MODULAR CIRCUIT TECHNOLOGY

MOD-MUP

Owner's Reference Guide

Modular Programming System: Universal Programmer/Tester

Table of Contents

NOTES

1. Introduction

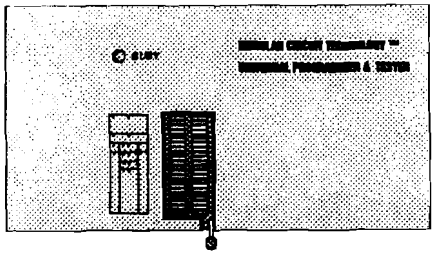
What is the MOD-MUP?

The MOD-MUP is one of several programmers/ testers in the Modular Programming System. It is a "Universal" chip programmer and tester, and can perform the functions of a variety of different programmers:

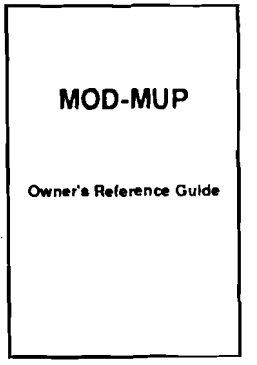
- MOD-MEP EPROM programmer.
- MOD-MPL PAL programmer.
- MOD-MIC IC tester.
- MOD-MBP Bipolar PROM programmer.
- MOD-MMP Microprocessor programmer.
- Several functions not found on the above programmers, such as
  - GAL programming.
  - Self test during power-up.

What's Included

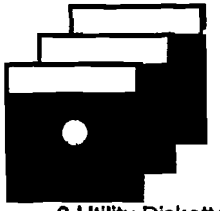
Your MOD-MUP should come with the following:



MOD-MUP



This Manual



3 Utility Diskettes

System Requirements

To use the MOD-MUP with your computer system, the following requirements must be met:

- A system that can support the **MOD-MAC Host Interface Card**. See the MOD-MAC manual.
- A MOD-MAC Host Interface Card.

Manual Overview

The following chapter summaries will help you identify the chapters you need to read. For best results, however, we recommend that you read the entire manual before installing and using your MOD-MUP.

1. **Introduction**

   Product overview & system requirements.

2. **Hardware Installation**

   Step-by-step instructions for installing your MOD-MUP.

3. **Software Installation**

   Step-by-step instructions for installing and using your MOD-MUP software.

4. **Technical Reference**

   Features, connectors, & pin-outs.

5. **Trouble-Shooting**

   A guide to solving possible problems you may have.

6. **Glossary**

   A glossary of computer terminology.

Conventions Used

Certain nomenclature, typographical and iconic conventions will be used throughout this manual. A brief explanation of these follows.

Bold Type usually indicates a heading. If bold type is used outside a heading, it is used to highlight a term of importance. Definitions for these terms can be found in the glossary.

Bold Italic Type is used to show default values or settings.

Bold Roman Type is used to show responses from the computer, or commands to the computer.

Keys on the keyboard are enclosed in "brackets", i.e., <Del> represents the Delete key, <A> represents the capital letter "A", etc. Combination keystrokes run together without spaces, i.e., <Ctrl><Alt><Del>.

Hexadecimal numbers are followed by a lower-case "h", as in 80h or A23h.

NOTE

Indicates a special note on a related subject.

CAUTION!

Indicates an area where caution should be exercised.

WARNING!

Indicates an area where damage could occur to the MOD-MUP.

WARNING!

Indicates static precautions should be taken to prevent damage to the MOD-MUP or your system.

TECH TIP

Indicates a helpful hint.

2. Hardware Installation

Installing the MOD-MAC

The first step is to install the MOD-MAC. Follow the directions in the MOD-MAC manual.

Installing the MOD-MUP

After the MOD-MAC has been installed, the programmer can be attached. This is simply a matter of attaching the cable from the MOD-MAC to the MOD-MUP. The end with the pins (DB25 MALE) connects to the MOD-MAC; the end with the holes (DB25 FEMALE) connects to the MOD-MUP.

WARNING!

Make sure the computer is *OFF* when attaching or detaching the adapter cable. Power is applied to the programmer via this cable, and the "in-rush" current could damage the MOD-MUP.

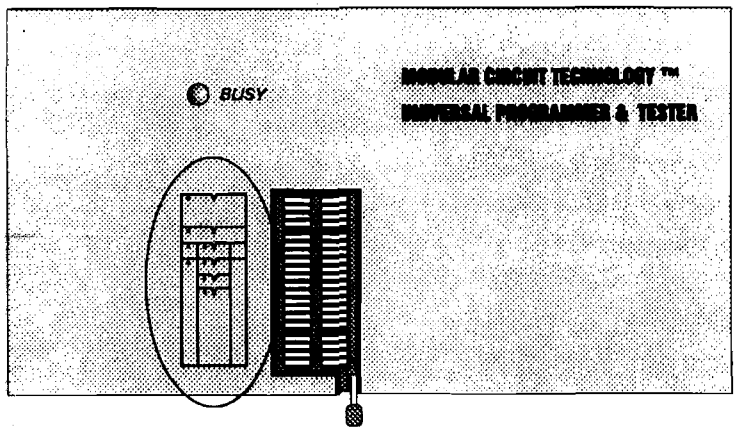
CAUTION!

Do *not* use a standard serial cable to connect the MOD-MAC to the MOD-MUP! Most serial cables only have 9 wires, even though they have 25 pins on each end.

Placing Chips on the MOD-MUP

The MOD-MUP has a ZIF (**Z**ero **I**nsertion **F**orce) socket designed to accept DIP style chips of up to 40

pins. There is a drawing next to the socket to show proper alignment of the chips. See Figure 1 below.



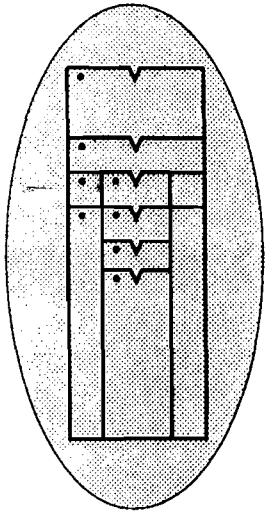


Figure 1
Chip Orientation on the MOD-MUP.

Note the "Notch" on the drawing. This corresponds with the "notch" on most IC chips, and indicates the location of Pin 1. Make sure that the "bottom" of the chip lies flush with the "bottom" of the ZIF socket.

To insert a chip, simply flip the lever into the upright position, insert the chip (making sure the "bottom" of the chip is in the bottom of the socket), and flip the lever down.

3. Software Installation

Copying the Software to your System

The utility software for the MOD-MUP comes on 3 diskettes. These utilities control the MOD-MUP in different ways. Use the list below to check that all the files are there.

Diskette 1

48P02.EXE	Microprocessor programming (8748 series)
48P.DAT	Microprocessor data (8748 series)
51P02.EXE	Microprocessor programming (8751 series)
51P.DAT	Microprocessor data (8748 series)
BPP02.EXE	Bi-polar PROM programming
BPP.DAT	Data for Bi-polar PROMs
DASM48.EXE	Microprocessor dis-assembler (8748 series)
DMT02.EXE	Digital & Memory Tester
EPP02.EXE	E(E)PROM programming
EPP.DAT	E(E)PROM data
HEXOBJ.EXE	HEX to OBJ format converter
HEXOBJ2.EXE	With 256K buffer
README.DOC	Installation notes
SETUP02.EXE	Setup program for MOD-MUP
SETUP.DAT	Setup data
Z8P02.EXE	Microprocessor programming (Z-80 series)
Z8P.DAT	Microprocessor data (Z-80 series)

Diskette 2

PALFORM	A **DIRECTORY** containing "Maps" of various PALs, etc.
20G10-02.EXE	20G10 programming
DASM51.EXE	Microprocessor dis-assembler (8751 series)
PAP.DAT	PAL programming

PAP02.EXE	PAL programming	
PAPA02.EXE	AMD PAL programming	
PARTS02.LST	Supported devices list	
SETUP.DAT	Setup data	
PALFORM Directory:		
10H8.FRM	10L8.FRM	12H6.FRM
12L10.FRM	12L6.FRM	14H4.FRM
14L4.FRM	14L8.FRM	16A4.FRM
16C1.FRM	16H2.FRM	16H8.FRM
16L2.FRM	16L6.FRM	16L8.FRM
16P8.FRM	16R4.FRM	16R6.FRM
16R8.FRM	16RP4.FRM	16RP6.FRM
16RP8.FRM	16X4.FRM	18CV8.FRM
18L4.FRM	20C1.FRM	20G10.FRM
20L10.FRM	20L2.FRM	20L8.FRM
20R4.FRM	20R6.FRM	20R8.FRM
20RA10.FRM	20X10.FRM	20X4.FRM
20X8.FRM	22V10.FRM	22VP10.FRM
AMP18P8.FRM	GAL16.FRM	GAL20.FRM
PLC16V8.FRM	PLC18V8Z.FRM	PLC20V8.FRM
PLC22V8Z.FRM	PLS153.FRM	PLS173.FRM
PLUS16L8.FRM	PLUS20L8.FRM	

Diskette 3

22V10-02.EXE	22V10 programming
22V10.DAT	22V10 data
A18P8-02.EXE	A18P8 programming
A18P8.DAT	A18P8 data
FPL02.EXE	FPL programming
FPL.DAT	FPL data
GAL02.EXE	GAL programming
GAL.DAT	GAL data
IOCHK02.EXE	I/O port check utility
PEEL02.EXE	PEEL programming
PEEL.DAT	PEEL data
S-GAL02.EXE	S-GAL programming
S_GAL.DAT	S-GAL data
SETUP.DAT	Setup data
TEST02.EXE	IC testing

Before installing the software to your system, it is a good idea to make backup copies. Use the backup copies rather than the distribution diskettes for installing the software. The diskettes are not copy-protected.

To make a backup copy, use the DOS DISKCOPY command. See your DOS manual for details.

To install the software onto your system, simply copy the files from the backup diskettes to your hard disk using the DOS XCOPY command. Be sure to use the correct switches (/s & /e). The complete format is **XCOPY A: C: /S /E**. This is assuming that you are copying from drive A: to drive C:. See your DOS manual for details.

If you do not have a hard disk, use the backup copies rather than originals as your working diskettes.

The software does cover a variety of devices, but since many of these devices are handled in a similar fashion, we will divide the software into groups and describe the menu options available. Examples of programming and testing different types of chips will also be given.

Menu options of the various programs will be presented in alphabetical order, which is not necessarily the order that they appear on the screen.

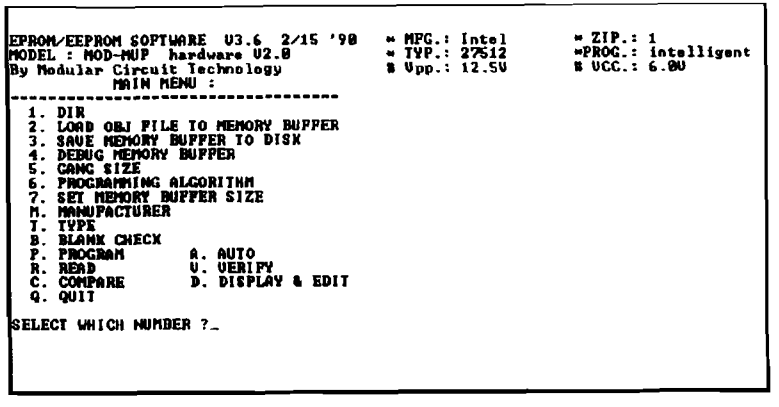
NOTE

The software provided will not function properly in a multi-tasking environment such as Windows or DESQview.

