HI-LO SYSTEM

USER'S MANUAL
ALL-03
PC-BASED UNIVERSAL PROGRAMMER

USER'S MANUAL

ALL-03

PC-Based UNIVERSAL

Programmer

Manual Released Version : V3.0
Date : Apr. 1990

BI-LO SYSTEM

TABLE OF CONTENT

5. USING THE UTILITY FILES

1. INTRODUCTION

1.1. MANUAL CONTENTS

This manual describes the method of installing and operating the UNIVERSAL Programmer on IBM PC running MS-DOS or PC-DOS.

The user is assumed to be an experienced user who is familiar with the general problem of software installation on the PC.

This manual also contains information about the UNIVERSAL Programmer usage and detailed function description.

1.2. PRODUCT CONFIGURATION

Before using this product, carefully check the contents of the package as follows :

* PC System Adapter Card (SAC-201).
* 1 meter long cable with two 25 pins D-type connectors on cable ends.
* Main module.
* This manual.
* 4 diskettes include files :

#1 diskette:

README1.DOC	: Software & Hardware Revision message.
EPP512.EXE	: Software of EPROM under 512 K bits.
EPP1024.EXE	: Software of EPROM over 1 M bits.
BPPGM.EXE	: Software of BIPOLAR PROM.
PGM48.EXE	: Software of 8748 series MPU.
PGM51.EXE	: Software of 8751 series MPU.
PGMZ8.EXE	: Software of Z8 series MPU.

#2 diskette:

README2.DOC	:	Software & Hardware Revision message.
PALPx.EXE	:	Software of PAL devices. x : from 1 to 5.
FPLP1.EXE	:	Software of FPL devices.

#3 diskette:

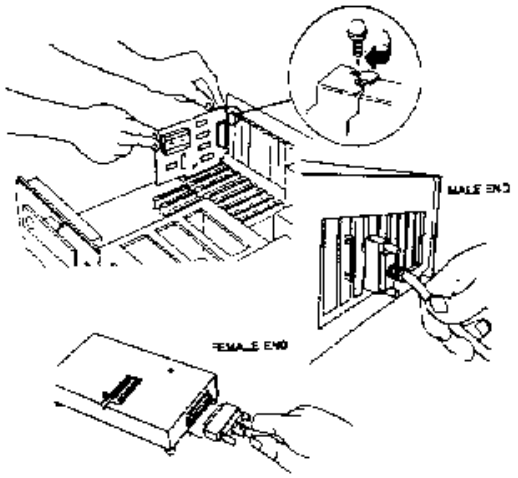
README3.DOC	:	Software & Hardware Revision message.
EPLD1.EXE	:	Software of ALTERA,TI EPLD.
GAL1.EXE	:	Software of LATTICE,NS GAL.
PEEL1.EXE	:	Software of ICT PEEL.

#4 diskette:

README4.DOC	:	Software & Hardware Revision message.
ICTEST.EXE	:	Software of IC TESTER.
TTL74.LIB	:	Library of TTL 74 series.
CMOS40.LIB	:	Library of CMOS 40 series.
CMOS45.LIB	:	Library of CMOS 45 series.
7406.VEC	:	Sample VECTER of TTL 7406.
4040.VEC	:	Sample VECTER of CMOS 4040.

UTIL <DIR> : Subdirectory of UTILITY files includes :

DUMP.EXE	:	DUMP file to console in BINARY format.
HEXBIN.EXE	:	HEX TO BINARY file converter.
HEXBIN2.EXE	:	EXTENDED HEX TO BINARY file converter.
SPLIT2.EXE	:	2 ways file splitter.
SPLIT4.EXE	:	4 ways file splitter.
SHUFF2.EXE	:	2 ways file shuffler.
SHUFF4.EXE	:	4 ways file shuffler.



Product outline

2. INSTALLATION

This chapter discusses the method of installing a UNIVERSAL Programmer on an IBM PC/XT/AT computer or compatible under the MS-DOS or PC-DOS operating system.

It is assumed that the installation is being performed by someone who is familiar with installing PC add-cards and software under MS-DOS.

2.1. HOST SYSTEM REQUIREMENT

* IBM PC/XT/AT or compatible PC.
* MAX. System Clock speed is up to 25 MHZ. zero memory wait state.
* Min. 640K bytes memory.
* Min. 1 floppy disk drive.
* Operating system : MS-DOS or PC-DOS,version 2.0 or later.

2.2. INSTALLATION PROCEDURE

Before doing the installation procedure, it is necessary for the user to make and use a work-copy disk of the original software diskette. **Do not use the original software diskette** ! If you attempt to use the original diskette, your monitor will display a "disk write error" message whenever you terminate the main software.

Follow these steps to install the UNIVERSAL Programmer and the distribution software.

1. Switch off your computer system, and open the computer cover carefully.

2. Check the DIP switch of PC system adapter card(SAC-201).

for I/O address selection: 2E0H (default)
SW1 : all off.
SW2 : position 7 on,others off.
for I/O wait state selection:
SW3 : position 2 on. 4 waits (default)

3. Insert the card(SAC-201) gently into the PC slot, and fasten it to the PC frame with the slot cover screw.

4. Finally, connect the main module to the SAC-201 card using the attached cable. The male cable end must be inserted into SAC-201 card, the female end must be connected to the main module.

CAUTION

Do not connect the main module to the SAC-201 card while computer is turned on. Such an installation can put the module in an "unknown" state, resulting in damage to the DEVICE and/or the module.

5. Turn on the computer and check LEDs on main module.

ON LED must be ON.
BUSY LED must be OFF.
Other LEDs are in random state.

If the LEDs are not in the correct state, please turn off the PC and check all the connections between SAC-201 and PC slot, cable connections between SAC-201 and main module.
Then try this step again.

6. Booting DOS in disk drive a:
Under the operating system prompt, replace DOS diskette with work-copy diskette and type following command : (EX. of PAL software)

A>PALP1 <CR>

<CR> is RETURN or ENTER key.

NOTE

PALP1 is one of the PAL programming software. You may execute other software to program other DEVICE.

The main software will be executed, and will look for the parameter file PALP1.DAT. If it exists in the default disk drive, the parameters of I/O address,Manufacturer,Type, Programming speed will be loaded and become the default state. Otherwise the default value I/O address(2E0), Manufacture(MMI PAL), Type(10H8/-2),Programming speed (standard) will be considered as the default state.

At the same time, the software will also check the identification circuit in main module. Once it is checked , the software will display the function menu on the screen as follows and wait for your command.

7. If the function menu is not displayed on screen, the work-copy diskette is possibly bad. Remake and use another work-copy diskette to try step 6 again.

NOTE

If this problem continues, please contact the local distributor for help.

8. If the error message appears on the screen as follows:

```
Error identification on hardware!
  Press "Q" to quit,
  or press <CR> to continue...
```



```
* MFG.: CYPRESS PAL
* TYPE: C16R4

PALP1 section V3.00
------------ Main Menu ------------
1. DOS SHELL
2. Load JEDEC file to buffer
3. Save buffer to disk
4. Edit buffer      7. disp buffer
5. Change I/O base adr. of SAC-201

T. Type select      M. MFG. select

B. Blank check      C. Display
P. Program          A. Auto(B&P)
R. Read             V. Verify
S. Security fuse blow
Q. Quit
-----------------------------------

 TARGET ZONE
check sum : 0000

Select which function ?
```

Main menu

The hardware I/O address of SAC-201 is impacted to other add-on card inserted in PC slot.

You may press "Q" to quit to DOS prompt, then turn off the computer and re-check all the connections between SAC-201 and PC slot, cable connections between SAC-201 and main module.

After re-checking all these connections, repeat the installation procedure from step 1.

NOTE

Most problems are due to poor contact between SAC-201 and PC slot,cable connection between SAC-201 and main module.

You may also press the **RETURN** or **ENTER** key to enter into main menu and change the I/O address of SAC-201 to other available address on PC by **Change I/O Base Address** function.

Please ref. to chapter 4 .. Change I/O Base Address function to setup new I/O address.

3. USING THE PROGRAMMER

In this chapter we provide an example which will be described step by step to help you become familiar with your new UNIVERSAL Programmer tools. It is assumed that the installation procedure described in the previous chapter has been performed.

3.1. DEFINITION OF SYMBOL

Let us first define some commonly used symbols. You may later on refer to them whenever necessary.

3.1.1. Symbol of EPROM,BPROM and MPU DEVICES

*** BIT,NIBBLE,BYTE,WORD,ADDRESS and BUFFER BASE ADDRESS**

BIT : Element of Binary data.
NIBBLE : 4 BITS Binary data. Value from 0H to FH.
BYTE : 8 BITS Binary data. Value from 0H to FFH.
WORD : 16 BITS Binary data. Value from 0H to FFFFH.
ADDRESS: Location of data on buffer. Value from 0H to FFFFFH.
BUFFER BASE ADDRESS : The actual physical address of buffer.
Value from 0000:0000 to FFFF:0000

*** MEMORY BUFFER**

The buffer is a block area of PC memory allocated by the main software through DOS. This buffer is used by the main software as an intermediate storage.

The main software can read DEVICE contents to the buffer and save it on a disk file or perform the reverse operation, that is to load a disk file to buffer and program it to DEVICE. You have the advantage of manipulating the buffer contents at will. The modified buffer can always be saved to a disk file for future reference.

The minimum size of buffer is 64K bytes, and the maximum size is the maximum available memory in PC.

Since memory buffer is dynamically allocated through DOS, the actual base address on PC may vary from system to system, software to software. User will not have to refer to the actual base address of the memory buffer.

*** BUFFER START and BUFFER END ADDRESSES**

The buffer start and end addresses are offset addresses specified from the base address of the MEMORY BUFFER.

This is the specified portion within which information can be programmed to DEVICE and the DEVICE contents can be read into this portion.

*** DEVICE START and DEVICE END ADDRESSES**

The start and end addresses are offset addresses specified from the DEVICE contents.

*** CHECK SUM**

This is the sum of all data contents between buffer start and end addresses. This value will be calculated during DEVICE reading, file loading, type changing or after buffer editing.

Bits count of the data contents:

```
NIBBLE WIDE PROM is 4.
BYTE   WIDE PROM is 8.
WORD   WIDE PROM is 16.
MPU is 8.
```


*** EVEN and ODD address mapping**

The address sequence of data contents can be assigned to **CONTIGUOUS,EVEN or ODD** whenever you want to **READ,PROGRAM or VERIFY** the EPROM.

For ex. : In program function sub-menu, the software will ask you :

Ready to start (Y/Even/Odd/<CR>) ?

You may press **<CR>** to go back to main menu.

You may press **Y** to program the data in buffer to device **CONTIGUOUSLY** as follows :

Buffer start	+ 0	to	Device start + 0
	+ 1	to	+ 1
	+ 2	to	+ 2
	...	to	...

You may press **E** to program the data in **Even** address to device as follows :

Buffer start	+ 0	to	Device start + 0
	+ 2	to	+ 1
	+ 4	to	+ 2
	...	to	...

You may press **O** to program the data in **Odd** address to device as follows :

Buffer start	+ 1	to	Device start + 0
	+ 3	to	+ 1
	+ 5	to	+ 2
	...	to	...

In READ operation, the software will perform in the reverse direction.

* **I/O ADDRESS**
This is the I/O base address of system adapter card SAC-201.
Any I/O interface card added on PC slot will occupy one or more I/O addresses.
The default I/O base address of SAC-201 is 2E0 and occupies 4 contiguous spaces (2E0 to 2E3).

* **COUNTER**
This is the programming address counter. During DEVICE programming, the counter value will be displayed increasingly on screen.

* **MFR,TYPE,VPP,SPEED**
Every DEVICE has its own manufacturer(MFR) , type number(TYPE),programming voltage(VPP) and programming speed (SPEED or algorithm).
Please refer to APPENDIX A for the differences between each DEVICE.
Please also refer to chapter 4.. for the detailed description.

