu and the "END" menu. If this menu is selected then a box will be displayed asking what key would be assigned for a macro file which is about to be created.

The reserved key "Alt-C" will bring out this menu from any menu and any time.

2. End

This indicates the end period of the macro file making.

The reserved key "Alt-Q" will bring out this menu from any menu and at any time.

3. Load

When this menu is selected, a box will ask for a macro file name to be loaded.

The reserved key "Alt-F3" will bring out this menu from any menu and at any time.

4. Save

This saves the macro file created under the name specified by an user.

The reserved key "Alt-F6" will bring out this menu from any menu and at any time.

5. Batch Make

Instead of assigning a sequence of executions to a function key, users can assign the sequence to a batch file and run it as soon as the program is invoked. Follow the steps below.

1. Select the menu, "Macro->bAtch make". A box for a batch file name will be displayed.

2. Specify a name for the batch file and press the return key. A message, "Press <Alt-Q> to end Macro" at the bottom line of the screen is displayed.

3. Start selecting menus and execute the necessary steps.

4. Press Alt-Q to end the batch file making.

5. Go to the menu, "Option->Environment->bAtch file name" and press the return key to specify the batch file just created.

6. Next time as soon as the software is invoked, the batch file will run.

Note: Users can set up the time delay between the executions by selecting "Option->Environment->maCro delay time" and specifying the amount of time to be delayed.

7 OPTION

This menu deals with optional settings such as port address change, directory change, all the optional menu settings, and user-setup-configurations.

Figure 7.1 Option

1. Interface Port

This selects a 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 HEX. If the port settings for the interface and the software are not matching, "Communication Error" message will be displayed.

Figure 7.2 Interface Port

2. Directory

This specifies the directory where the LOPS files and libraries are.

Sub menus ;

System directory
Library directory

Note: It is mandatory that after the installing of Superpro in a directory, the name of the directory name should be entered in this menu.

3. Environment

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

Sub menus ;

> HEX Load Method
> Blank Char
> Multiple Load
> Batch File Name
> Macro Delay Time

3.1. Hex Load Method

This applies to HEX files. "Sequence" is used to load files with sequentially increasing addresses, and "Random" is used for the files without sequence. If you use the wrong option the suggestion will be displayed.

3.2. Blank Char

This sets the standard characters for the default blank characters. In general, the default state is FF but for the 8748 series the default state is 00. Only 00 and FF are acceptable.

3.3. Multiple load on

This mode should be on whenever a user deals with binary files and wants to load many small files in the different locations of the buffer.

3.4. Batch File Name

If there is a batch file name set as soon as the program is invoked the batch file will be executed.

3.5. Macro Delay Time

On executing a macro file this gives the specified amount of the delay between unit executions.

4. Load Configuration

This menu will load the file called lops.cfg which has all the information about the company, chip type and name, port address, and setups for all the option menus.

5. Save Configuration

This menu will save the file called lops.cfg which records all the information about the manufacturer, chip type and name, port address, and setup for all the option menus.

APPENDIX

A   JEDEC FILE FORMAT

XELTEK's SUPERPRO 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   HEX FILE FORMAT

1. INTEL Format

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 without 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.

2. Motorola Format

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.

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.

C   CUSTOMER SUPPORT

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.

1. 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?

2. 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:

Pre-Call Check List

.   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.

.   If there seems to be a problem with the software, can you reproduce it?

.   If the program has displayed an error message, please write down the message.

.   When calling in for technical assistance, have your invoice number ready.

.   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).

.   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.

3. 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.

4. 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.

5. 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 product.

6. 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.

D SELF MAINTENANCE

COMMUNICATION & TRANSISTOR TEST

Note: This test only works with all the port address. The default address is the fifth jumper which is 280 HEX.

Testing

1. Set up your SUPERPRO as if you are programming, but switch up the lever without any chip in the DIP socket.
2. Find out about the model of your programmer and the jumper position of the interface card.
3. Invoke the test software for the universal programmer which is designed to test both Unipro and Superpro. The main menu is as follows.

    A. Communication Check
    B. Transistor Check
    C. Change Mode: Two selections are available - Unipro and Superpro
    D. Change Port: Eight selections are available - 1. 200H, 2. 220H, 3. 240H, 4. 260H, 5. 280H, 6. 2A0H, 7. 2C0H, 8. 2E0H.
    E. By selecting the option 4 match the port address of the software and the card.
    F. By using the option 3, select the model.
    G. After the port address and the model have been set, select the option number 1, and press the return key. This will check the communication of the PCB of the programmer and the test result will be displayed.

4. Next, mount the "testor", which is the array of 33 (ohm) resistors, and pull down the lever to lock the chip in the socket. Select the option number 2 of the menu. If you don't have the "testor", you 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 path.

Repairing

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

1. Open the programmer.

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(NPN), and 2N 4403(PNP).

4. Test Unipro again following procedure A.

These procedures will solve 65% of problems assuming you have updated software. If you have further questions, please contact the division of technical support at (408) 745-7974 or (800) 541-1975.