E(E)PROM Software: EPP02.EXE

The EPP02.EXE program is used to program EPROMs and EEPROMs. The main menu is shown below. Each menu item will be discussed individually. To start the EPP02 program, type in the following:

[d:] [path] **EPP02** <Enter>



1. DIR

This menu choice lets you view a directory. The format is very similar to the DOS DIR command:

[d:] [path] [filename] [/P] [/W]

If none of these parameters are specified, the directory will be that of the current drive and directory.

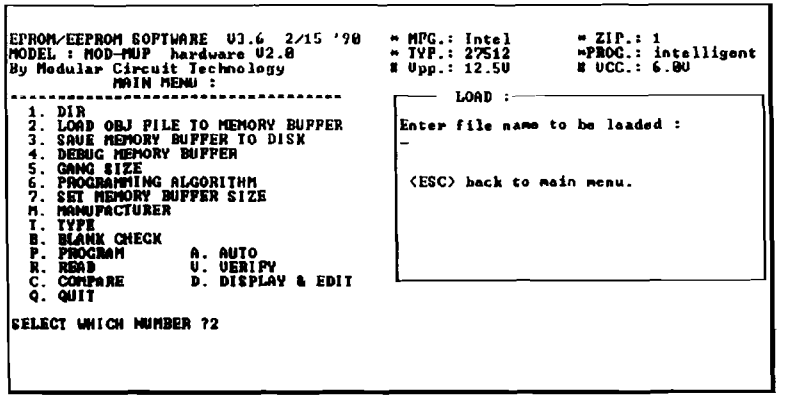
2. LOAD OBJ FILE TO MEMORY BUFFER

This option lets you load an oject file from a disk into the memory buffer. The memory buffer is an area set aside in your computers memory to hold data. Once in the memory buffer, the .OBJ file can be "burned" into

the chip.

An object file is a file that contains the data you wish to program in a form readable by other devices, such as microprocessors or controllers.

See the menu below for details.

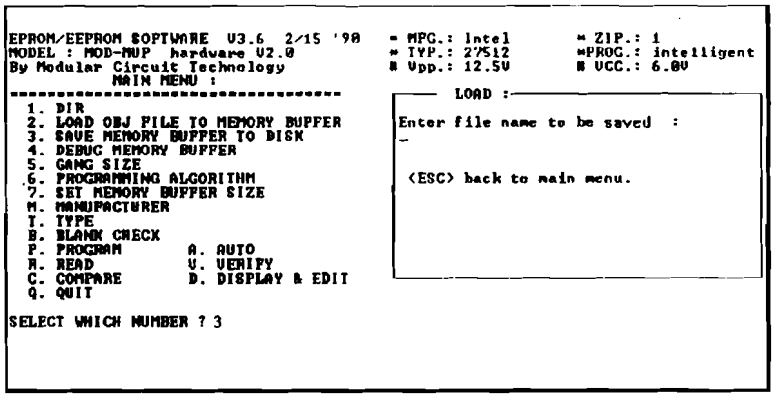


3. SAVE MEMORY BUFFER TO DISK

This option lets you save the contents of the memory buffer to a disk file. If your buffer is empty, this option will write a file full of zeros (00h).

The program will ask you for the file name and the starting address of the buffer you wish to save. The starting addresses will be displayed near the top of the screen.

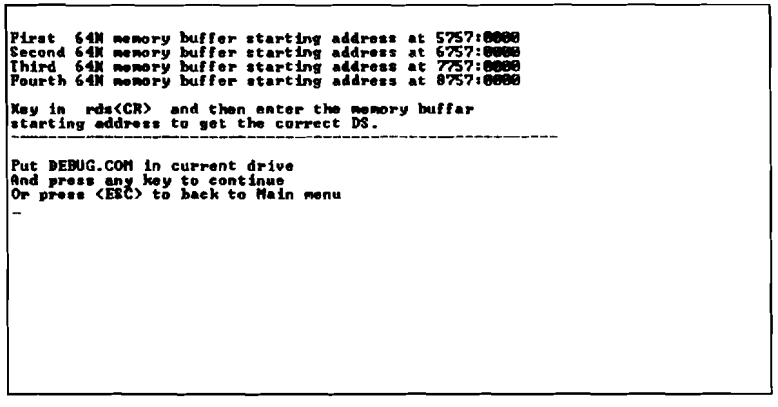
See the menu on the next page for details.



4. DEBUG MEMORY BUFFER

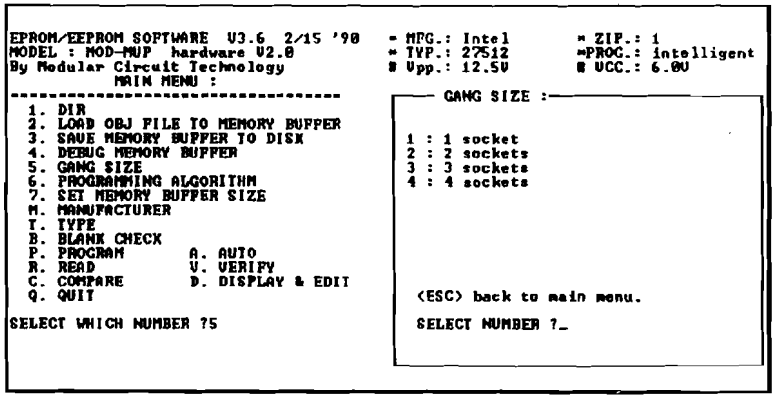
This option uses the DOS DEBUG utility to edit the contents of the memory buffer. This program does not check the path when searching for DEBUG.EXE, so you have to make sure there is a copy of DEBUG.EXE in the same directory as you are working in. See your DOS manual for more information on the DEBUG program.

When <4> is pressed, the following screen appears:



5. GANG SIZE

This option allows you to set the number of sockets used if you are using a multi-socket adapter.



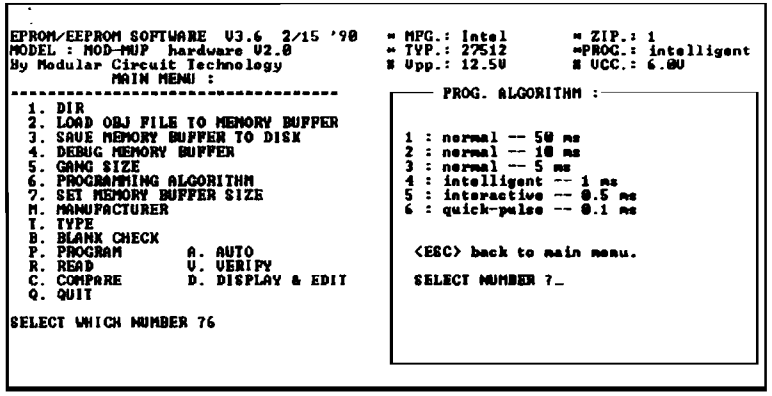
6. PROGRAMMING ALGORITHM

Use this option if you wish to select a different programming algorithm than the one automatically selected when you change manufacturer or chip type.

The screen will display the available algorithms for the type of chip you have selected. Press the number that corresponds with to the desired algorithm. Press <Esc> to return to the Main Menu.

You would change the programming algorithm if you had a newer or different version of a "standard" chip. For example, the new ACME Semiconductor (not a real company) 27256 uses the Quick-Pulse Programming algorithm; but when ACME Semiconductor 27256 is selected, the default algorithm is based on the old ACME Semiconductor 27256, which has a pulse width of 50µS.

Do not change the programming algorithm unless you have detailed information regarding the programming specifications for your chip.



7. SET MEMORY BUFFER SIZE

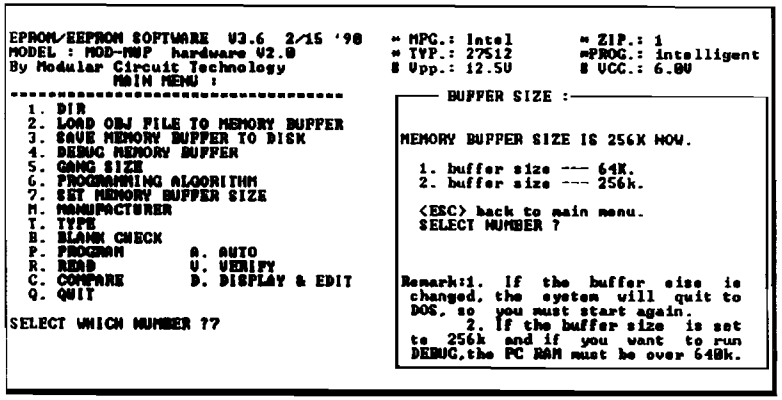
This option allows you to specify the amount of memory to be used by the buffer. Valid choices are 64K and 128K. The default value is 64K. If you change the buffer size, you must exit to DOS and re-enter the program for the changes to take effect.

You would increase the buffer size if you were going to be working with chips of greater than 64K total capacity. It is recommended that the memory buffer be increased if you are going to be working with 64K chips, although it is not necessary. Most programmable memory devices are 8 bits wide, so the total capacity of the chip may be less than 64K. Use the chart on page 16 to help you determine if you need to change the buffer size.

Chip Number	Organization (in bits)	Total capacity (in bytes)
2716	2K x 8	2K
2732	4K x 8	4K
2764	8K x 8	8K
27128	16K x 8	16K
27256	32K x 8	32K
27512	64K x 8	64K
271024	128K x 8	128K

NOTE

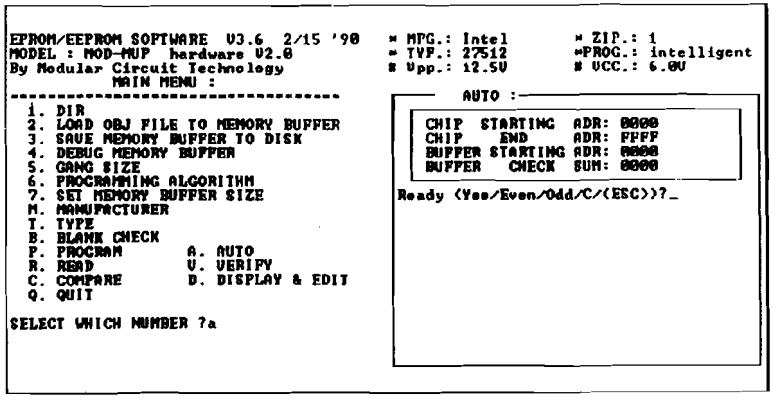
In the chart above, the chip numbers shown are strictly generic. This means that a 27C256, a 28256 and a 27256A, for example, are all laid out the same way.



A. AUTO

This option will perform a BLANK CHECK on the chip currently in the ZIF socket, then attempts to PROGRAM it with the contents of the memory buffer. If an error occurs, the software will terminate the AUTO function and return to the Main Menu. If no error occured during the programming phase, a VERIFY is done. See

the BLANK CHECK, PROGRAM and VERIFY menu options for a more detailed description of these functions.



B. BLANK CHECK

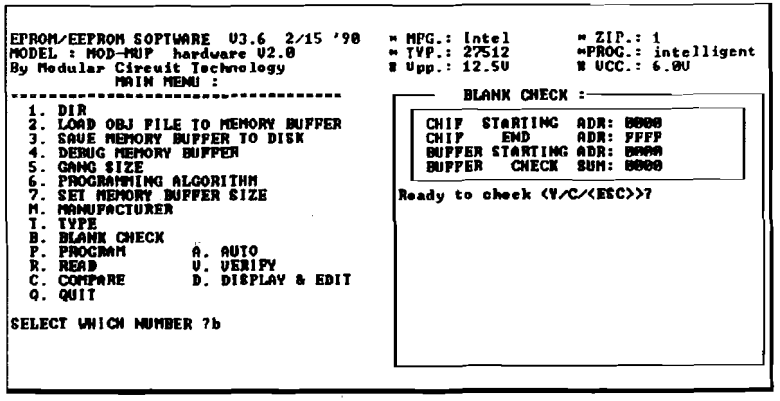
This option checks the chip currently in the ZIF socket for any data that may be present on the chip. If the chip has data, it cannot be programmed.