3.1.2. Special symbol of PAL,FPL,GAL,PEEL and EPLD DEVICES

* **Programmable Logic Device (PLD)**
Generally speaking, a device that can be programmed to perform many different logic operations is a PLD.

There are only 3 kinds of PLD as follows :
PLD : One time PLD such as PAL devices.
EPLD : UV erasable PLD such as EPLD devices.
EEPLD : Electrical erasable PLD such as GAL, PEEL devices

* **JEDEC Fuse Map file of PLD DEVICE**
The JEDEC Fuse Map file is the standard format which can be loaded by the programmer. It contains fuses (BLOWN/INTACT) information and

FUNCTION TEST VECTOR of the PLD. Most PAL assemblers or compilers, such as PALASM,PLAN,ABEL,AMAZE and PDK-1, will produce JEDEC Fuse Map file.

* **Array Fuses, configuration fuses**

Array fuses are the main logic fuses in PLD. Different arrangement (BLOWN/INTACT) will have a different logic combination.

The configuration fuses always indicate the I/O architecture of the PLD , such as COMBINATORIAL/REGISTERED, OUTPUT FEED BACK/OUTPUT ENABLE.

* **Security Fuse**

Comparing with the PROM device, there is an extra fuse inside the PLD. The arrangement of array fuses and configuration fuses will not be normally copied after the security fuse is blown.

* **Fuse BLOWN and INTACT**

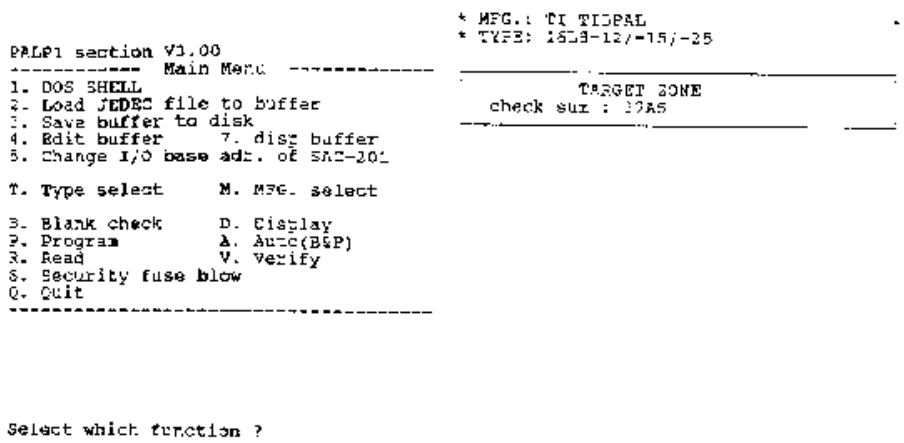
Originally in JEDEC Fuse Map definition, the fresh PLD(Blank device) are all in Fuse INTACT state (eg. 0 state). Designer can only blow the INTACT fuses into BLOWN state(eg. 1 state).
Due to the fast growth of PLD technology, some fresh PLDs are in Fuse BLOWN state, and can only be programmed into INTACT state.
There is no way to return to fresh state from unfresh state, except that the PLD is erasable.

3.2. VIEWING THE MAIN MENU

Before starting, study the main menu displayed on the screen to familiarize yourself with the menu driven features.

3.2.1. Main function menu

On the left side of the screen there is a **Main function menu.** shown as follows :



Main menu screen

3.2.2. Status field

On the right side of the screen you will find another group of text. From top to bottom :

Current DEVICE's MFR. TYPE VPP programming speed, working buffer and device address, ... we call this **status field.**

Whenever you are going to use this programmer, please make sure that the statuses in this field meet your requirement. Otherwise the DEVICE will be destroyed or programmed to an unknown state.

3.2.3. LOGO, HARDWARE MODEL and SOFTWARE VERSION

On the upper left side of the screen , you will see the LOGO, HARDWARE MODEL and SOFTWARE VERSION.
These messages are for reference only.

By now you should be familiar with the main menu screen. You can find out more about these features by reading the rest of this section. Functions will be discussed in detail.

3.3. GETTING STARTED

This section presents a simple example to help you become familiar with the programmer and some commonly used functions. Exercise and experiment with the suggested option. Look up individual function description in Chapter 4. If you do not get desired results, make sure all the cables are connected firmly. Always double check cable connection.

The following example will demonstrate how to start and exit the main menu and use functions such as **MFR TYPE LOAD BLANK** and **PROGRAM.**

In the following example, we are going to use a TIBPAL 16L8-12 PLD. You may also use other series PLD or EPROM devices to practice this exercise.

By now, you should have at least one TIBPAL 16L8-12 PLD on hand.

3.3.1. Execute the proper software

Your first step should always be to execute the proper software included in your work-copy diskette. Please ref. to APPENDIX A. for detailed description about Device list of each software.
For the TIBPAL 16L8-12 PLD, please execute software "**PALP1.EXE**" by typing text as follows :

A> PALP1 <CR>

The main function menu will appear on the screen as follows :

```
                                        * MFG.: TI TIBPAL                +
                                        * TYPE: 16L8-12/-15/-25

PALP1 section V3.00                     ┌──────────────────────┐
------------ Main Menu -------------     │      TARGET ZONE     │
1. DOS SHELL                            │ check sum : 37A5     │
2. Load JEDEC file to buffer            └──────────────────────┘
3. Save buffer to disk
4. Edit buffer       7. disp buffer
5. Change I/O base adr. of SAC-201

T. Type select       M. MFG. select

B. Blank check       D. Display
P. Program           A. Auto(B&P)
R. Read              V. Verify
S. Security fuse blow
Q. Quit
----------------------------------------

Select which function ?
```

Main function menu

3.3.1. SELECT PLD MANUFACTURER and TYPE

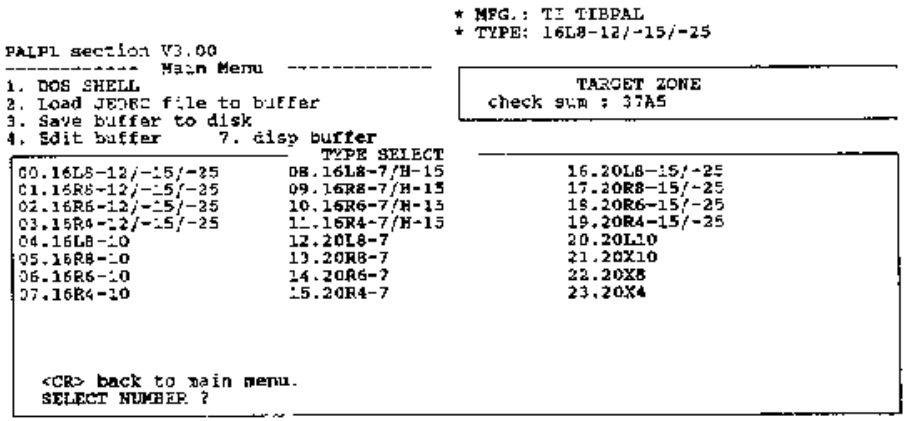
The second step should always be to select the PLD manufacturer and type . Start by typing **M** from the main menu. The screen should display the following sub-menu:


```
                                        * MFG.: TI TIBPAL                +
                                        * TYPE: 16L8-12/-15/-25

PALP1 section V3.00                     ┌──────────────────────┐
------------ Main Menu -------------     │      TARGET ZONE     │
1. DOS SHELL                            │ check sum : 37A5     │
2. Load JEDEC file to buffer            └──────────────────────┘
3. Save buffer to disk
4. Edit buffer       7. disp buffer     ┌──── MFG. SELECT ─────┐
5. Change I/O base adr. of SAC-201      │0. MMI PAL            │
                                        │1. NS DMPAL           │
T. Type select       M. MFG. select     │2. TI TIBPAL          │
                                        │3. AMD AmPAL          │
B. Blank check       D. Display         │4. CYPRESS PAL        │
P. Program           A. Auto(B&P)       │5. SIGNETICS PAL      │
R. Read              V. Verify          │                      │
S. Security fuse blow                   │                      │
Q. Quit                                 │                      │
----------------------------------------│                      │
                                        │  <CR> back to main menu.
                                        │  SELECT NUMBER ?     │
                                        └──────────────────────┘
Select which function ? ■
```

Sub-menu of manufacturer

Type the number of the manufacturer which matches your PLD (for ex.: 2 for TI PLD). The manufacturer should appear and be updated in the **status field**. Then press **<ESC>** or **<CR>** to go back to main menu.

Next, type **T** from the main menu. The screen should display the available types for your particular manufacturer as follows :



Sub-menu of type

Type in the number that corresponds to the type of your PLD (for ex.: 0 for 16L8-12). The type will automatically be updated and displayed in the **status field**

NOTE

Besides selection the of manufacturer and type , user has to take more care about the programming voltage on EPROM devices.

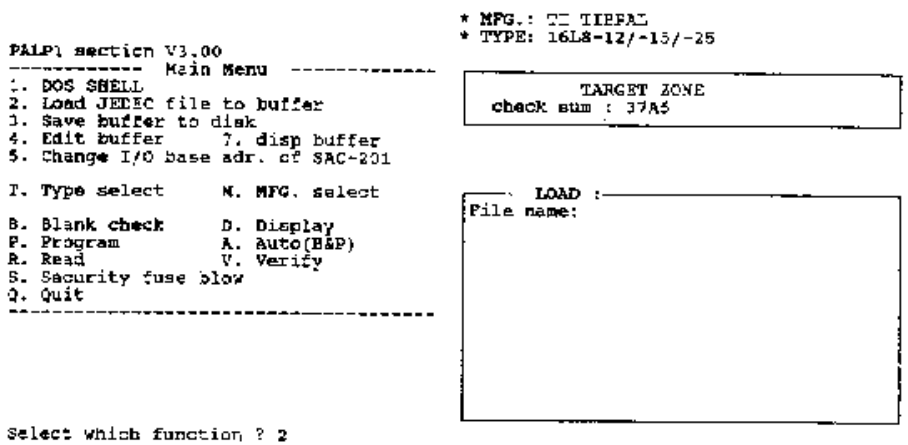
AS EPROM chips are not always clearly labelled, it may be difficult,for example, to determine whether the EPROM you wish to program is a 2764 requiring 21V, or whether it is a 2764A, which requires 12.5V. If you encounter this problem, try to program the EPROM at the lower voltage first. If this does not work, then erase the EPROM and try the higher EPROM voltage.

Please refer to APPENDIX B,C for more information about manufacturer, type and voltage selection about EPROM.

Check to make sure that the values displayed in the **status field** match the specifications of your PLD . If they do, press **<ESC>** or **<CR>** to return to the main menu.

3.3.3. LOAD DISK FILE INTO BUFFER

After you have done the above-mentioned step, transfer your disk file to the memory buffer(the file must be in JEDEC format). To do this,press 2, and the file loading sub-menu will appear as follows :



Sub-menu of file loading

Type in the name of the file you wish to transfer to the memory buffer. The drive letter and any path names must be included.

After the file name has been entered, the software will begin to load the disk file to the internal memory buffer and display :

Loading now...
Ok!

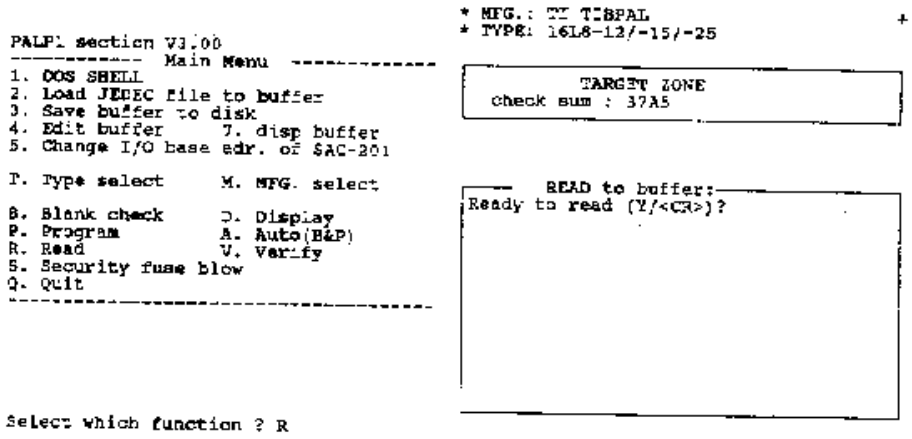
The disk file is now in the memory buffer. Press **<ESC>** or **<CR>** to return to the main menu.

NOTE

Besides entering the filename on EPROM devices, the sub-menu ask you to enter the buffer starting address you want to load from. You must now specify the memory buffer location(address) to which the disk file will be sent. Type 0000 if you wish the disk file to be transferred to the very beginning of the memory buffer.

3.3.4. READ CONTENTS FROM MASTER PLD

If the PLD data is in a MASTER PLD(not in disk file), you have to transfer it by typing R to enter into **READ** function. After pressing R, the following sub-menu will appear:



Sub-menu of READ

Insert the MASTER PLD into textool socket . With the pull stick on top, the lower left pin should match with the PLD **GND** pin.
Then press **Y**,the data of MASTER PLD will be transferred to the internal memory buffer and display:

Reading now...
Ok!

Press **<ESC>** or **<CR>** to return to the main menu.

3.3.5. INSERT THE BLANK PLD INTO TEXTOOL SOCKETS

After transferring the data from disk file or MASTER PLD, take out the MASTER PLD and insert the blank PLD into textool socket, with the pull stick on top, the lower left pin should match with the PLD **GND** pin.

CAUTION

The notch-end of the PLD must not be inserted at the end of the socket, otherwise the PLD will be destroyed or programmed to an unknown state.

3.3.6. PROGRAM BUFFER CONTENTS TO PLD

After loading the disk file or reading the MASTER PLD data into memory buffer, and inserting the blank PLD, you can now program the PLD by typing P.

After typing P, the **PROGRAM** function will be invoked and the following sub-menu will be displayed :

```
                                        * MFG.: TI TIBPAL
                                        * TYPE: 16L8-12/-15/-25
PALPI section V3.00
------------ Main Menu ------------      ┌──── TARGET ZONE ────┐
1. DOS SHELL                            │ check sum : 57A5    │
2. Load JEDEC file to buffer            └─────────────────────┘
3. Save buffer to disk
4. Edit buffer      7. disp buffer
5. Change I/O base adr. of SAC-201
                                        ┌──── PROGRAM : ──────┐
T. Type select      M. MFG. select      │Ready to program (Y/<CR>)?
                                        │
B. Blank check      D. Display          │
P. Program          A. Auto(B&P)        │
R. Read             V. Verify           │
S. Security fuse blow                   │
Q. Quit                                 │
-------------------------------------   │
                                        │
                                        │
                                        └─────────────────────┘
Select which function ? P
```

Sub-menu of PROGRAM

Then press Y, and the programmer will attempt to program the buffer contents onto the blank PLD. At the end of the programming process, the programmer will compare PLD contents with the memory buffer. Any discrepancies between buffer and PLD will be displayed.

This completes the programming process. To program other PLDs, wait for the **BUSY** LED to shut off, then replace the PLD and type Y again.

If you want to exit the programming process,press **<ESC>** or **<CR>** to return to the main menu.

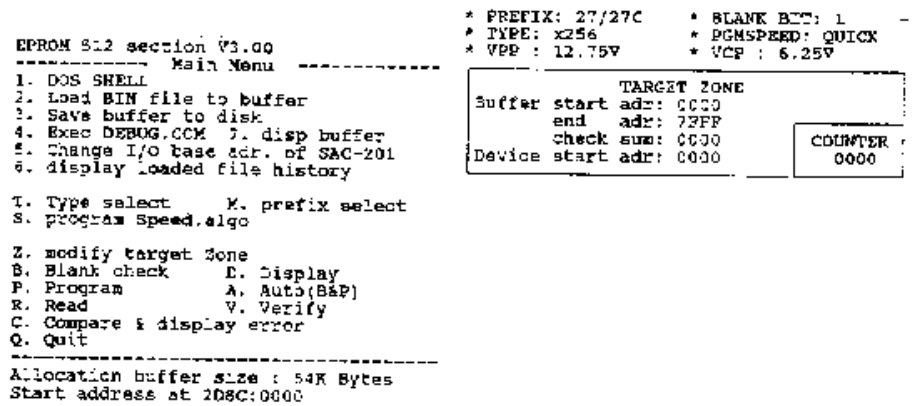
By now , you will understand that it is very simple to program PLDs. We hope this tutorial section will help you to become a professional user of this programmer.

The advanced user may want to modify the contents in memory buffer, to save buffer contents to disk file, to change I/O address to other address.., We will describe each function in detail on Chapter 4 **FUNCTION REFERENCE GUIDE.**

4. FUNCTION REFERENCE GUIDE

The following reference guide describes each of the functions found in the main menu. The functions will be described in the order in which they are listed in the menu. To enter into the desired function, you only press the first character of that function .

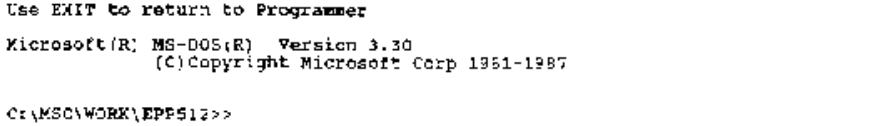
FOR EPROM , BPROM and MPU



Screen of main menu

4.1. Dos shell

Press 1 to select DOS shell function.
After entering into this function, the software will look for the file **COMMAND.COM** at the DOS boot disk drive. If it exists, the software will execute this file and pass control to it and display a DOS command prompt as follows :



DOS SHELL

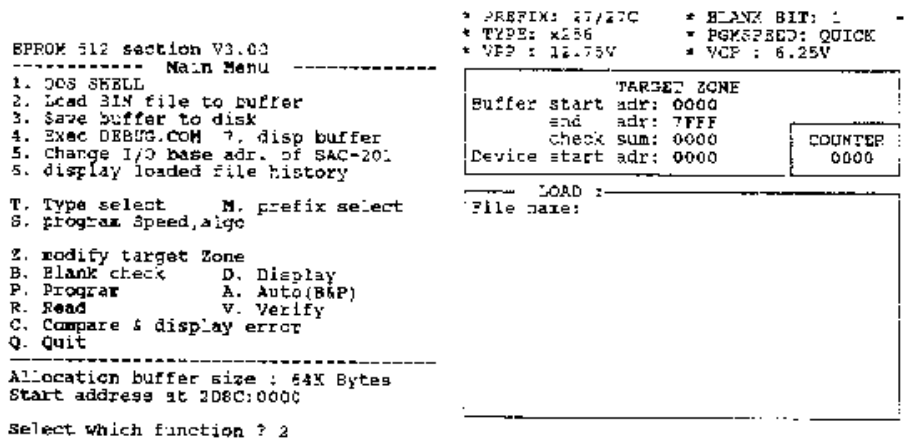
By now, the software is controlled by DOS and waiting for your command. The command format is the same as DOS command.
Under DOS command prompt, enter text **EXIT** and **<CR>** will return to main menu.

NOTE

This function will invoke the COMMAND.COM. User has to prepare this file on the DOS boot disk drive, otherwise the function will not work.

4.2. Load BIN file to buffer

Press 2 to select load file function .
A dialog window will be displayed on screen as follows:



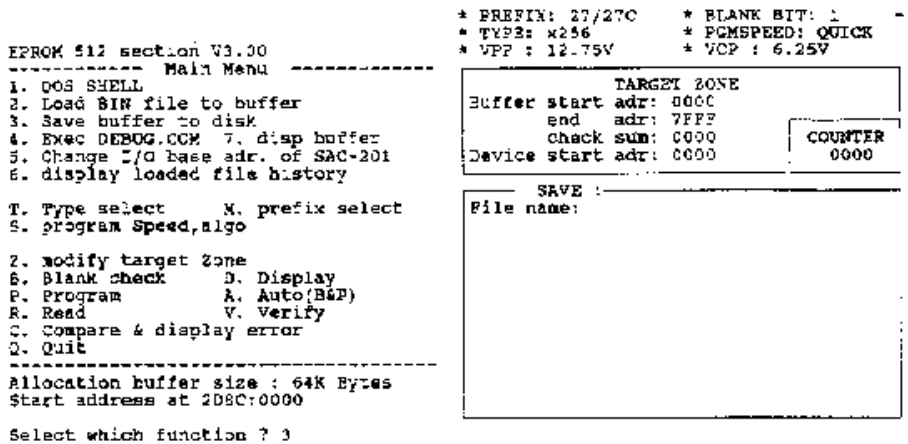
Load file

When prompted,type in the complete name of the file to include the drive letter and any path names, Then press **<CR>**. Next, type in the address at which you want to load the file in the buffer and press **<CR>**. The software will load the file into the specified buffer address.

Press **<ESC>** to return to the main menu.

4.3. Save buffer to disk

Press 3 to select save file function
A dialog window will be displayed on screen as follows:



Save file

When prompted,type in the complete name under which you want the file to be saved, including drive letter and any path names, then press **<CR>**. Next, type in the starting address of the portion of the buffer you want to save,and press **<CR>**. Then,type in the end address of the portion of the buffer you want to save, and press **<CR>**.

The buffer has been saved when **Ok!** message appears. Press **<ESC>** to return to the main menu.

4.4. Edit buffer

Press 4 to select buffer editing function. After entering into this function, the editing command summary will be displayed on screen as follows:

```
            EDITING COMMAND SUMMARY
=========================================================
D [start],[end]           <RETURN> : DUMP
E start                   <RETURN> : EDIT
M start,end,destination   <RETURN> : MOVE BLOCK
F start,end,data          <RETURN> : FILL BLOCK
P start,end               <RETURN> : PRINT BLOCK
C start,end               <RETURN> : CHECK SUM
S start,end,ASCII data    <RETURN> : ASCII  SEARCH MAX. 15 characters
B start,end,BINARY data   <RETURN> : BINARY SEARCH MAX.  7 BYTEs
. filename [argu1] [argu2]...<RETURN> : SHELL
?                         <RETURN> : HELP
Q                         <RETURN> : QUIT
=========================================================
* The information listed below is for reference only :
  The absolute start address of BUFFER : 2D6C:0000
  The Buffer size : 64 K BYTES

==
```

Edit command summary

You may now proceed with the Edit procedure under command prompt == by using the above command format.

Besides the command format, the actual buffer base address is also shown at the bottom of the command summary. It is the real base address of buffer in PC memory and is displayed in **SEGMENT:OFFSET** form.

For special purposes, some advanced users have to know this address and pass it to their own editing program which will edit the buffer contents in different ways.

EXAMPLE : The user's editor is FRED.EXE, then he can use the DOT command to pass the base address as follows :

. FRED SEGMENT:OFFSET <CR>

The user must get this parameter from the command line to his program just like executing his program under DOS command prompt :

A>FRED SEGMENT:OFFSET <CR>

You may return to the main menu by typing **Q** and **<CR>**.