NOTE

The BLANK CHECK operation is very sensitive to electronic noise, and can sometimes report a chip as having data when it really doesn't. Try to avoid operating the MOD-MAC in areas of high electronic emmisions, such as near power trunk lines, heavy appliances, etc.

When option B is chosen, the screen will display the chip starting address, the chip ending address, the buffer starting address, and the buffer checksum. If everything is in order, press <Y> to start the blank check. If you wish to change any of these values, press <C>. If a chip fails the blank check, the first address that is not

blank will be displayed. If the chip tests as being blank, an "**OK**" message will be displayed.



TECH TIP

If a chip fails the blank check, perform a READ operation on the chip, then choose the DISPLAY & EDIT function. If the buffer shows a FF in every location, then the chip is blank.

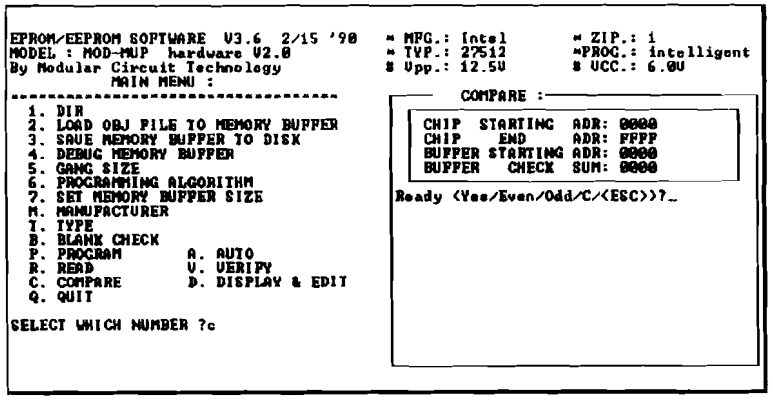
C. COMPARE

Use this option to compare the contents of a chip to the contents of the memory buffer. When option C is chosen, the screen will display the chip starting address, the chip ending address, the buffer starting address, and the buffer checksum. If everything is in order, press <Y> to start the blank check. If you wish to change any of these values, press <C>.

If there are any differences in the two, the screen will display the differences in this form:

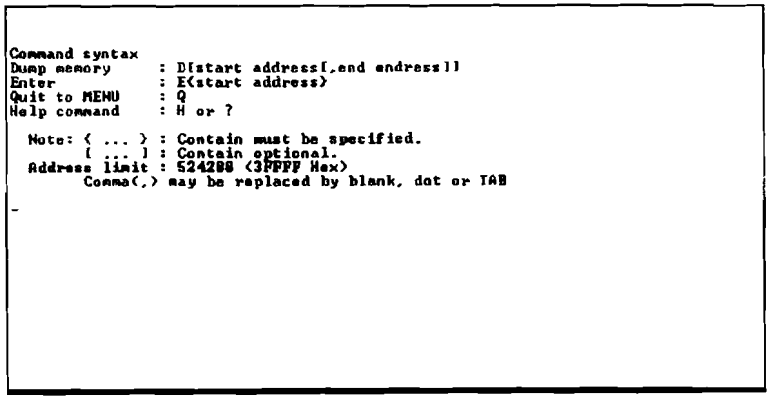
CHIP ADDRESS:DATA - BUFFER ADDRESS:DATA

The monitor will scroll rapidly throught the chip and buffer contents. To pause the display, press <Ctrl><S>. If no differences are found, an "OK" message will be displayed.



D. DISPLAY & EDIT

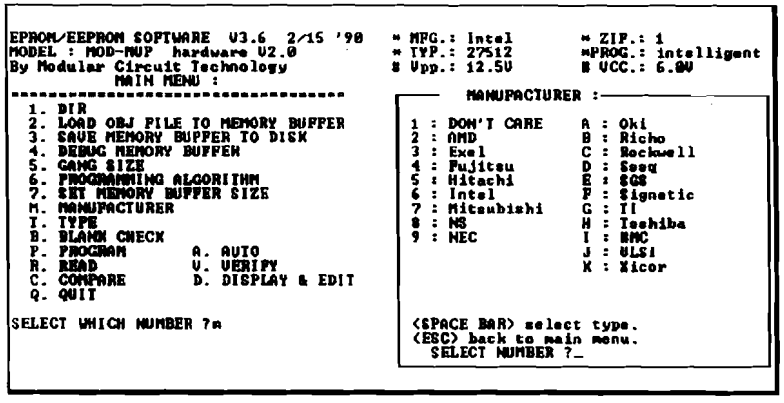
This option allows you to modify the buffer contents, or simply display them. This is similar to option 4 (DEBUG MEMORY BUFFER), but not identical. Option 4 shells out to the DEBUG program, while option D does not.



Command syntax for manipulating the buffer is shown on the screen. All information required by the MOD-MUP software must be in **HEX**!

M. MANUFACTURER

This option allows you to select the manufacturer of the chip you wish to work with. If the manufacturer of your particular chip does not appear on the list, your chip may not be supported.

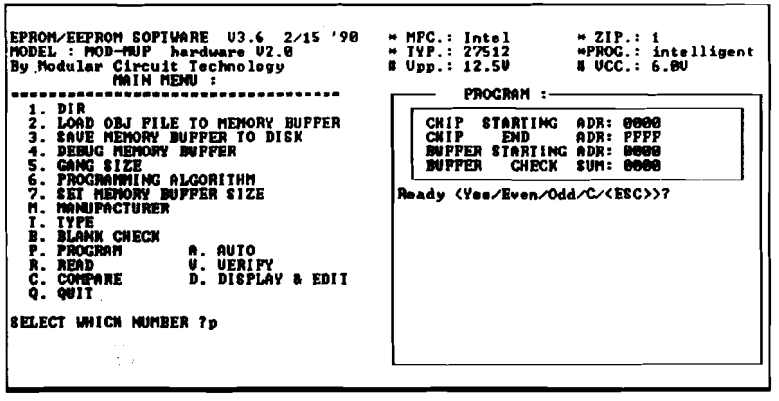


A possible way around this is to choose the "Don't care" option. This allows the chip to be treated as a "generic" chip. If you choose the "Don't care" option, there is no way to guarantee that the programmer will handle the chip properly.

P. PROGRAM

Use this option to program the contents of the memory buffer into a chip. When option P is chosen, the screen will display the chip starting address, the chip ending address, the buffer starting address, and the buffer checksum. If everything is in order, press <Y> to

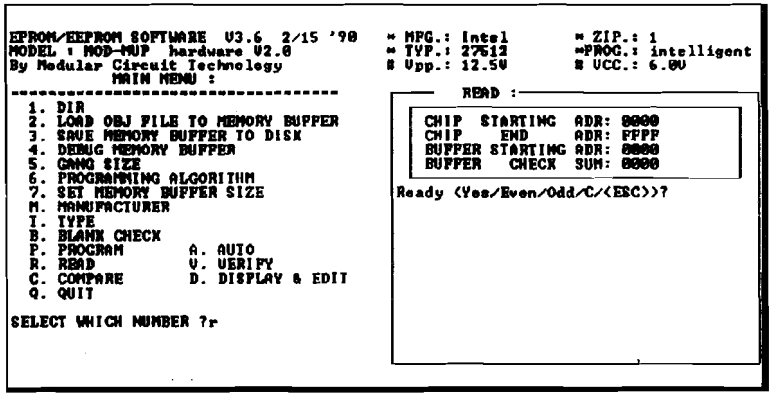
start the programming process. If you wish to change any of these values, press <C>.



Q. QUIT

This option lets you quit the MOD-MUP software and return to DOS. If you are in one of the sub-menus, you must first exit that menu - usually by pressing <Esc> - before exiting the MOD-MUP software.

R. READ

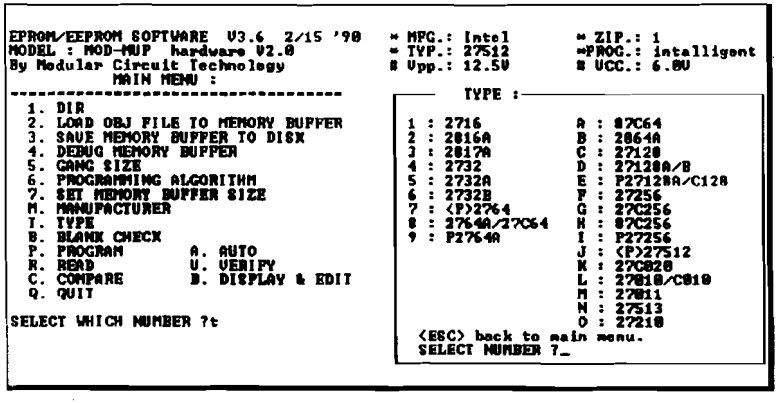


This option will read the contents of a chip and transfer the data to the memory buffer. When option R is chosen, the screen will display the chip starting

address, the chip ending address, the buffer starting address, and the buffer checksum. If everything is in order, press <Y> to start the read process. If you wish to change any of these values, press <C>.

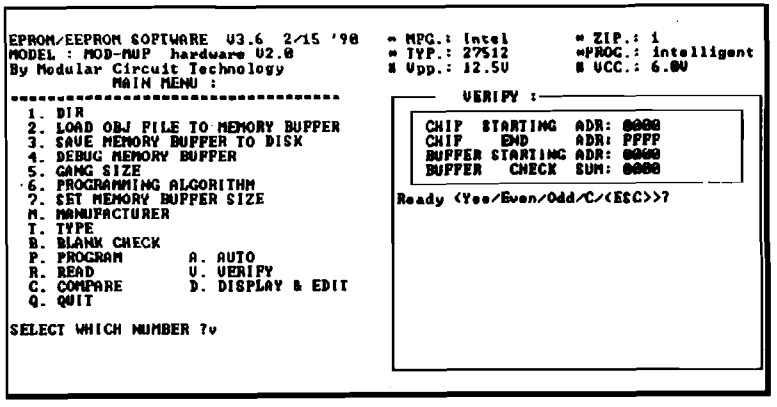
T. TYPE

This option allows you to choose the type of chip to be used. The screen will show the available types of chips for the current manufacturer. In some instances, there are more chips available than can be shown in one screen. Use the <PgDn> and <PgUp> keys to scroll through the list.



V. VERIFY

This option lets you compare a portion of the contents of the chip with the a portion of the contents in the buffer. Usually this is done after a program procedure to insure that the buffer programmed the chip correctly.



Example

For example purposes, we are going to copy the contents of a Texas Instruments 27C256 to a Hitachi 27256.

1. Making sure all cables are attached and your computer system is working properly, start the EPP02 program.

2. Insert the SOURCE chip (TI 27C256) into the ZIF socket and close the handle.

3. Press <M> to select the manufacturer. When the manufacturer menu comes up, press <G> to select TI. Do not press return. You will notice that the manufacturer displayed in the status area near the top of your screen changed to TI.

4. Press <spacebar> to access the type selection menu.

- or -

4. Press <Esc> to return to the main menu. Press <T> to access the type selection menu. Do not press <Return>.

5. Press <7> to select 27(P)C256.

6. Press <R> to read the contents of the source chip into the memory buffer.

7. Remove the source chip from the socket.

8. Insert the TARGET chip (Hitachi 27256) into the ZIF socket and close the handle.

9. Press <M> to select the manufacturer. When the manufacturer menu comes up, press <5> to select Hitachi. Do not press return. You will notice that the manufacturer displayed in the status area near the top of your screen changed to Hitachi.