4.5. Display buffer

Press 7 to select display buffer function. After entering into this function, the buffer contents in the range between **BUFFER START ADDRESS and BUFFER END ADDRESS** shown in **status field** will be displayed on the screen, in BINARY format, as follows:

```
Press <ESC> to terminate display

0000  80 0A C0 08 65 70 70 35 --31 32 2E 43 20 3B 07 00 ....epp512.C ...
0010  D0 00 4D 53 20 43 6E 88 --05 00 00 9F 45 4D 42 88 ..MS Cn.....EMB.
0020  09 00 C0 9F 53 4C 49 42 --46 50 10 88 08 C0 00 9F ....SLIBFP......
0030  53 4C 49 42 43 64 8B 07 --00 00 9F 4C 49 42 48 B3 SLIBCd.....LIBH.
0040  88 06 00 00 9D 30 73 4F --E3 B8 06 00 00 A1 01 43 .....0sO.......C
0050  56 37 96 2E 00 00 06 44 --47 52 4F 55 50 05 57 54 V7.....DGROUP.WT
0060  45 58 54 04 43 4F 44 45 --05 5F 44 41 54 41 04 44 EXT.CODE._DATA.D
0070  41 54 41 05 43 4F 4E 53 --54 04 5F 42 53 53 03 42 ATA.CONST._BSS.B
0080  53 53 3F 98 07 00 28 26 --27 03 C4 01 F4 98 07 00 SS?...(&'.......
0090  48 96 0B 05 06 01 6C 98 --07 00 48 0C 00 07 07 01 H.....l...H.....
00A0  0A 98 07 00 48 00 02 08 --09 01 C7 9A 08 00 02 FF ....H...........
00B0  03 FF 04 FF 02 56 9C 0D --00 00 C3 01 02 02 01 03 .....V..........
00C0  04 40 01 45 01 C0 8C CC --01 0A 5F 5F 61 63 72 74 .@.E...._acrt
00D0  75 73 65 64 00 C7 5F 68 --61 6C 6C 6F 63 00 07 5F used.._halloc.._
00E0  63 68 6B 63 75 72 0C 08 --5F 69 6F 5F 6D 51 6E 75 chkcur.._io_manu
00F0  00 06 5F 63 68 6B 69 64 --00 0A 5F 68 69 73 74 5F .._chkid.._hist_
0100  6D 61 6E 75 00 0C 5F 64 --73 70 62 75 66 5F 6D 61 manu.._dspBuf_ma
0110  6E 75 00 0B 5F 73 70 65 --65 64 5F 6D 61 63 75 00 nu.._speed_manu.
0120  C9 5F 6D 66 67 5F 6D 61 --6E 75 00 06 5F 67 65 74 ._mfg_manu.._get
0130  63 68 00 09 5F 74 79 70 --5F 6D 61 6E 75 00 0D 5F cH.._typ_manu.._
Press any key to continue
```

Display buffer

You may press ^**S** to hold the display or press **<ESC>** to terminate the display.

After terminating the display, press any key to return to main menu.

4.6. Change I/O base address of SAC-201

Before performing this function to change to new I/O address, you also have to turn the DIP SWITCHs ON or OFF on system adapter card.(This must be done when the computer is switched off.)

Please take a look at the following list of I/O address and its corresponding DIP SWITCH position. Only one position can be selected **ON.**

	DIP SW	POSITION	I/O ADDRESS
	SW1	1	200H
	SW1	2	210H
	SW1	3	220H
	SW1	4	230H
	SW1	5	240H
	SW1	6	250H
	SW1	7	260H
	SW1	8	270H
	SW2	1	280H
	SW2	2	290H
	SW2	3	2A0H
	SW2	4	2B0H
	SW2	5	2C0H
	SW2	6	2D0H
(DEFAULT)	SW2	7	2E0H
	SW2	8	2F0H

Press **5** to select I/O address function ,
A dialog window will be displayed on screen as follows:

```
                                * PREFIX: 27/27C      * BLANK BIT: 1
                                * TYPE: x256          * PGMSPEED: QUICK
                                * VPP : 12.75V        * VCP : 6.25V
EPROM 512 section V3.30         ---------- TARGET ZONE ----------
---------- Main Menu ----------  Buffer start adr: 0000
1. DOS SHELL                              end   adr: 7FFF
2. Load BIN file to buffer                check sum: 4EC5     COUNTER
3. Save buffer to disk          Device start adr: 0000        0000
4. Exec DEBUG.COM  7. disp buffer
5. Change I/O base adr. of SAC-201
6. display loaded file history
                                ----Change I/O base adr. of SAC-201----
I. Type select      M. prefix select
S. program Speed,algo           0. 200          8. 280
                                1. 210          9. 290
Z. modify target Zone           2. 220          A. 2A0
B. Blank check      D. Display  3. 230          B. 2B0
P. Program          A. Auto(P&P)4. 240          C. 2C0
R. Read             V. Verify   5. 250          D. 2D0
C. Compare & display error      6. 260          E. 2E0
Q. Quit                         7. 270          F. 2F0
-----------------------------------  <CR> back to main menu.
Allocation buffer size : 64K Bytes  Current I/O base-adr: 2E0
Start address at 208C:0000           --- Programmer not existed ---
                                Select number ?
Select which function ? 5
```

I/O select

When prompted, type in the number of new I/O address of system adapter you wish to work with.

The I/O address you have selected will be updated in the internal buffer for programming reference.

Press **<ESC> or <CR>** to return to the main menu. It is necessary to perform the **QUIT** function which will save the new I/O address in the parameter file **????????.DAT** for later use.

4.7. Display loaded file history

Press 6 to select list file information function. The information on loaded file names, loaded addresses will be displayed on screen for reference as follows:

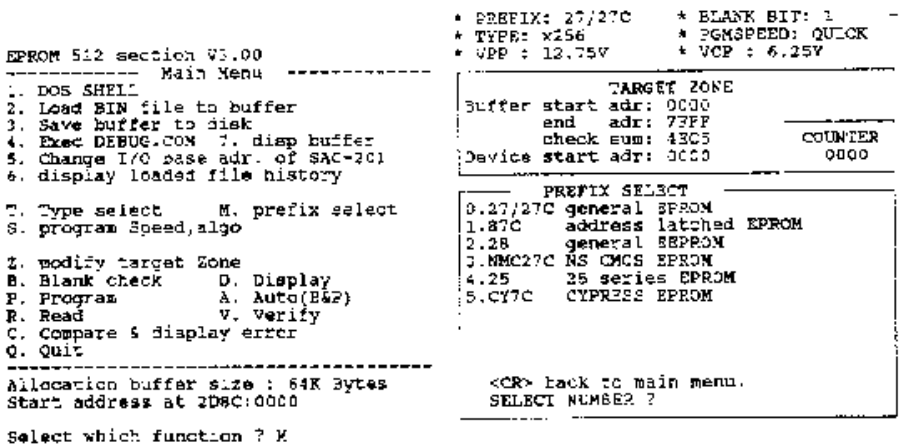
```
Start   End          File name
0000---336E : EPP512.OBJ
0000---3EFA : EPP512E.C
0000---336E : EPP512.OBJ
0000---0000 :
0000---0000 :
0000---0000 :
0000---0000 :
0000---0000 :
0000---000C :
0000---000C :
0000---000C :
0000---000C :
0000---000C :
0000---0000 :
0000---0000 :
0000---0000 :
0000---0000 :
0000---0000 :
0000---0000 :
0000---0000 :

   Press any key to continue ....
```

List file information

Press **<ESC>** to return to the main menu.

4.8. Manufacturer, prefix select

Press **M**, the sub-menu will be displayed as shown.



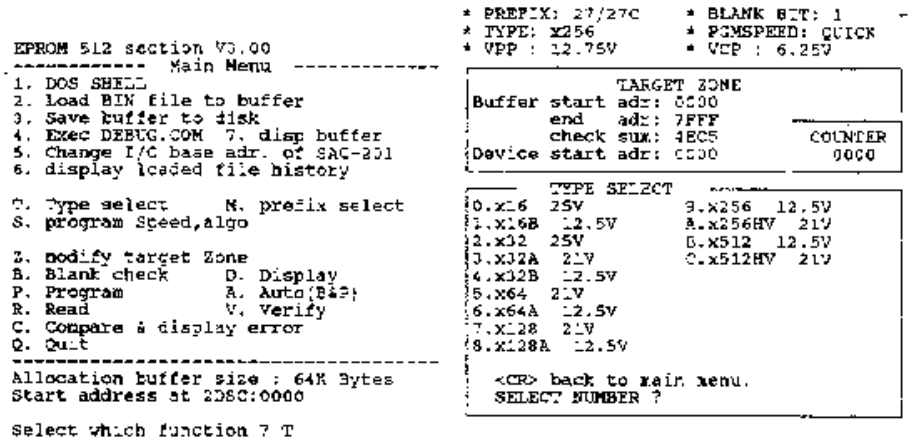
Sub-menu of Mfr

Select the manufacturer by typing in the number which corresponds to your EPROM. After selecting the manufacturer, the new manufacturer will be updated in the **STATUS FIELD** for programming reference.

Press **<ESC>**, or **<CR>** to return to main menu.

4.9. Type select

Press **T** , the sub-menu will be displayed as shown.



Sub-menu of Type

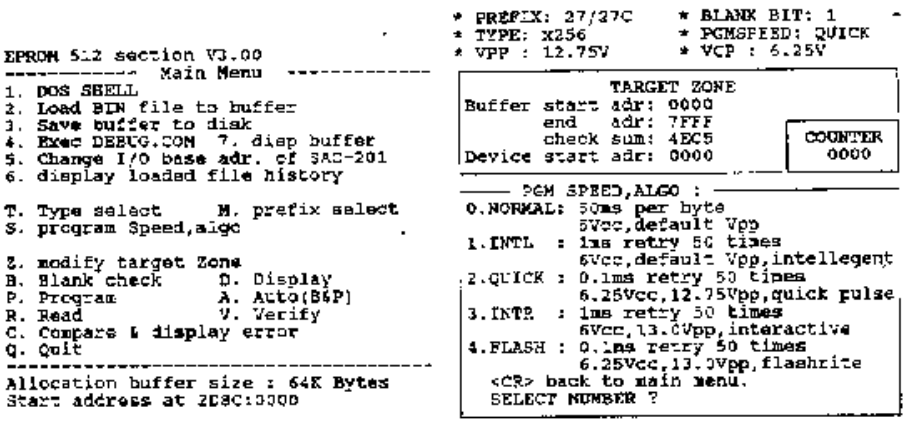
Type in the number or letter that corresponds to the type of your EPROM. The type and voltages will automatically be selected and displayed in the **STATUS FIELD** for programming reference.
Press **<ESC>**, or **<CR>** to return to main menu.

NOTE

EPROM chips are not always clearly labelled; it may be difficult, for example, to determine whether the EPROM you wish to program is a 2732 requiring 25V, or whether it is a 2732A, which requires 21V. If you encounter this problem, try to program the EPROM at the lower voltage first; if this does not work, then erase the EPROM and try the higher EPROM voltage.

4.10. Program speed,algo

Press **S** to select speed function.
A dialog window will be displayed on screen as follows:



Speed select

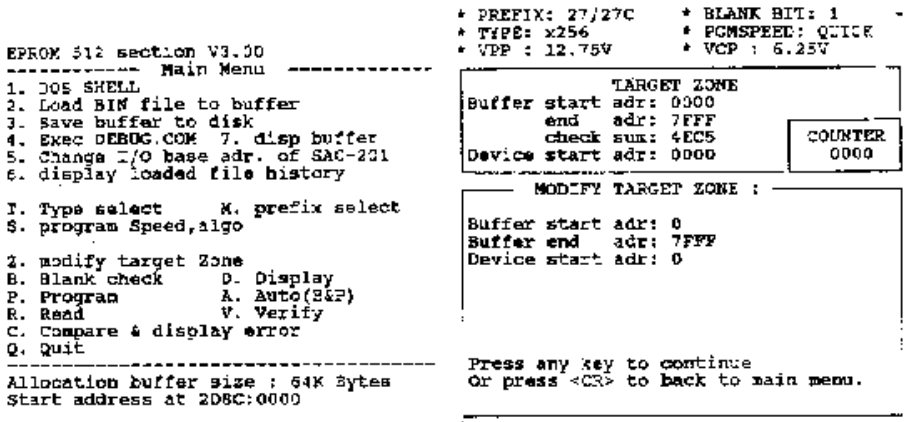
When prompted, type in the number of new programming speed you wish to work with.