10. Press <spacebar> to access the type selection menu.

- or -

10. Press <Esc> to return to the main menu. Press <T> to access the type selection menu. Do not press <Return>.

11. Press <9> to select 27(C)256. Note that this selection will work for both a 27256 and a 27C256.

12. Press <B> to blank check the target chip. If the chip passes the blank check, proceed to step 13, otherwise, erase the chip in accordance with the manufacturers instuctions.

13. Press <P> to program the contents of the memory buffer into the target chip.

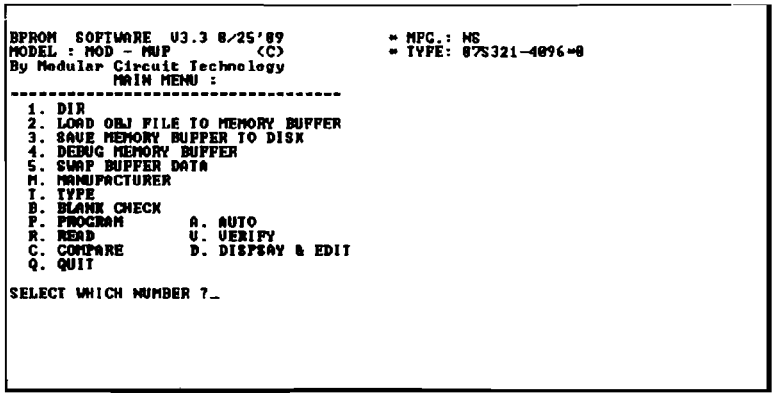
14. When programming is completed, press <V> to verify that the chip has been properly programmed.

Once you are familiar with the steps involved, you can combine steps 12, 13 & 14 into one step by using the AUTO option.

Bi-polar PROM Software: BPP02.EXE

The BPP02.EXE program is used to program Bi-polar PROMs. The main menu is shown below. Each menu item will be discussed individually. To start the BPP02 program, type in the following:

[d:] [path] **BPP02** <Enter>



1. DIR

This menu choice lets you view a directory. The format is very similar to the DOS DIR command:

[d:] [path] [filename] [/P] [/W]

If none of these parameters are specified, the directory will be that of the current drive and directory.

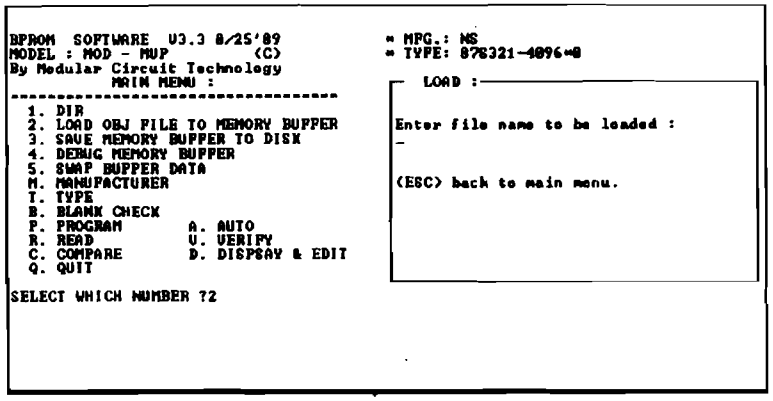
2. LOAD OBJ FILE TO MEMORY BUFFER

This option lets you load an oject file from a disk into the memory buffer. The memory buffer is an area set aside in your computers memory to hold data. Once in the memory buffer, the .OBJ file can be "burned" into

the chip.

An object file is a file that contains the data you wish to program in a form readable by other devices, such as microprocessors or controllers.

See the menu below for details.

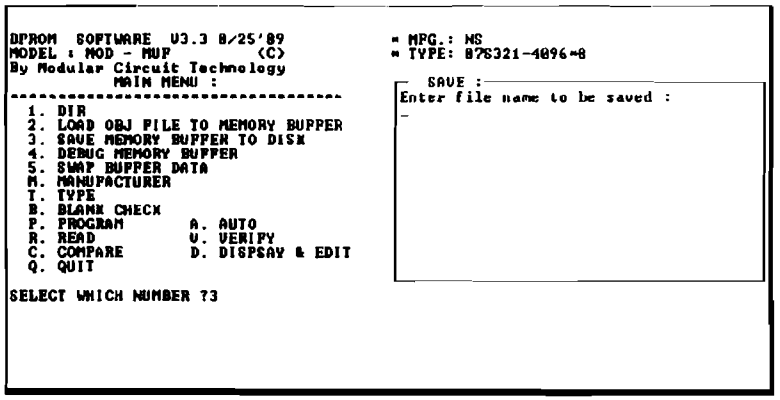


3. SAVE MEMORY BUFFER TO DISK

This option lets you save the contents of the memory buffer to a disk file. If your buffer is empty, this option will write a file full of zeros (00h).

The program will ask you for the file name and the starting address of the buffer you wish to save. The starting addresses will be displayed near the top of the screen.

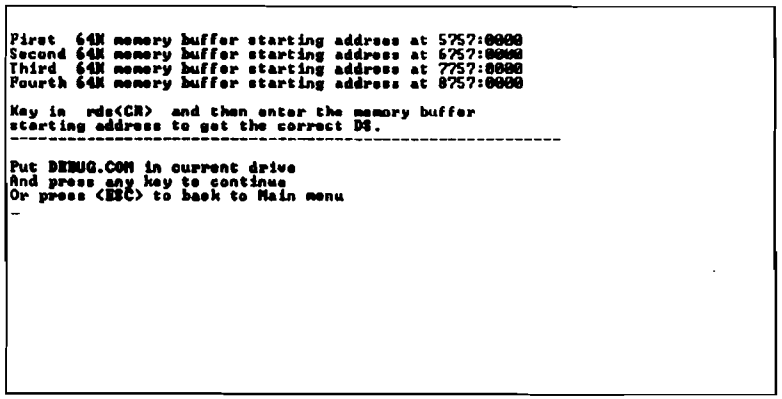
See the menu on page 28 for details.



4. DEBUG MEMORY BUFFER

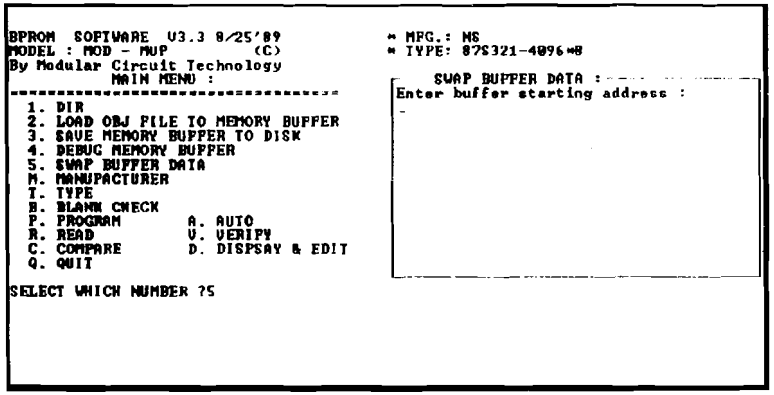
This option uses the DOS DEBUG utility to edit the contents of the memory buffer. This program does not check the path when searching for DEBUG.EXE, so you have to make sure there is a copy of DEBUG.EXE in the same directory as you are working in. See your DOS manual for more information on the DEBUG program.

When <4> is pressed, the following screen appears:



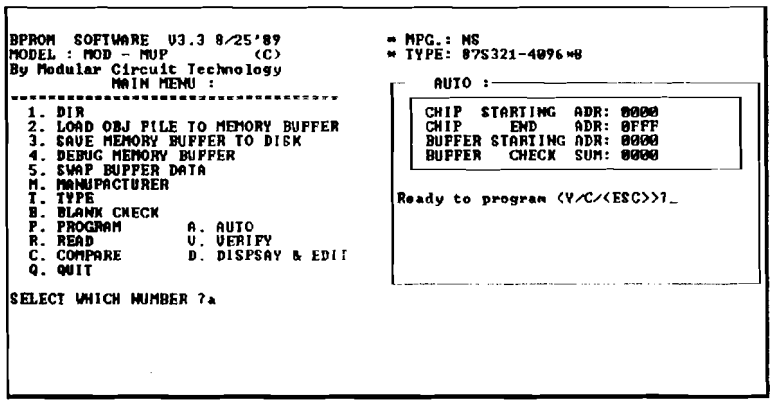
5. SWAP BUFFER DATA

This option allows you to exchange the high and low nybbles (A nybble is half of a byte, or 4 bits) of 8-bit data when you are working with 4-bit devices. If 5 is selected, you must enter the starting and ending addresses of the data to be swapped.



A. AUTO

This option will perform a BLANK CHECK on the chip currently in the ZIF socket, then attempts to PROGRAM it with the contents of the memory buffer. If an error occurs, the software will terminate the AUTO



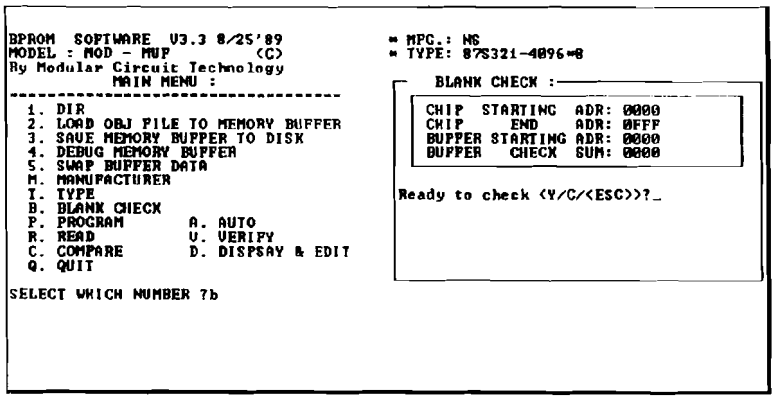
function and return to the Main Menu. If no error occured during the programming phase, a VERIFY is done. See the BLANK CHECK, PROGRAM and VERIFY menu options for a more detailed description of these functions.

B. BLANK CHECK

This option checks the chip currently in the ZIF socket for any data that may be present on the chip. If the chip has data, it cannot be programmed.

NOTE

The BLANK CHECK operation is very sensitive to electronic noise, and can sometimes report a chip as having data when it really doesn't. Try to avoid operating the MOD-MAC in areas of high electronic emmisions, such as near power trunk lines, heavy appliances, etc.



When option B is chosen, the screen will display the chip starting address, the chip ending address, the buffer starting address, and the buffer checksum. If everything is in order, press <Y> to start the blank check.

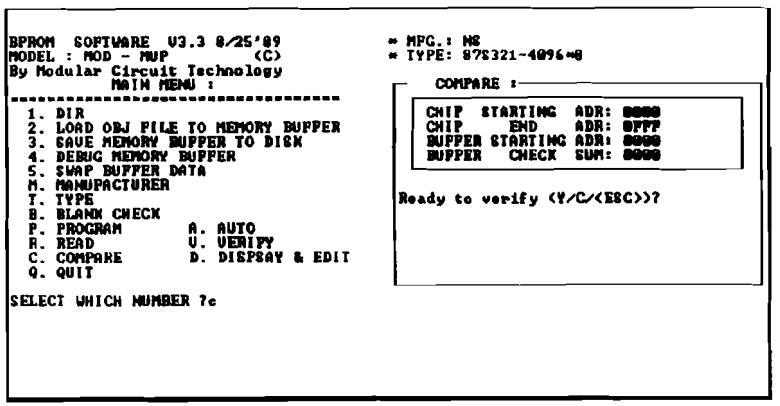
If you wish to change any of these values, press <C>. If a chip fails the blank check, the first address that is not blank will be displayed. If the chip tests as being blank, an "**OK**" message will be displayed.

TECH TIP

If a chip fails the blank check, perform a READ operation on the chip, then choose the DISPLAY & EDIT function. If the buffer shows a FF in every location, then the chip is blank.

C. COMPARE

Use this option to compare the contents of a chip to the contents of the memory buffer. When option C is chosen, the screen will display the chip starting address, the chip ending address, the buffer starting address, and the buffer checksum. If everything is in order, press <Y> to start the blank check. If you wish to change any of these values, press <C>.



If there are any differences in the two, the screen will display the differences in this form:

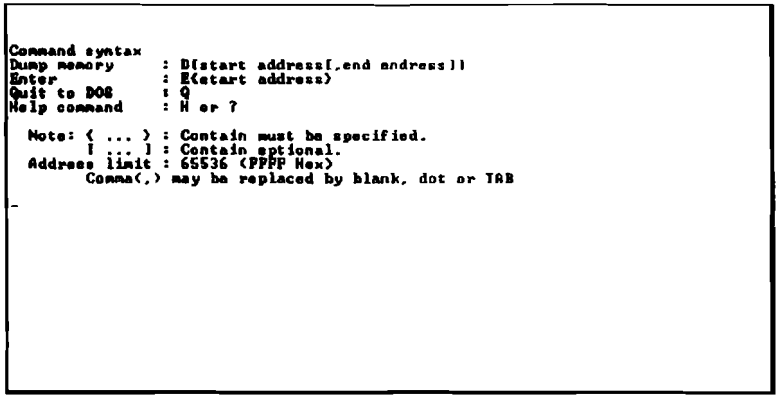
CHIP ADDRESS:DATA - BUFFER ADDRESS:DATA

The monitor will scroll rapidly throught the chip and buffer contents. To pause the display, press <Ctrl><S>. If no differences are found, an "**OK**" message will be displayed.

D. DISPLAY & EDIT

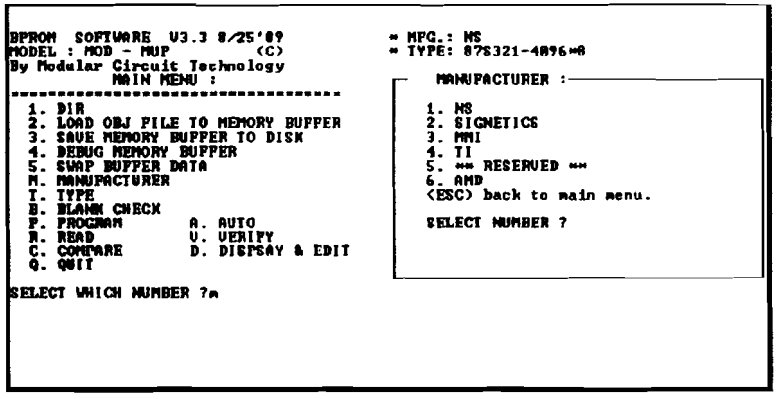
This option allows you to modify the buffer contents, or simply display them. This is similar to option 4 (DEBUG MEMORY BUFFER), but not identical. Option 4 shells out to the DEBUG program, while option D does not.

Command syntax for manipulating the buffer is shown on the screen. All information required by the MOD-MUP software must be in **HEX**!



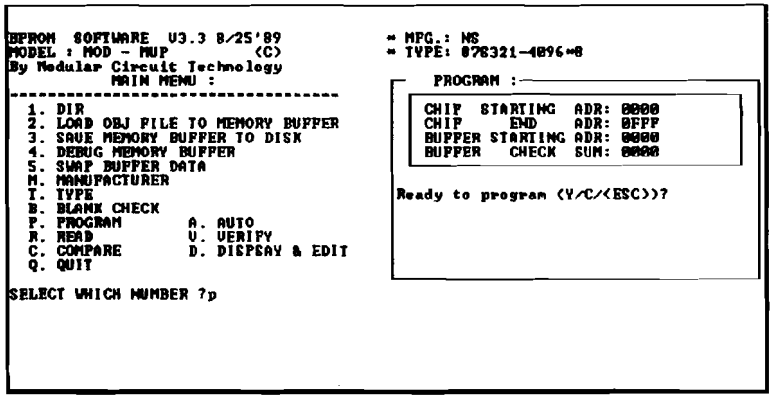
M. MANUFACTURER

This option allows you to select the manufacturer of the chip you wish to work with. If the manufacturer of your particular chip does not appear on the list, your chip may not be supported.



P. PROGRAM

Use this option to program the contents of the memory buffer into a chip. When option P is chosen, the screen will display the chip starting address, the chip ending address, the buffer starting address, and the



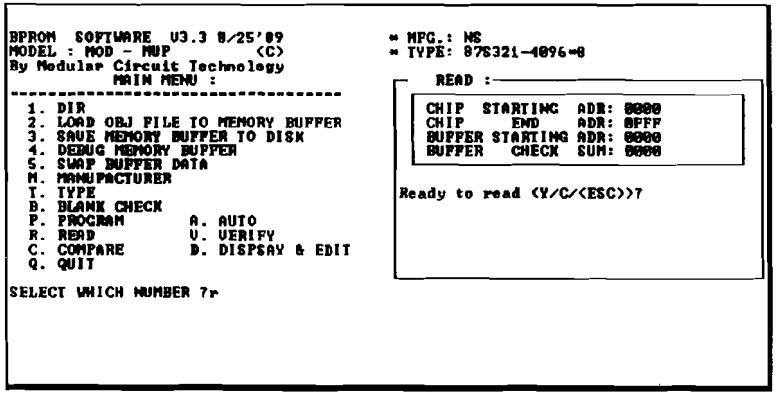
buffer checksum. If everything is in order, press <Y> to start the programming process. If you wish to change any of these values, press <C>.

Q. QUIT

This option lets you quit the MOD-MUP software and return to DOS. If you are in one of the sub-menus, you must first exit that menu - usually by pressing <Esc> - before exiting the MOD-MUP software.

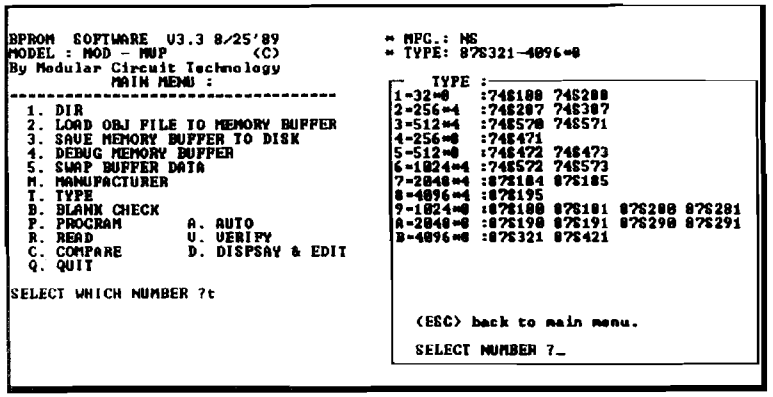
R. READ

This option will read the contents of a chip and transfer the data to the memory buffer. When option R is chosen, the screen will display the chip starting address, the chip ending address, the buffer starting address, and the buffer checksum. If everything is in order, press <Y> to start the read process. If you wish to change any of these values, press <C>.



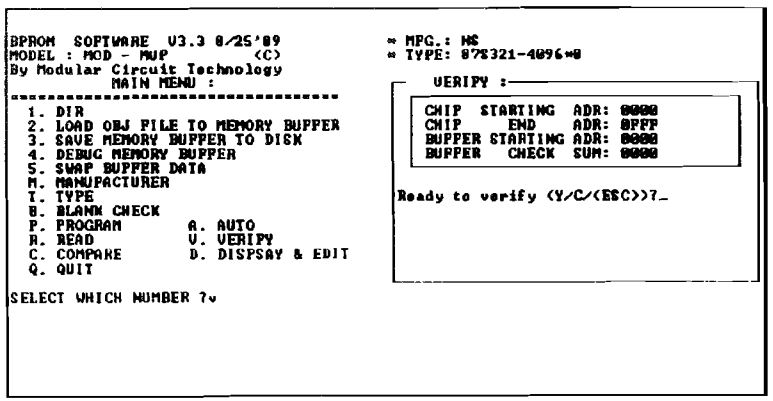
T. TYPE

This option allows you to choose the type of chip to be used. The screen will show the available types of chips for the current manufacturer. In some instances, there are more chips available than can be shown in one screen. Use the <PgDn> and <PgUp> keys to scroll through the list.



V. VERIFY

This option lets you compare a portion of the contents of the chip with the a portion of the contents in the buffer. Usually this is done after a program procedure to insure that the buffer programmed the chip correctly.



Example

For example purposes, we are going to copy the contents of a National Semiconductor 87S321 to another National Semiconductor 87S321.

1. Making sure all cables are attached and your computer system is working properly, start the EPP02 program.

2. Insert the SOURCE chip (NS 87S321) into the ZIF socket and close the handle.

3. Press <M> to select the manufacturer. When the manufacturer menu comes up, press <1> to select National Semiconductor (NS). Do not press return. You will notice that the manufacturer displayed in the status area near the top of your screen changed to NS.

4. Press <spacebar> to access the type selection menu.

- or -

4. Press <Esc> to return to the main menu. Press <T> to access the type selection menu. Do not press <Return>.

5. Press <B> to select 87S321.

6. Press <R> to read the contents of the source chip into the memory buffer.

7. Remove the source chip from the socket.

8. Insert the TARGET chip (NS 87S321) into the ZIF socket and close the handle.

9. Press <B> to blank check the target chip. If the chip passes the blank check, proceed to step 13, otherwise, erase the chip in accordance with the manufacturers instuctions.

10. Press <P> to program the contents of the memory buffer into the target chip.

11. When programming is completed, press <V> to verify that the chip has been properly programmed.

Once you are familiar with the steps involved, you can combine steps 9, 10 & 11 into one step by using the AUTO option.

Microprocessor Software

The programs used to program microprocessors are all very similar, and we will discuss them as though they were one program. When specific information is presented, there will be an appropriate note.

Be sure you are using the right program for the type of chip you wish to program. Use the chart below for help:

Microprocessor	**Filename**
8748 Series	48P02.EXE
8751 Series	51P02.EXE
Z8 Series	Z8P02.EXE

The main menu for 48P02.EXE is shown below. To start a program, type in the following:

[d:] [path] [filename] <Enter>

```
8741/42/48/49 SOFTWARE  V3.4  9/2 '89    * MPG.: Intel        * ZIP.: 1 socket
MODEL : MOD-MUP  hardware V2.0           * TYP.: 8742         *PROG.: intelligent
By Modular Circuit Technology            * Vpp.: 21.0V        * VCC.: 5.0V
          MAIN MENU :
-------------------------------------
  1. DIR
  2. LOAD OBJ FILE TO MEMORY BUFFER
  3. SAVE MEMORY BUFFER TO DISK
  4. DEBUG MEMORY BUFFER
  5. GANG SIZE
  6. PROGRAMMING ALGORITHM
  M. MANUFACTURER
  T. TYPE
  B. BLANK CHECK
  P. PROGRAM          A. AUTO
  R. READ             V. VERIFY
  C. COMPARE          D. DISPLAY & EDIT
  S. SECURITY FUSE BLOW
  Q. QUIT
SELECT WHICH NUMBER ?_
```

1. DIR

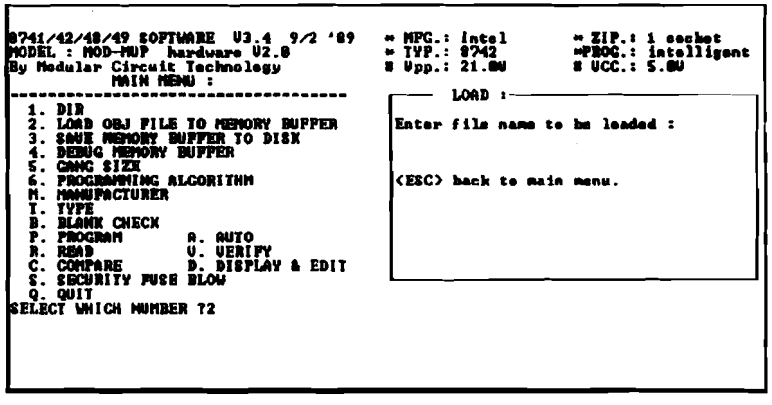
This menu choice lets you view a directory. The format is very similar to the DOS DIR command:

[d:] [path] [filename] [/P] [/W]

If none of these parameters are specified, the directory will be that of the current drive and directory.