The speed you have selected will be updated in the **STATUS FIELD** for programming reference.

Press **<ESC>** or **<CR>** to return to the main menu.

4.11. Modify buffer(target zone)

Press **Z** to select modify buffer function.
A dialog window will be displayed on screen as follows:



Modify buffer

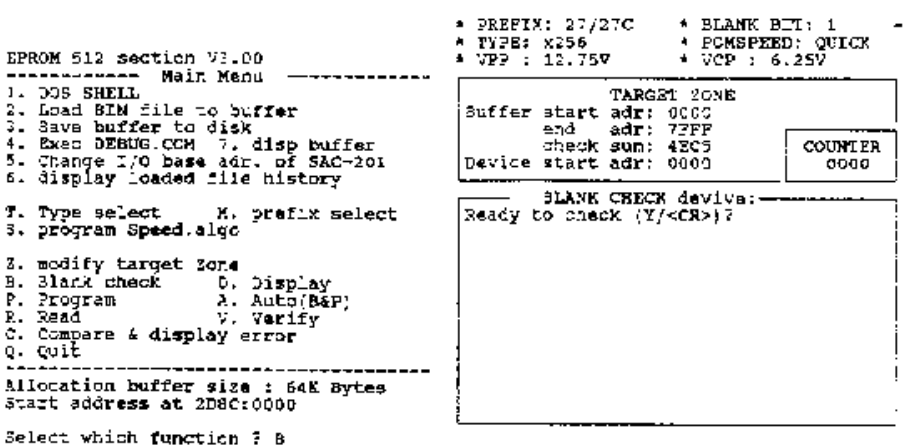
When prompted,type in the new buffer start address and press **<CR>**. Next, type in the new buffer end address and press **<CR>**. Next, type in the new EPROM start address and press **<CR>**.

The values you have typed(in HEX) will be updated in the **STATUS FIELD** for programming reference.

Press any key to return to the main menu.

4.12. Blank

Press **B** , the sub-menu will be displayed as shown.

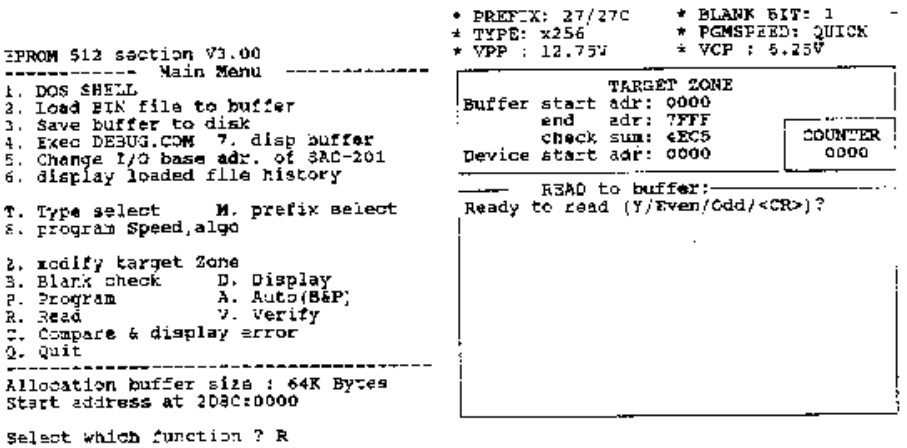


Sub-menu of Blank

Press **Y** to start the blank check, or **<CR>** to return to the main menu. If the chip fails the test, the first address that is not blank will be displayed. Otherwise, the **Blank check OK !** message will be displayed.

4.13. Read

Press **R**, the sub-menu will be displayed as shown.



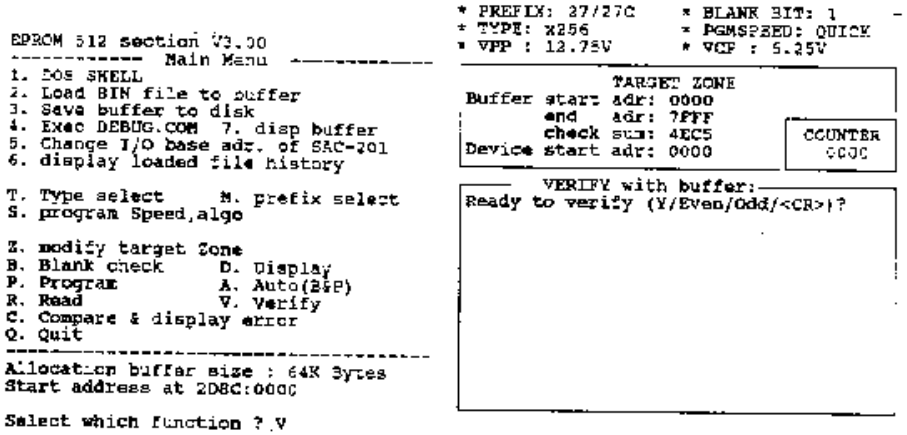
Sub-menu of Read

Press **Y** to read data from the EPROM to the buffer shown in **STATUS FIELD**, or press **<CR>** to return to main menu. The sub-menu will display the **Reading now ...** message, and when the reading is completed, the **Read Ok !** message will be displayed.

The check sum will be automatically calculated after device reading.

4.14. Verify

Press **V** , the sub-menu will be displayed as shown.



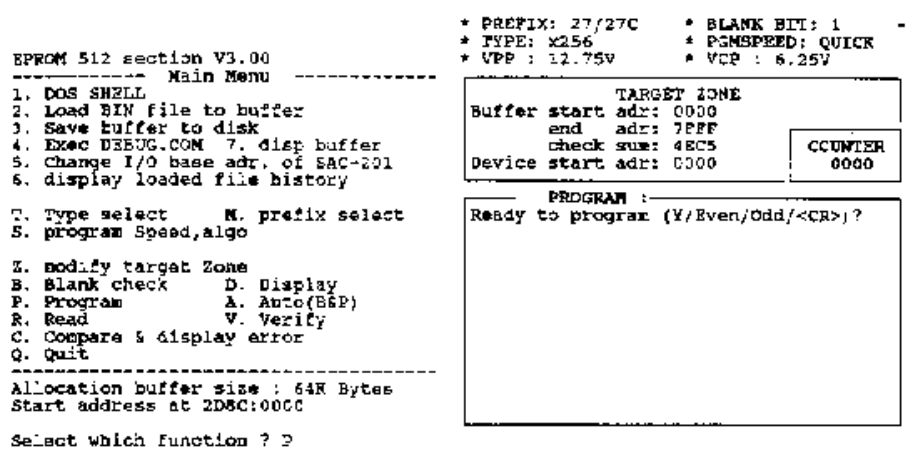
Sub-menu of Verify

Press **Y** to verify the EPROM with buffer shown in **STATUS FIELD**, or press **<CR>** to return to main menu. The sub-menu will display the **Verifying now** message. When the verifying is complete, the **Verify Ok !** message will be displayed on the screen.

The verifying routine will be terminated if no error is found.

4.15. Program

Press **P** , the sub-menu will be displayed as shown.



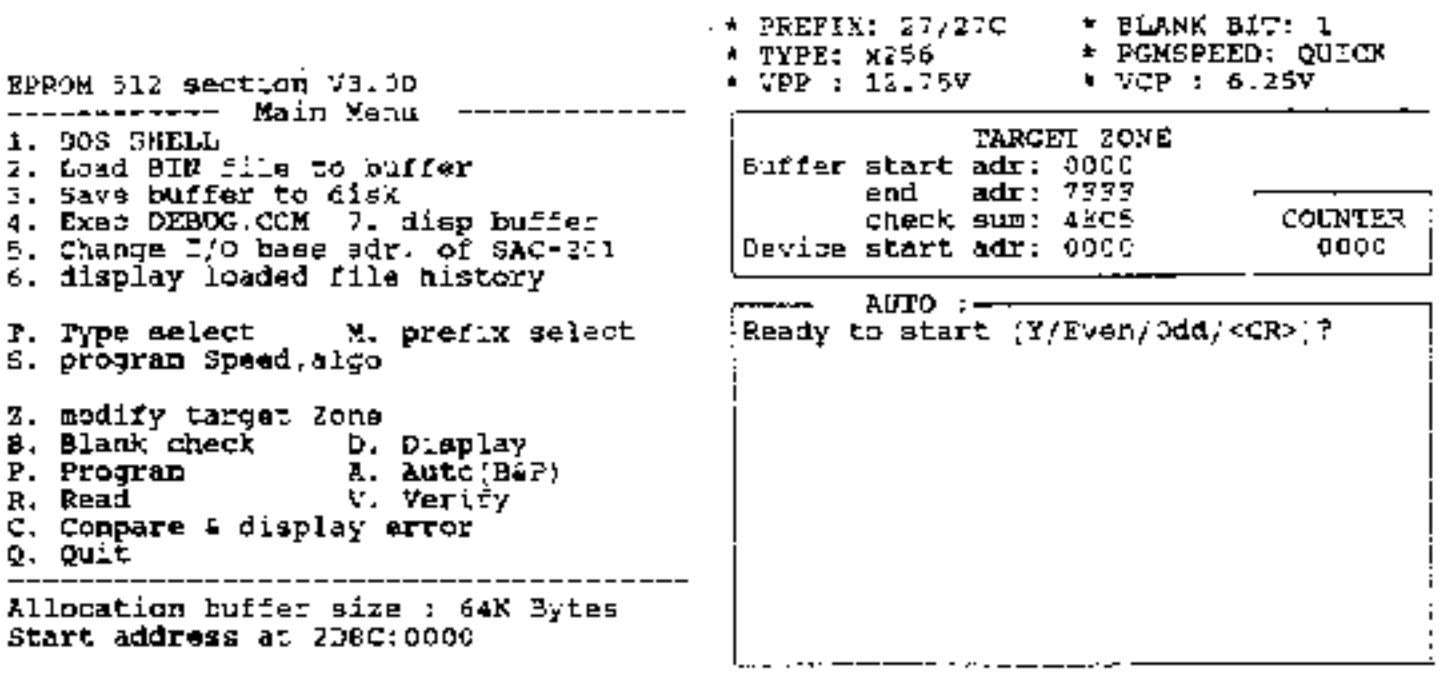
Sub-menu of program

Now select **Y** to begin transferring program data from the memory buffer to the EPROM, or press **<CR>** to return to main menu. The programming message will appear on the screen as well as scrolling numbers on the **COUNTER** in the **STATUS FIELD**. When the transfer is completed, **Program OK !** message will be displayed on the screen.

Auto verify program data after program completed.

4.16. Auto (B & P)

Press **A** , the sub-menu will be displayed as shown.

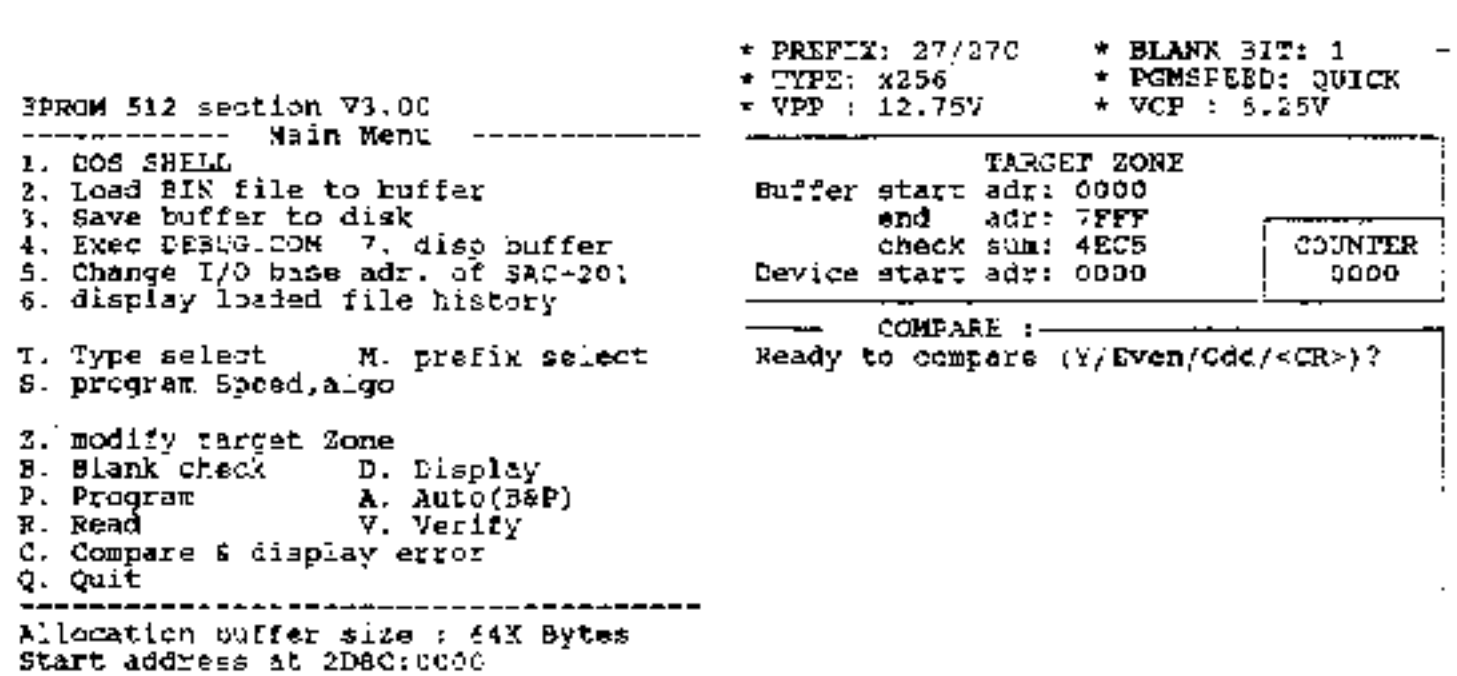


Sub-menu of Auto

You can then press **Y** to start **Auto** function, or **<CR>** to return to main menu. This function is almose the same as **Program** function except that the blank check function is automatically performed before programming.

4.17. Compare & display error

Press **C** , the sub-menu will be displayed as shown.



Sub-menu of Compare

NOTE

The programmer will compare the buffer shown in STATUS FIELD with the chip in the socket .
Press **Y** if you wish to compare data in chip with the memory buffer. Otherwise, press **<CR>** to return to the main menu. The screen will display the differences in the following format:

Error at:
Press <ESC> to terminate display.

```
00000:DA - (00000:21) , 00001:1F - (00001:05) ,
```

Press any key to continue.

While the differences are being displayed, you can press the **Ctrl** and the **S** keys simultaneously to hold the display. Press **<ESC>** to terminate the display, then press any key to return to the main menu. If no errors are found, the **Compare OK !** message will be displayed.

4.18. Display

Press **D** , the sub-menu will be displayed as shown.

```
Press <ESC> to terminate display

0000  80 CA 00 08 65 70 70 35 --31 32 2E 43 20 85 07 00 ....app512.C ...
0010  00 00 4D 53 20 43 6E 88 --05 00 00 9F 45 4D 42 38 ..MS Cn.....EMB.
0020  09 C0 00 9F 53 4C 49 42 --46 50 10 88 08 C0 00 9F ....SLIBFP......
0030  53 4C 49 42 43 54 88 07 --00 00 9F 4C 49 42 48 B3 SLIBCt.....LIBH.
0040  88 06 00 00 9D 39 72 4F --E3 88 06 00 00 A1 01 43 .....9sO.......C
0050  56 37 9E 2E 00 00 06 44 --47 52 4F 55 50 05 5F 54 V7.....DGROUP._T
0060  45 58 54 04 43 4F 44 45 --05 5F 44 41 54 41 04 44 EXT.CODE._DATA.D
0070  41 54 41 05 43 4F 4E 53 --54 04 5F 42 53 53 03 42 ATA.CONST._BSS.B
0080  53 53 3F 98 07 00 28 26 --17 03 C4 01 F4 98 07 00 SS?...(&........
0090  48 96 0E 05 26 01 6C 98 --07 00 48 00 00 07 07 01 H.....l...H.....
00A0  0A 98 07 00 48 00 00 08 --09 01 07 9A 08 00 02 3F ....H..........
00B0  03 FF 04 FF 02 56 9C 0D --00 00 03 01 02 02 01 C3 .....V..........
00C0  04 4C 01 45 01 C0 8C CC --01 0A 5F 5F 61 63 72 74 .L.E...... acrt
00D0  75 73 65 64 00 07 5F 68 --61 6C 6C 6F 63 00 07 5F used.._halloc.._
00E0  63 68 6B 63 75 72 00 08 --5F 69 6F 5F 6D 61 6E 75 chkcur.._io_manu
00F0  00 06 5F 63 68 6B 69 64 --00 0A 5F 68 69 73 74 5F .._chkid.._hist_
0100  6D 61 6E 75 00 0C 5F 64 --73 70 62 75 66 5F 6D 61 manu.._dspbuf_ma
0110  6E 75 00 08 5F 73 70 65 --65 64 5F 6D 61 6E 75 00 nu.._speed_manu.
0120  09 5F 6D 66 67 5F 6D 61 --6E 75 00 06 5F 67 65 74 ._mfg_manu._get
0130  63 68 00 09 5F 74 79 70 --5F 6D 61 6E 75 00 0D 5F ch.._typ_manu.._
Press any key to continue
```

Sub-menu of Display

The screen will display the address and the current contents of that address in the EPROM. Press **<ESC>** to exit display, and press any key to return to the main menu.

4.19. Quit

Press **Q** to exit the main menu and return to DOS. If you are in one of the sub-menus, you must first exit that sub-menu and return to the main menu; then press **Q** to exit.

Before exiting to DOS, software will save parameters of manufacturer, type, I/O address and programming speed in the parameter file ????????**.DAT** for later use.

4.20. External key and LEDs

There is only one external key on the main module. It is **"YES"** key.
The function of this key is the same as Y on PC keyboard. When sub-menu ask you to enter Y, you may type this key to replace Y on PC keyboard.

There are 3 LEDs on the main module. Those are **ON LED, BUSYLED and GOOD LED.**

The ON LED will be turned on after switching on the computer power. The BUSY LED will be turned on when performing device programming,reading and verifying. The GOOD LED will be turned on when the verifying result is good.

FOR PLD

The functions on PLD programming are almost the same as on the EPROM. This section will describe the extra functions only.

Extra functions of PLD programming

Security fuses blow

Press S to blow the security fuses. This is the final step in PLD programming. By blowing the security fuses, you prevent any further access to the PLD either for modification or for "reading". Blowing the fuses prevents anyone from making unauthorized copies of you PLD.

5. USING THE UTILITY FILES

5.1. HEX to BINARY code converter

The converter can convert HEX format ASCII file to RAW BINARY file. Some ASSEMBLERs or COMPILERs produce HEX format file from user source program, then can transmit it to stand alone programmer or ICE through RS-232C hardware interface.

This programmer is not a stand alone. We use direct I/O control technic to control it by the CPU on PC, and also share the large memory resources on PC. This programmer software will load any type of file without converting anything during file loading.
It loads file content byte by byte continuously into software buffer for programming use.

This HEX converter will convert your HEX format file to an executable ROM code that is meaningful to your target CPU.

We assume that user understands the difference between HEX format file and ROM code BINARY file.

There are 3 types of HEX format file that can be converted to BINary file by HEXBIN.EXE

1. INTEL HEX format
2. MOTOROLA S HEX format
3. TEKTRONICS HEX format (seldom used)

Max. converted size is 64 K bytes

HEXBIN2.EXE can convert INTEL, MOTOROLA extended hex format file.Maximum conversion size is 256k bytes.

The starting address may be specified on HEXBIN2, and the leading garbage will be skipped out to maintain a small size of output binary file.

The input command under DOS command prompt is :

```
A>HEXBIN2  [HEX  FILE NAME] [BIN FILE  NAME]
           [HEX  FORMAT] [start address] <CR>
```

[] : Option <CR> : return key or enter key.

HEX FILE NAME and BIN FILE NAME : standard file name that is specified by DOS.

HEX FORMAT : I for INTEL HEX
M for MOTOROLA S HEX
T for TEKTRONICS HEX

start address: HEXADECIMAL digit, need not assign this parameter in HEXBIN converter.

1. In INTEL extended HEX format, the start address represents the start segment address.
   All data lying between start segment x 16 and next 256 K bytes will be picked up. Segment ange is 0 to FFFF.

2. In MOTOROLA S HEX format, the start address represents the actual start address.
   All data lying between start address and next 256 K bytes will be picked up. Address range is 0 to FFFFFFFF.

for ex :

```
A>HEXBIN2 DEMO.HEX DEMO.TSK I 1000 <CR>
```

HEXBIN2 will convert HEX file DEMO.HEX to BINARY file DEMO.TSK, using INTEL HEX converting technic, only data lying between 10000H and 50000H needs to be converted.

This program also can be used in prompt mode as follows :

```
A> HEXBIN2 <CR>

   HEX FILE NAME :DEMO.HEX
   BIN FILE NAME :DEMO.BIN
   HEX FORMAT (<I>INTEL <M>MOTOROLA <T>TEKTRONICS) : I
   SEGMENT ADDRESS : 1000
```

5.2. Dump BINARY file to console

Most of any ROM code BINARY file can't be displayed on screen by DOS **TYPE** command. DUMP.EXE can convert BINARY file to HEXADECIMAL character and display them on the console or printer. Although it is meaningless , a designer may need to keep a copy of paper listing for later use.

The input command under DOS command prompt is :

```
A>[^P] DUMP FILENAME [start address] <CR>

  [] : Option    <CR> : Return key or Enter key.
  [^P] : Ctrl + P
```

It will connect PC with printer, the data displayed on screen will output to printer.

FILENAME : standard file name that is specified by DOS.

start address : HEXADECIMAL digit
start dumping from this address, range from 0 to FFFFF.

5.3. 2 ways , 4 ways BINARY file splitter

SPLIT2.EXE can split a 16 bits source file into two 8 bits files. One is the collection of data lying on LOW byte of 16 bits file. The other is the collection of data lying on HI byte of source file. This utility can be used to split 16 bits file into two 8 bits files that can be programmed to an EVEN EPROM and an ODD EPROM.

SPLIT4.EXE can split a 32 bits source file into four 8 bits files. 1st file is the collection of data lying on 1st byte of 32 bits file. 2nd file is the collection of data lying on 2nd byte of 32 bits file. 3rd file is the collection of data lying on 3rd byte of 32 bits file. 4th file is the collection of data lying on 4th byte of 32 bits file.

The input command under DOS command prompt is :

```
A>SPLIT2 [input file][output EVEN file][output ODD
         file] <CR>
A>SPLIT4 [input file][output 1st file][output 2nd
         file][output 3rd file][output 4th file]
         <CR>
```

[] : Option <CR> : Return key or Enter key

input file,output EVEN file,output ODD file,output 1st file, output 2nd file, output 3rd file, output 4 th file : standard file name that is specified by DOS.

5.4. 2 ways , 4 ways BINARY file shuffler

SHUFF2.EXE can shuffle two 8 bits source files into a 16 bits file. The 1st 8 bit file will be collected to LOW byte of 16 bits file, the 2nd 8 bits file will be collected to HI byte of 16 bits file.