2. LOAD OBJ FILE TO MEMORY BUFFER

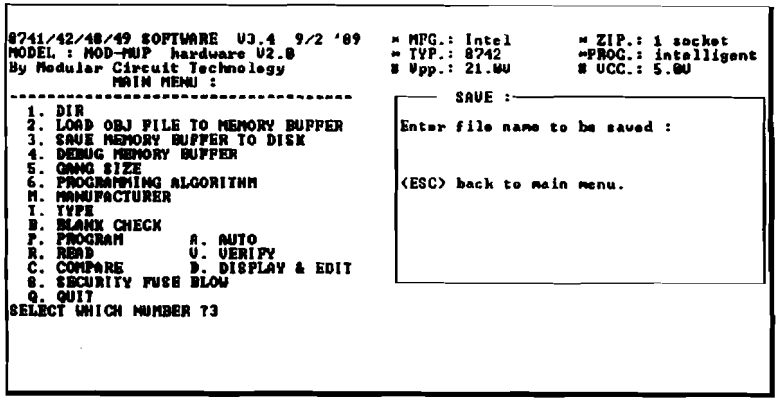
This option lets you load an oject file from a disk into the memory buffer. The memory buffer is an area set aside in your computers memory to hold data. Once in the memory buffer, the .OBJ file can be "burned" into the chip.



An object file is a file that contains the data you wish to program in a form readable by other devices, such as microprocessors or controllers.

3. SAVE MEMORY BUFFER TO DISK

This option lets you save the contents of the memory buffer to a disk file. If your buffer is empty, this option will write a file full of zeros (00h).



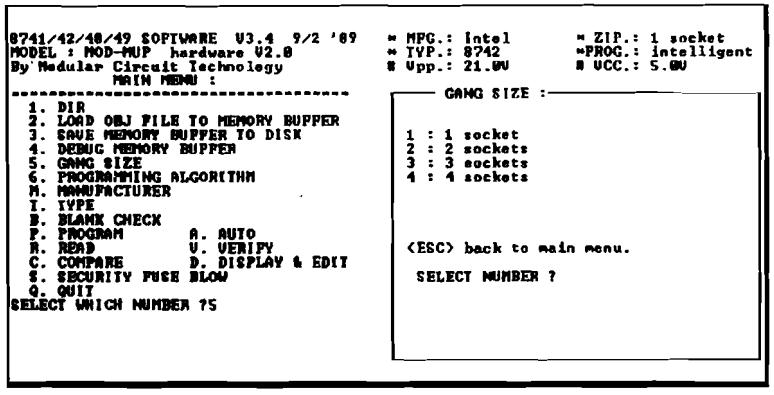
The program will ask you for the file name and the starting address of the buffer you wish to save. The starting addresses will be displayed near the top of the screen.

4. DEBUG MEMORY BUFFER

This option uses the DOS DEBUG utility to edit the contents of the memory buffer. This program does not check the path when searching for DEBUG.EXE, so you have to make sure there is a copy of DEBUG.EXE in the same directory as you are working in. See your DOS manual for more information on the DEBUG program.

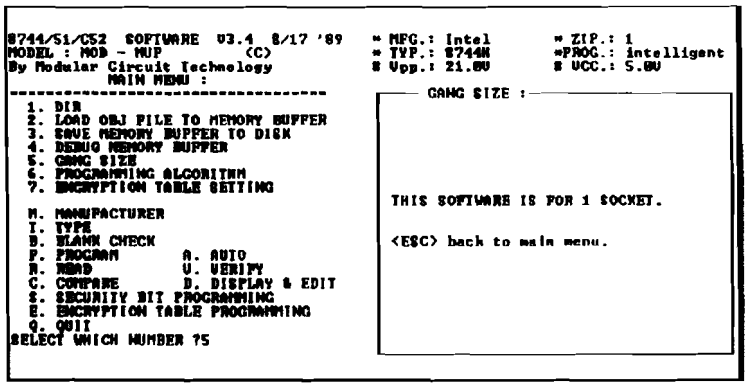
5. GANG SIZE: 48P02.EXE

This option allows you to set the number of sockets used if you are using a multi-socket adapter.



5. GANG SIZE: 51P02.EXE, Z8P02.EXE

Although this option appears on the menu, it is not functional. See below.



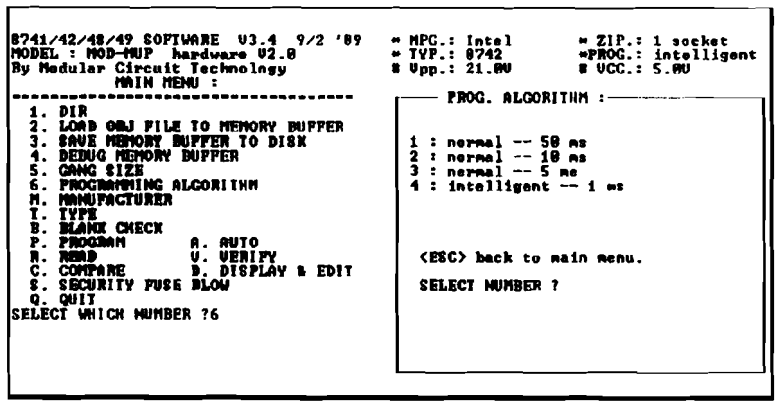
6. PROGRAMMING ALGORITHM

Use this option if you wish to select a different programming algorithm than the one automatically selected when you change manufacturer or chip type.

The screen will display the available algorithms for the type of chip you have selected. Press the number that corresponds with to the desired algorithm. Press <Esc> to return to the Main Menu.

You would change the programming algorithm if you had a newer or different version of a "standard" chip. For example, the new ACME Semiconductor (not a real company) 8748 uses the Quick-Pulse Programming algorithm; but when ACME Semiconductor 8748 is selected, the default algorithm is based on the old ACME Semiconductor 8748, which has a pulse width of 50µS.

Do not change the programming algorithm unless you have detailed information regarding the programming specifications for your chip.



7. ENCRYPTION TABLES SETTINGS: 51P02.EXE

```
ENCRYPTION TABLE :
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF


ENCRYPTION TABLE MANU :

 1.Edit encryption table
 2.Load encryption table from disk
 3.Save encryption table to disk
 4.Enable/Disable encryption table

SELECT WHICH ONE
   or PRESS <ESC> TO MAIN MANU ?


  *** ENCRYPTION TABLE DISABLED ***
```

This option allows you to program the Encryption Table on 8751 series Microprocessors.

A. AUTO

This option will perform a BLANK CHECK on the chip currently in the ZIF socket, then attempts to PROGRAM it with the contents of the memory buffer. If an error occurs, the software will terminate the AUTO function and return to the Main Menu. If no error occured during the programming phase, a VERIFY is done.

```
8741/42/48/49 SOFTWARE  V3.4  9/2 '89     * MFG.: Intel        * ZIP.: 1 socket
MODEL : MOD-MUP  hardware V2.0             * TYP.: 8742         *PROG.: intelligent
By Modular Circuit Technology              # Vpp.: 21.0V       # VCC.: 5.0V
          MAIN MENU :                                    AUTO :
-----------------------------------      ------------------------------------
  1. DIR                                   CHIP  STARTING  ADR: 0000
  2. LOAD OBJ FILE TO MEMORY BUFFER        CHIP    END     ADR: 07FF
  3. SAVE MEMORY BUFFER TO DISK            BUFFER STARTING ADR: 0000
  4. DEBUG MEMORY BUFFER                   BUFFER  CHECK   SUM: 0000
  5. GANG SIZE
  6. PROGRAMMING ALGORITHM                 Ready to program <Y/C/<ESC>>?_
  M. MANUFACTURER
  T. TYPE
  B. BLANK CHECK
  P. PROGRAM          A. AUTO
  R. READ             V. VERIFY
  C. COMPARE          D. DISPLAY & EDIT
  S. SECURITY FUSE BLOW
  Q. QUIT
SELECT WHICH NUMBER ?a
```


B. BLANK CHECK

This option checks the chip currently in the ZIF socket for any data that may be present on the chip. If the chip has data, it cannot be programmed.

NOTE

The BLANK CHECK operation is very sensitive to electronic noise, and can sometimes report a chip as having data when it really doesn't. Try to avoid operating the MOD-MAC in areas of high electronic emmisions, such as near power trunk lines, heavy appliances, etc.

```
8741/42/48/49 SOFTWARE  V3.4  9/2 '89     * MFG.: Intel        * ZIP.: 1 socket
MODEL : MOD-MUP  hardware V2.0             * TYP.: 8742         *PROG.: intelligent
By Modular Circuit Technology              # Vpp.: 21.0V       # VCC.: 5.0V
          MAIN MENU :                                    BLANK CHECK :
-----------------------------------      ------------------------------------
  1. DIR                                   CHIP  STARTING  ADR: 0000
  2. LOAD OBJ FILE TO MEMORY BUFFER        CHIP    END     ADR: 07FF
  3. SAVE MEMORY BUFFER TO DISK            BUFFER STARTING ADR: 0000
  4. DEBUG MEMORY BUFFER                   BUFFER  CHECK   SUM: 0000
  5. GANG SIZE
  6. PROGRAMMING ALGORITHM                 Ready to check <Y/C/<ESC>>?
  M. MANUFACTURER
  T. TYPE
  B. BLANK CHECK
  P. PROGRAM          A. AUTO
  R. READ             V. VERIFY
  C. COMPARE          D. DISPLAY & EDIT
  S. SECURITY FUSE BLOW
  Q. QUIT
SELECT WHICH NUMBER ?b
```

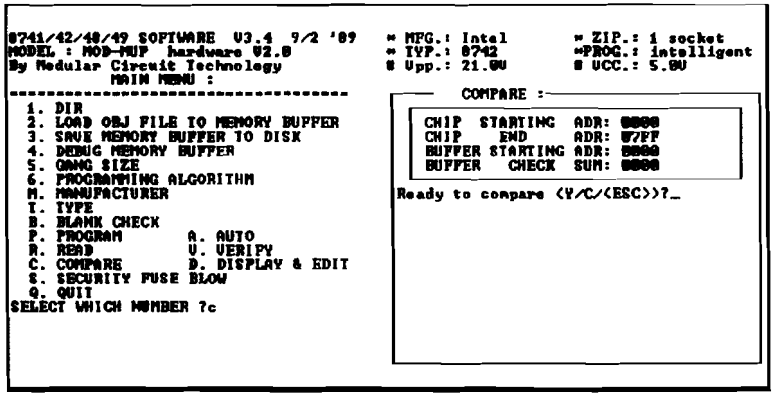
When option B is chosen, the screen will display the chip starting address, the chip ending address, the buffer starting address, and the buffer checksum. If everything is in order, press <Y> to start the blank check. If you wish to change any of these values, press <C>. If a chip fails the blank check, the first address that is not blank will be displayed. If the chip tests as being blank, an "**OK**" message will be displayed.

TECH TIP

If a chip fails the blank check, perform a READ operation on the chip, then choose the DISPLAY & EDIT function. If the buffer shows a FF in every location, then the chip is blank.

C. COMPARE

Use this option to compare the contents of a chip to the contents of the memory buffer. When option C is chosen, the screen will display the chip starting address, the chip ending address, the buffer starting address, and the buffer checksum. If everything is in order, press <Y> to start the blank check. If you wish to change any of these values, press <C>.