SHUFF4.EXE can shuffle four 8 bits source files into a 32 bit files. 1st 8 bits file will be collected to the 1st byte of 32 bits file. 2nd 8 bits file will be collected to 2nd byte of 32 bits files. 3rd 8 bits file will be collected to 3rd byte of 32 bits file. 4th 8 bits file will be collected to 4th byte of 32 bits file.

The input command under DOS command prompt is :

```
A>SHUFF2 [output file][input EVEN file][input ODD
         file] <CR>
A>SHUFF4 [output file][input 1st file][input 2nd
         file][input 3rd file][input 4th file]
         <CR>
```

[] : Option <CR> : Return key or Enter key

output file,input EVEN file,input ODD file, input 1st file, input 2nd file, input 3rd file, input 4th file : standard file name that is specified by DOS.

APPENDIX A. TROUBLE SHOOTING

We have provided the following troubleshooting guide to help you overcome some of the more commonly experienced problems. This guide, however, is not intended to be a repair manual. If you encounter problems other than those described here,please contact your dealer or our Sales Department.

INSTALLATION PROBLEMS

PROBLEM:

When the computer is turned on, the power supply fan does not turn and nothing appears on the monitor.

SOLUTION:

Check the power cord. One end of the cord must be securely inserted into a power outlet and the other end must be pressed firmly into the power supply. If you have a surge protector, make sure that it is switched on.

PROBLEM:

When the computer is turned on, the power supply makes a clicking sound.

SOLUTION:

The programmer card may be improperly installed or may have a short circuit. Turn the computer off and push the card all the way into the slot to make sure it is firmly seated. Turn on the computer. If the problem persists, switch the computer off, remove the programmer card from the motherboard and turn on the computer again. If there is no longer any clicking sound when the card is removed. there may be a short either in the SAC-201 card or in the power supply. Contact your dealer or our Sales Department for further assistance.

PROBLEM:

When the computer boots up or attempts to load one of the programmer operating programs, the screen displays the following message:

> **Error identification on hardware !**
> **Press <Q> to Quit**
> **or press <CR> to continue...**

SOLUTION:

Make sure that the card is fully seated in its slot. Check that cables are plugged in correctly. Also you should be aware that the programmer software will not run on programmers made by other manufacturers.

PROBLEM:

When the computer attempts to exit a programmer operating program, the following error message is displayed:

> **WRITE PROTECT ERROR WRITING DRIVE A**
> **Abort,Retry,Fail?**

SOLUTION:

You must not use the original utility diskette. The original diskette is write-protected and will not allow the software pre-exit disk write activity. Make a work copy of the original and use the work copy.

GENERAL TROUBLESHOOTING CHECKLIST

If your problem is not described above, check the following:

1. Is the SAC-201 adapter card fully seated in its slot ?
2. Are all cable connections securely fastened ?
3. Does the SAC-201 jumper setting match the I/O address displayed by the programmer software ?
4. Does any other card on the bus have the same I/O address as the adapter card ?

APPENDIX B. DEVICES SUPPORTED ON EACH SOFTWARE

The devices that can be programmed by this programmer are listed with their own software below. If there are devices that are not included here, please contact us or local distributor for help.

#1. diskette

EPP512.EXE (Under 512 K Bits EPROM)

Please ref. to APPENDIX B. for detailed description.

* General EPROM : 27/27C16 to 27/27C512.
* General EEPROM : 2816,16A,17A,64A,256A.
* NS CMOS EPROM : 27C16 to 27C512,27CP128.
* General EPROM : 2516 to 2564.
* Address latched EPROM : 8764 to 87256.
* CYPRESS PROM TYPE EPROM : 7C251,253,254,271, 261,263,264(Read only).

EPP1024.EXE (From 1 M to 4 M Bits EPROM)

---

* Byte Wide EPROM : 27C010,020,040.
  HITACHI 27C101, NS 27C1023
  NEC or FUJITSUU 27C1001.
  TOSHIBA 571000,2000,4000.
  OKI 27C1000.
* Word Wide EPROM : 27C210, 27C1024.
* Byte Wide Mask ROM PIN-OUT EPROM :
  HITACHI 27C301.
  NEC or FUJITSUU 27C1000.
  TOSHIBA 571001.
* Page Mode EPROM : 27513,27011.

BPPGM.EXE (NON-Registered PROM)

---

* MMI : 63S-080,081,140,141,240,241,440, 441,840,841,280,281,480,481,

1640,1641,485,881,1681,3281.

* NS : 74S/LS-188,288,287,387,570,571, 572,573,184,185,471,472, 473,195,474,475,180,181, 280,281,190,191,290,291, 321,421.
* SIGNETICS: 82S-23,123,126,129,130,131,136, 137,184,185,135,146,47,195, 140,141,180,181,182,183,2708, 190,191,321.
* TI : 18S030,24S,28S-10,41,81,22,42,45, 46,85,86,2708,165,166.
* MOTOROLA : 76-20,21,42,43,84,85,49,40,41,80, 81.
* HARRIS : 76-02,02,10,11,20,21,42,43,85,49, 165,40,41,81,161,321
* AMD : 27S-18,19,12,13,32,33,184,185,28, 29,41,31,180,181,280,281,190, 191,290,291,43,49,49A.
* CYPRESS : CY7C-281,282,291,292,27C291,27C292. CY7C-261,263,264(read only).

PGM48.EXE (8748 series MPU)

---

* INTEL : 8748,48H,49H,50H,41A,42H,48AH,49AH, 41AH,42AH.
* Other Manufacturer : NEC, FUJITSU,MITSUBISHI

PGM51.EXE (8751 series MPU)

---

* INTEL : 8751,51H,51BH,C51,52H,52BH,C252,44H.
* Other Manufacturer : AMD,SIGNECTICS.

PGMZ8.EXE (Z8 series MPU)

---

* SGS : 86E11,86E21.
* ZILOG : 86E11,86E21,86E21 OTP.

#2. diskette

PALP1.EXE (1 Generation PLD)

---

* MMI : 10H8/-2,10L8/-2,12H6/-2,12L6/-2,
14H4/-2,14L4/-2,16H2/-2,16L2/-2,
16C1/-2,16X4,
16L8/B-2/B-4/A/A-2/A-4,
16R8/B-2/B-4/A/A-2/A-4,
16R6/B-2/B-4/A/A-2/A-4,
16R4/B-2/B-4/A/A-2/A-4,
16L8D/B,16R8D/B,16R6D/B,16R4D/B,
C16L8Q-25,C16R8Q-25,C16R6Q-25,
C16R4Q-25,
12L10,14L8,16L6,18L4,20L2,20C1,
20L10A,20X10A,20X8A,20X4A,
20L8B/B-2/A/A-2/D,
20R8B/B-2/A/A-2/D,
20R6B/B-2/A/A-2/D,
20R4B/B-2/A/A-2/D.
* NS : A or B TYPE
10H8,10L8,12H6,12L6,14H4,14L4,
16H2,16L2,16C1,
16L8,16R8,16R6,16R4,
12L10,14L8,16L6,18L4,20L2,20C1,
20L10,20X10,20X8,20X4,
20L8,20R8,20R6,20R4.
* TI : 16L8-12/-15/-25,16R8-12/-15/-25,
16R6-12/-15/-25,16R4-12/-15/-25,
16L8-10,16R8-10,16R6-10,16R4-10,
16L8-7/H-15,R4-7/H-15,
R6-7/H-15,R8-7/H-15,
20L8-15/-25,20R8-15/-25,
20R6-15/-25,20R4-15/-25,
20L8-7,20R8-7,20R6-7,20R4-7,
20L10,20X10,20X8,20X4.
* AMD : 16L8/B/AL/A/Q/L,16R8/B/AL/A/Q/L,
16R6/B/AL/A/Q/L,16R4/B/AL/A/Q/L,
20L10B/-20/AL.
* CYPRESS : PALC16L8,C16R8,C16R6,C16R4.
* SIGNETICS : PLUS16L8,PLUS20L8.

PALP2.EXE (Polarity PLD)

---

* MMI : 16P8,16RP8,16RP6,16RP4,16RA8,20RA10

PALP3.EXE (UV Erasable PLD)

---

* CYPRESS : PLDC20G10, PALC22V10.
* MMI,SAMSUNG : PALC22V10-25/35.
* TI : TICPAL22V10.

PALP4.EXE (UV Erasable PLD)

---

* SIGNETICS : PLC16V8,PLC20V8,
PLC18V8Z,PLC22V8Z.
* TI : TICPAL18V8,
TIBPAL22V10/A/-15,
TIBPAL22VP10-20/-25.

PALP5.EXE (UV Erasable PLD)

---

* AMD : 22V10-15/-20,
* TI : 22V10/22V10L.

FPLP1.EXE (FPL)

---

* AMD : AmPAL18P8AL.
* SIGNETICS : PLHS18P8A,PLHS16L8A/B.
PLS153/A,PLS173,PLUS153,PLUS173.

#3. diskette

EPLD1.EXE (EPLD)

---

* ALTERA : EP310,EP320,EP600,EP900.
* TI : EP610,EP910.

GAL1.EXE (GAL)

---

* LATTICE : GAL16V8,GAL20V8,
  GAL16V8A-25/-35,
  GAL16V8A-10/-12/-15,
  GAL20V8A-25/-35,
  GAL20V8A-10/-12/-15.
* Other Manufacturer : NS,VLSI,SGS.

PEEL1.EXE (PEEL)

---

* ICT : PEEL153,173,253,273,
  PEEL18CV8,22CV10.
* Other Manufacturer : HYUNDAI,AMI,GOULD.

APPENDIX C. EPROM MANUFACTURER,TYPE,VPP

MFR.	TYPE	VPP
AMD	2716	25V
	2732	25V
	2732A	21V
	2732B	12.5V
	2764	21V
	2764A	12.5V
	27128	21V
	27128A	12.5V
	27256	12.5V
	27C256	12.5V
	27512	12.5V
	27C512	12.5V
	2817A	Vcc
	2864A/B	Vcc
ATMEL	27HC64	12.5V
	27C128	12.5V
	27256	12.5V
	27C256	12.5V
	27HC256	12.5V
	27C512	12.5V
CATALYST	2764A	12.5V
	27128A	12.5V
	27256	12.5V
	27512	12.5V
FUJITSU	8516	25V
	2732	25V
	2732A	21V
	2764	21V
	27C64	21V
	27128	21V
	27C128	21V
	27256	12.5V
	27C256	21V
	27C256A	12.5V

	27C256H	12.5V
	27C512	12.5V
HITACHI	462716	25V
	462732	25V
	482732A	21V
	27C64	21V
	482764	21V
	27128A	12.5V
	4827128	21V
	27256	12.5V
	27C256	12.5V
	27512	12.5V
INTEL	2716	25V
	2816A	Vcc
	2817A	Vcc
	2732	25V
	2732A	21V
	2732B	12.5V
	2764	21V
	2764A	12.5V
	27C64	12.5V
	87C64	12.5V
	P2764A	12.5V
	27128	21V
	27128A	12.5V
	27128B	12.5V
	27C128	12.5V
	P27128A	12.5V
	27256	12.5V
	27C256	12.5V
	P27256	12.5V
	27512	12.5V
MICROCHIP	27C64	12.5V
	27HC64	12.5V
	27C128	12.5V
	27256	12.5V
	27C256	12.5V
	27HC256	12.5V