If there are any differences in the two, the screen will display the differences in this form:

CHIP ADDRESS:DATA - BUFFER ADDRESS:DATA

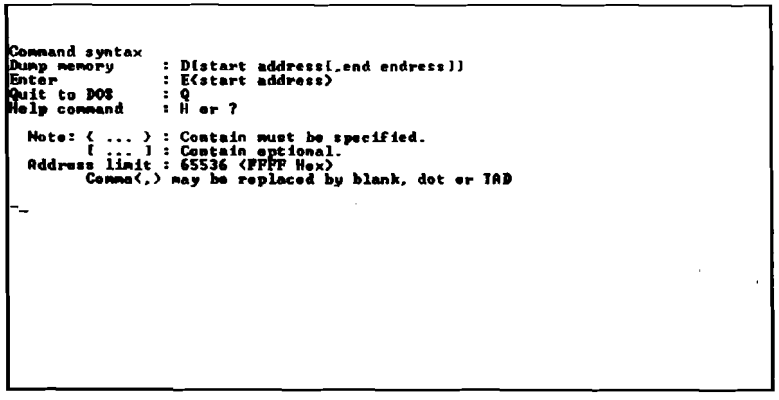
The monitor will scroll rapidly throught the chip and buffer contents. To pause the display, press

<Ctrl><S>. If no differences are found, an "OK" message will be displayed.

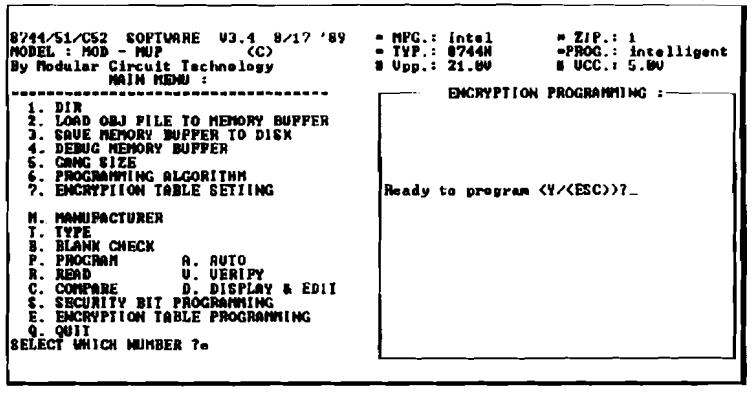
D. DISPLAY & EDIT

This option allows you to modify the buffer contents, or simply display them. This is similar to option 4 (DEBUG MEMORY BUFFER), but not identical. Option 4 shells out to the DEBUG program, while option D does not.

Command syntax for manipulating the buffer is shown on the screen. All information required by the MOD-MUP software must be in **HEX**!



E. ENCRYPTION TABLE PROGRAMMING: 51P02.EXE

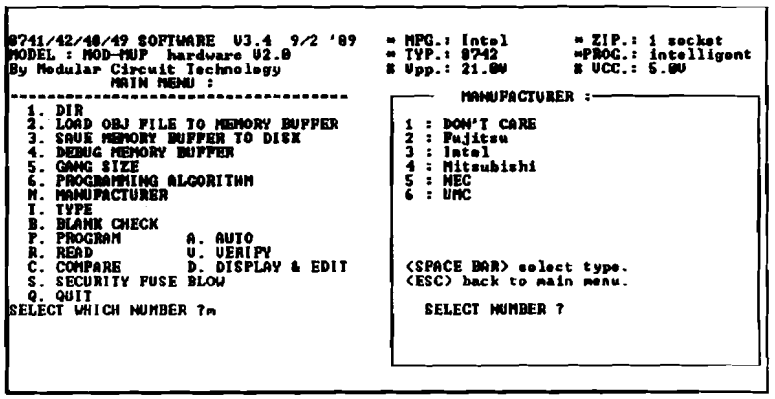


This option programs the encryption table with the current contents of the encryption table buffer.

M. MANUFACTURER

This option allows you to select the manufacturer of the chip you wish to work with. If the manufacturer of your particular chip does not appear on the list, your chip may not be supported.

A possible way around this is to choose the "Don't care" option. This allows the chip to be treated as a



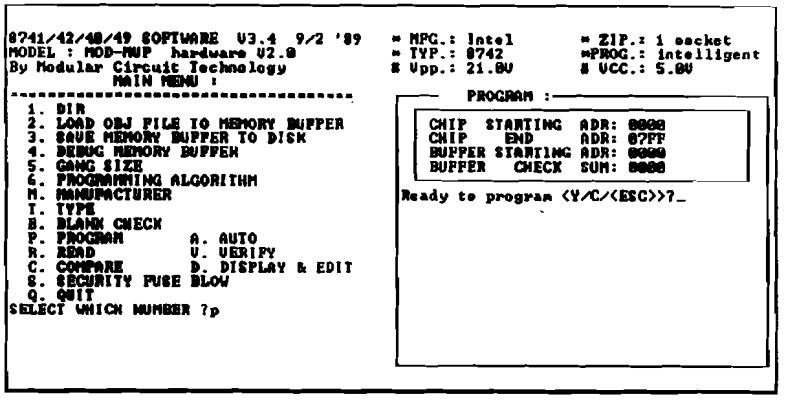
"generic" chip. If you choose the "Don't care" option, there is no way to guarantee that the programmer will handle the chip properly.

NOTE

The 51P02.EXE program does not have a DON'T CARE option.

P. PROGRAM

Use this option to program the contents of the memory buffer into a chip. When option P is chosen, the screen will display the chip starting address, the chip ending address, the buffer starting address, and the buffer checksum. If everything is in order, press <Y> to start the programming process. If you wish to change any of these values, press <C>.

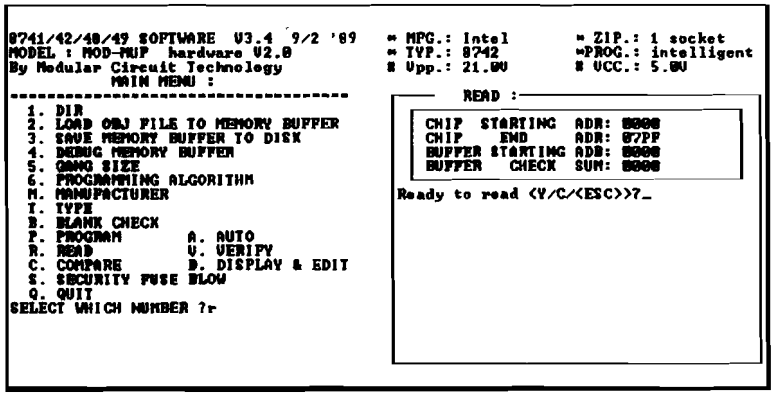


Q. QUIT

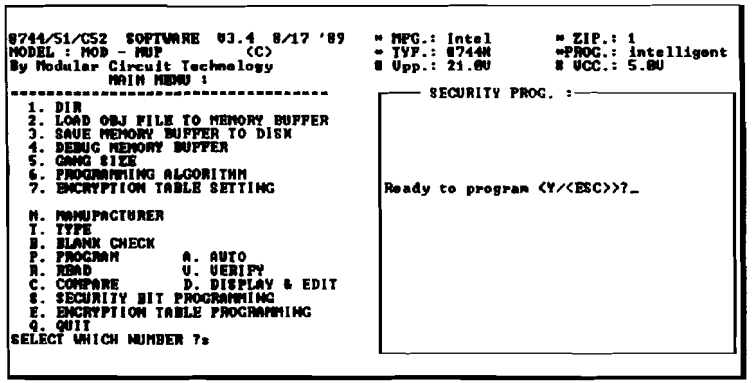
This option lets you quit the MOD-MUP software and return to DOS. If you are in one of the sub-menus, you must first exit that menu - usually by pressing <Esc> - before exiting the MOD-MUP software.

R. READ

This option will read the contents of a chip and transfer the data to the memory buffer. When option R is chosen, the screen will display the chip starting address, the chip ending address, the buffer starting address, and the buffer checksum. If everything is in order, press <Y> to start the read process. If you wish to change any of these values, press <C>.



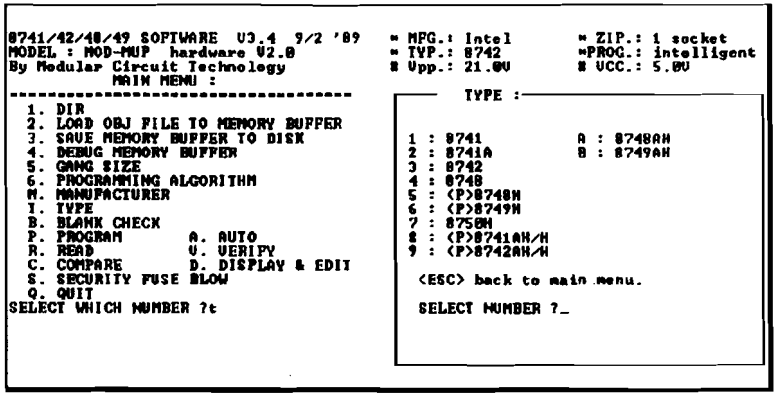
S. SECURITY FUSE PROGRAM: 51P02.EXE



This option will blow the security fuse on the chip currently in the ZIF socket.

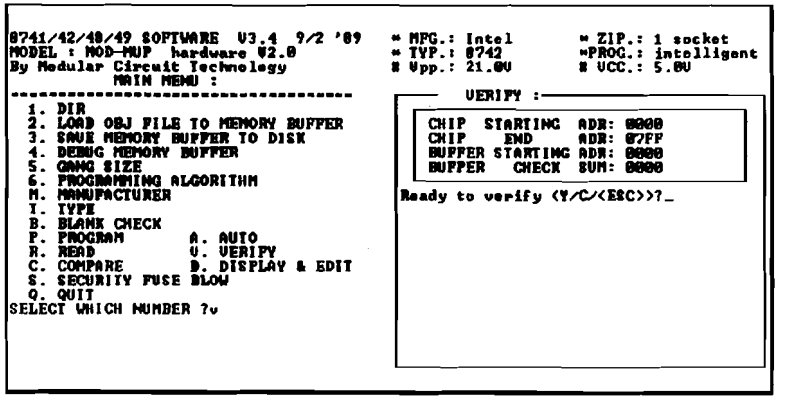
T. TYPE

This option allows you to choose the type of chip to be used. The screen will show the available types of chips for the current manufacturer. In some instances, there are more chips available than can be shown in one screen. Use the <PgDn> and <PgUp> keys to scroll through the list.



V. VERIFY

This option lets you compare a portion of the contents of the chip with the a portion of the contents in the buffer. Usually this is done after a program procedure to insure that the buffer programmed the chip correctly.



Example

For example purposes, we are going to copy the contents of an Intel 8741A to an ACME Semiconductor 87R41A.

1. Making sure all cables are attached and your computer system is working properly, start the EPP02 program.

2. Insert the SOURCE chip (Intel 8741A) into the ZIF socket and close the handle.

3. Press <M> to select the manufacturer. When the manufacturer menu comes up, press <3> to select Intel. Do not press return. You will notice that the manufacturer displayed in the status area near the top of your screen changed to Intel.

NOTE

When the 48P02.EXE program is first loaded, the default manufacturer is Intel. Step 3 could be skipped if this example were the first procedure performed.

4. Press <spacebar> to access the type selection menu.

- or -

4. Press <Esc> to return to the main menu. Press <T> to access the type selection menu. Do not press <Return>.

5. Press <2> to select 8741A.

6. Press <R> to read the contents of the source chip into the memory buffer.

7. Remove the source chip from the socket.

8. Insert the TARGET chip (ACME Semiconductor 87R41A) into the ZIF socket and close the handle.

9. Press <M> to select the manufacturer. When the manufacturer menu comes up, you will notice that ACME Semiconductor is not listed. Since we're not sure of its properties, we'll press <1> to select DON'T CARE. Do not press return. You will notice that the manufacturer displayed in the status area near the top of your screen changed to DON'T CARE.