	27C512	12.5V
MITSUBISHI	2716	25V
	2732	25V
	2732A	21V
	2764	21V
	27128	21V
	27C128	21V
	27256	12.5V
	27C256	12.5V
	27512	12.5V
	27C512A	12.5V
NS	2716	25V
	27C16	25V
	27C16H	25V
	27C32	25V
	27C32H	25V
	27C16B	12.5V
	27C32B	12.5V
	27C64	12.5V
	27C128	12.5V
	27CP128	12.5V
	27C256	12.5V
	27C256B	12.5V
	27C512	12.5V
	27C512A	12.5V
	9816A	Vcc
	9817A	Vcc
	98C64	Vcc
NEC	2716	25V
	2732	25V
	2732A	21V
	2764	21V
	27128	21V
	27256	21V
	27256A	12.5V
	27C512	12.5V
OKI	2716	25V

	2732	25V
	2732A	21V
	2764	21V
	2764A	12.5V
	27128	21V
	27128A	12.5V
	27256	12.5V
	27512	12.5V
	2816A	Vcc
RICOH	27C32	21V
	27C64	21V
	27C256	12.5V
SEEQ	2816AH	Vcc
	5516AH	Vcc
	2817A(H)	Vcc
	5517A(H)	Vcc
	2764	21V
	27128	21V
	27C256	12.5V
SHARP	5762	12.5V
	5763	12.5V
	5764	12.5V
	57126	12.5V
	57127	12.5V
	57128	12.5V
	57256	12.5V
SIGNETICS	27C64	21V
	27C64A	12.5V
	27C64AF	12.5V
	27C256	12.5V
	27C256F	12.5V
	27C512	12.5V
S-MOS	27C64H	21V
	27128H	21V
	27C256H	12.5V

SGS	2716	25V
	27C16	25V
	2732	25V
	2732A	21V
	27C32	25V
	2764	21V
	2764A	12.5V
	27C64	12.5V
	27128A	12.5V
	27256	12.5V
	27C256	12.5V
	27512	12.5V
TI	2732	25V
	27(P)32A	21V
	27(P)64	21V
	27C64	12.5V
	27128	21V
	27C128	12.5V
	27PC128	12.5V
	27256	12.5V
	27C256	12.5V
	27C512	12.5V
	27PC512	12.5V
TOSHIBA	2732D	25V
	2732A	21V
	2464	21V
	2464A	12.5V
	2764D	21V
	2764A	12.5V
	24128	21V
	24128A	12.5V
	27128D	21V
	27128A	12.5V
	24256	12.5V
	24256A	12.5V
	27256D	21V
	27256A	12.5V
	27256B	12.5V
	54256	21V

	54256A	12.5V
	57256	21V
	57256A	12.5V
	24512	12.5V
	27512	12.5V
	27512A	12.5V
	57512A	12.5V
VLSI	27C64	12.5V
	27C128	12.5V
	27C256	12.5V
	27C512	12.5V
	28H64	Vcc

APPENDIX D. IC TESTER function description

1. SPECIFICATIONS

1.1 Applicable IC types.

74/54 TTL,HC,HCT or equivalent CMOS series
40/14C CMOS series
45/14C CMOS series
SRAM 6116,5264,62256 series
DRAM 4164,41256,411000,4416,4464,44256

1.2 Test Options

FUNCTION TEST
LOOP-TEST
AUTO-SEARCE NUMBER
USER DEFINED VECTOR TESTS

1.3 Contents

This manual
Software disk
Interface card
Main module

1.4 Minimum computer system requirements.

IBM PC/XT/AT or compatible system
256k memory
DOS V2.0 or greater.

1.5 Files included on the software disk.

ICTEST.EXE : Main program file.
ICTEST.DAT : Setup and parameter data file.
README.DOC : Any revisions or ammendments to the IC Test card or software since the printing of this manual are listed in this file.
TTL74.LIB : TTL Library.
CMOS40.LIB : CMOS 40 Library.

CMOS45.LIB : CMOS 45 Library.
7406.VEC : Sample vector file.
4040.VEC : Sample vector file.

2. INSTALLATION & BASIC OPERATION

2.1 Turn off the power to the computer.

2.2 Insert the interface card into any slot in the computer and connect a cable to the rear of the card.

2.3 Turn on the computer and boot with your DOS diskette.

2.4 Insert the software diskette into default drive and type : ICTEST (ENTER)

The ICT TEST menu will be displayed. Proceed with testing by selecting the desired function number.
A detailed description of each function is described in SECTION 4.

NOTE: BE SURE TO BACKUP YOUR SOFTWARE DISKETTE & STORE IT IN A SAFE PLACE - DO NOT WRITE PROTECT THE WORKING COPY.

3. CARD I/O BASE ADDRESS SELECTION

The interface card's I/O base adddress has been preset at the factory to 2E0H.
If this I/O BASE ADDRESS conflicts with any other interface card you have in your computer, it will be necessary to change the SAC-201 I/O address on both the card AND the software as follows:

3.1 Turn the power off.

3.2 Set the DIP SWITCH to one of the available addresses.

DIPSW1	I/O BASE	DIPSW2	I/O BASE
1	200H	1	280H
2	210H	2	290H
3	220H	3	2A0H
4	230H	4	2B0H
5	240H	5	2C0H
6	250H	6	2D0H
7	260H	7	2E0H(def.)
8	270H	8	2F0H

3.3 Turn on power.

3.4 Type ICTEST (ENTER)

3.5 Select function 2

3.6 Enter the new I/O BASE ADDRESS.

3.7 This new ADDRESS will be saved automatically in the ICTEST.DAT

FILE when you QUIT to DOS. It will therefore not be necessary to redefine the ADDRESS next time you use the IC TEST card.

4. FUNCTION DESCRIPTIONS

To select a particular function press the capitalised character of that function.

1. DIR:

Displays the file directory on the selected drive.
Same as the DOS DIR command.
Press ESC to return to MENU.

2. Change I/O BASE ADDRESS (REF: SECTION 3)

T. IC Type selection:

There are 3 IC types which may be selected, TTL74, CMOS40 and CMOS45. Press the desired number and the corresponding library will be selected for testing.

N. IC Number specification:

Enter the IC number to be tested. eg. 174 (ENTER) (DO NOT ENTER IC TYPE, AS THIS IS SELECTED BY FUNCTION "T". The number entered will be checked to see if it is in the standard library. If the number is not available it is necessary to write your own test vectors. See function "U" below.

F. Function test :

This is a single loop function test i.e. the possible logic combinations are carried out once only.

L. Loop test :

This option will repeatedly test the chip logic combinations in an endless loop. In the event of an error during any of the tests, testing will stop & an error message will be shown. The test may be terminated by pressing any button.

S. Search unknown IC identification number :

The unknown IC will be compared with the currently selected standard library. If it is not found, a message to that effect will be shown. As the parameters of an unknown chip may be contained in any of the 3 libraries, it may be necessary to search all three in turn. If a library IC, or

IC's, meeting the the unknown IC's logic is found, then the IC number or numbers will be displayed.

U. User defined test vector :

This function allows any logic input, expected output, 5V and GND to be defined on the 24 pins of the test socket box.

You may write your own vectors to test any sequence of logic steps for new IC's, (including PAL, FPLA, PROM ETC.) The test vector syntax is described in detail in section 5 (NOTE:Only logic testing is carried out. Loading and IC speed tests are not included)

A. SRAM test program :

Pressing A will produce the SRAM test Menu.

N. SRAM number selection.

Allows selection of the following chips. 6116,6264,62256.

F. SRAM function test.

All bits on the SRAM will be written and read back for comparison.

Q. QUIT.

QUIT to DOS. The current I/O BASE ADDRESS and the current IC Type will automatically be saved to the ICTEST.DAT file.

5. USER DEFINED TEST VECTORS.

5.1 A test vector is the definition of input and output logic states applicable to the various pins of an IC. UP to 1000 vectors may be specified for a single IC thus allowing a sequence of logic events to be tested. There are 24 pins the on textool socket which can be defined as shown.

The syntax of a vector is as follows:

```
                    111111112222
Pin No.     123456789012345678901234

    V0001 XXX01NLHL01GC101HLHL0XLE<CR><LF>
    V0001 specifies vector line number
```

G	on pin 12 applies GND to IC.
E	on pin 24 applies 5 v to IC.
X	on pin 1,2,3,22 don't care input or expected output from IC.
N	on pin 6 applied input and expected output are same as last vector.
0	on pin 4,10,13,15,21 applies LOW to IC.
1	on pin 5,11,14,16 applies HI to IC.
L	on pin 7,9,18,20,23 expected output from IC is LOW.
H	on pin 8,17,19 expected output from IC is HI.

<CR><LF> line termination codes are: 0DH,0AH (i.e. press "ENTER")

Note: If for example a 14 pin IC is to be tested, then pin 6 of the ZIF socket is electrically equivalent to pin one of the IC and so on.

5.2 A vector can be executed by selecting the Function or Loop test. The test sequences are as follows:

1. Apply G and then apply E to IC pins.
2. Apply 0, 1, N, X to pins
3. Read value on each pin and compare with the expected value defined in the vector.
4. If an error occurs, the error is displayed and the test is stopped.
5. If the test is successful procede to the next vector.

5.3 Vector restrictions:

Maximum number of vectors : 1000
0,1,L,H,X,N can be defined on any pin
G can only be defined on pin 12
E can only be defined on pin 19,20,22,24

5.4 The vector can be edited by using the editor built into the software by pressing key "4" under USER TEST VECTOR MENU, The editing key functions will be shown on screen. Other word processors may be used provided they are in an ASCII mode. e.g. Wordstar in the N mode.

5.5 "3" saves the vector to disk.

5.6 "2" Loads the vector into the buffer.

5.7 USER function test and LOOP test.

"F" for vector function testing.

"L" for vector loop testing.

5.8 "D" will invoke the DISPLAY RESULT DURING TESTING function.

This function will display the test results on a single step basis or in continuous mode. If required the results of each step can be output to the printer.

This operates in a manner similar to a logic function debugger.

TTL74.LIB

0000	0001	0002	0003	0004	0005	0006	0007	0008	0009
0010	0011	0012	0013	0014	0015	0016	0017	0020	0021
0022	0025	0026	0027	0028	0030	0032	0033	0037	0038
0040	0042	0043	0045	0046	0047	0048	0050	0051	0052
0053	0054	0055	0060	0070	0072	0074	0085	0086	0095
0107	0108	0109	0110	0111	0112	0113	0114	0116	0125
0126	0128	0132	0133	0136	0137	0138	0139	0145	0147
0148	0150	0151	0152	0153	0154	0155	0156	0157	0158
0159	0160	0161	0162	0163	0164	0165	0166	0168	0170
0173	0174	0175	0180	0183	0189	0190	0191	0192	0193
0194	0195	0240	0241	0242	0243	0244	0245	0247	0248
0249	0251	0253	0257	0258	0259	0260	0266	0273	0276
0279	0280	0283	0290	0293	0295	0298	0299	0322	0323
0352	0353	0365	0366	0367	0368	0373	0374	0375	0377
0378	0386	0390	0393	0465	0540	0541	0573	0574	0590
0640	0641	0643	0644	0645	0670	0688	0804	0805	0870
add:	0490	0646	0648						

CMOS40.LIB

0000	0001	0002	0006	0007	0008	0009	0010	0011	0012
0013	0014	0015	0016	0017	0018	0019	0020	0021	0022
0023	0025	0026	0027	0028	0029	0030	0032	0033	0035
0038	0040	0041	0042	0043	0044	0048	0049	0050	0051
0052	0053	0054	0055	0056	0060	0063	0066	0067	0068
0069	0070	0071	0072	0073	0075	0076	0077	0078	0081
0082	0085	0086	0093	0094	0095	0096	0097	0099	0101
0102	0103	0105	0106	0108	0109	0160	0161	0162	0163
0174	0175	0192	0193	0194					

CMOS45.LIB

0001	0002	0003	0004	0006	0008	0010	0011	0012	0014
0015	0016	0017	0018	0019	0020	0029	0032	0038	0043
0053	0055	0056	0072	0084	0085				

SRAM

6116 6264 62256

DRAM

4164 41256 411000
4416 4464 44256