10. Press <spacebar> to access the type selection menu.

- or -

10. Press <Esc> to return to the main menu. Press <T> to access the type selection menu. Do not press <Return>.

11. The part number on the ACME chip is 87R41A. Since this corresponds most closely with part number 8741A, we will press <2> to select 8741A.

12. Press <B> to blank check the target chip. If the chip passes the blank check, proceed to step 13, otherwise, erase the chip in accordance with the manufacturerss instuctions.

13. Press <P> to program the contents of the memory buffer into the target chip.

14. When programming is completed, press <V> to verify that the chip has been properly programmed.

Once you are familiar with the steps involved, you can combine steps 12, 13 & 14 into one step by using the AUTO option.

Programmable Logic Devices (PLDs)

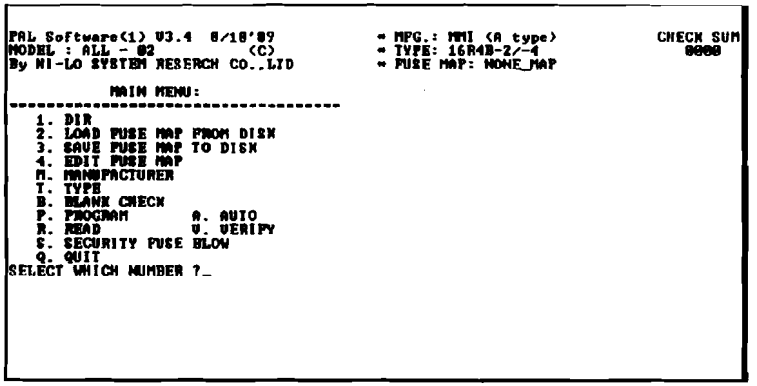
The programs used to program microprocessors are all very similar, and we will discuss them as though they were one program. When specific information is presented, there will be an appropriate note.

Be sure you are using the right program for the type of chip you wish to program. Use the chart below for help:

Type of chip	Filename
PAL	PAP02.EXE
GAL	GAL02.EXE
S-GAL	S-GAL02.EXE
PEEL	PEEL02.EXE
FPL	FPL02.EXE
20G10	20G10-02.EXE
22V10	22V10-02.EXE
A18P8	A18P8-02.EXE

The main menu for PAP02.EXE is shown below. To start a program, type in the following:

[d:] [path] [filename] <Enter>



1. DIR

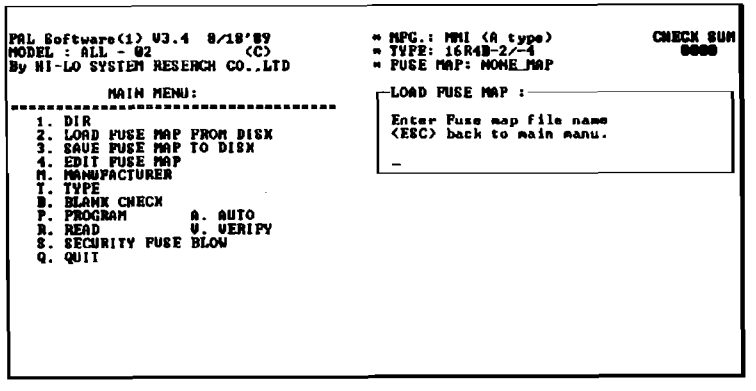
This menu choice lets you view a directory. The format is very similar to the DOS DIR command:

[d:] [path] [filename] [/P] [/W]

If none of these parameters are specified, the directory will be that of the current drive and directory.

2. LOAD FUSE MAP FROM DISK

This option lets you load a fuse map from a disk into the memory buffer. The fuse map must be in JEDEC format or a files saved from this or another modular programmer. Once in the memory buffer, the fuse map can be "burned" into the chip.

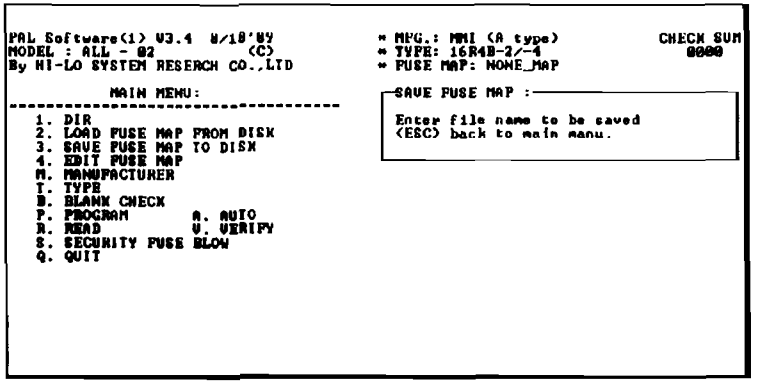


3. SAVE FUSE MAP TO DISK

This option lets you save the fuse map in the memory buffer to a disk file. The program will ask you for the file name. Type in the complete name, including drive and path if you are not using the current directory.

NOTE

Option 3 does *NOT* save the fuse map in JEDEC format. The file created by using option 3 can only be used by the MCT Modular Programmers MOD-MUP and MOD-MPL.



4. EDIT FUSE MAP

This option lets you view or edit the fuse map. Use the arrow keys on your numeric keypad to move around in the display.

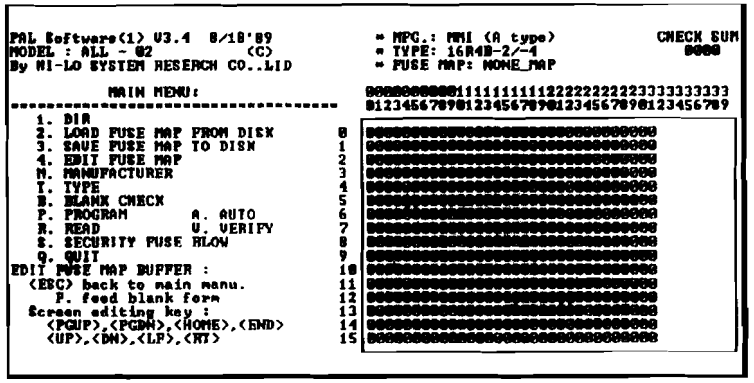
The display has 3 symbols to represent possible conditions of the fuse. Use the chart below to help you decipher the display.

Symbol	Meaning
0	Fuse is not blown/will not be blown.
1	Fuse is blown/will be blown.
N	No fuse at this location.

To edit the fuse map, simply move the cursor to the desired position, and press either <1> or <0>.

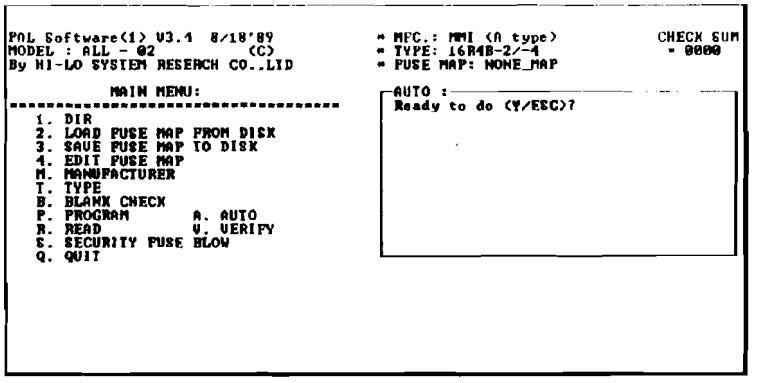
NOTE

The <F> key will erase the contents of the memory buffer. If you press <F> while editing a fuse map, the current fuse map will erase, and the blank form of the PLD will be shown.



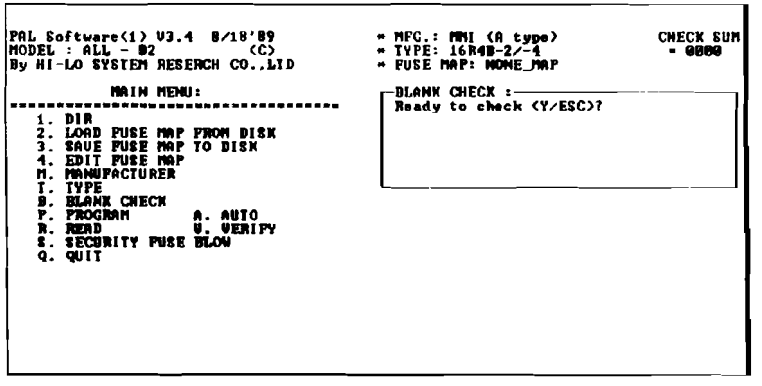
A. AUTO

This option will perform a BLANK CHECK on the chip currently in the ZIF socket, attempts to PROGRAM it with the contents of the fuse map in the memory buffer, performs a VERIFY, then blows the security fuse. If an error occurs, the software will terminate the AUTO function and return to the Main Menu.



B. BLANK CHECK

This option checks the chip currently in the ZIF socket for any data that may be present on the chip. If the chip has data, it cannot be programmed.



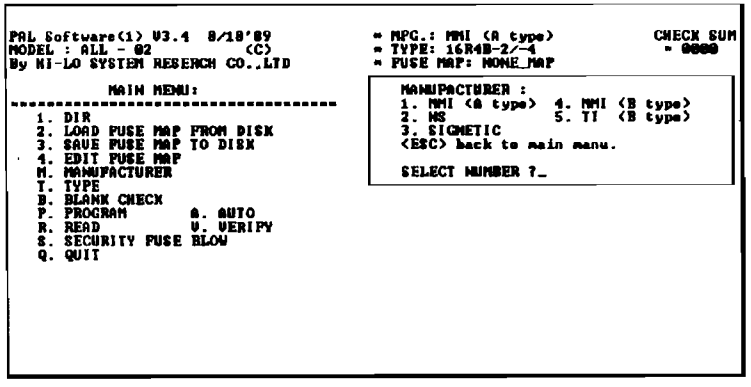
NOTE

The BLANK CHECK operation is very sensitive to electonic noise, and can sometimes report a chip as having data when it really doesn't. Try to avoid operating the MOD-MAC in areas of high electronic emmisions, such as near power trunk lines, heavy appliances, etc.

If a chip passes the blank check, a "**Blank Check OK**" message will be displayed.

M. MANUFACTURER

This option allows you to select the manufacturer of the chip you wish to work with. If the manufacturer of your particular chip does not appear on the list, your chip may not be supported.



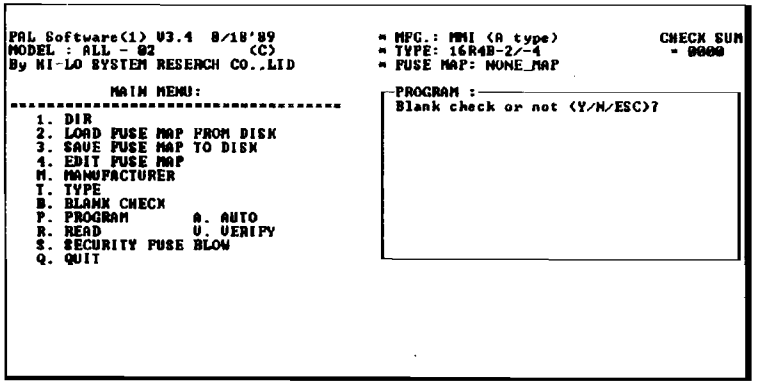
P. PROGRAM

Use this option to program the contents of the fuse map in the memory buffer into a chip. When option P is chosen, you will be asked if you wish to blank check the PLD prior to programming. Press <Y> to do a blank check, or <N> to start programming. If a program operation is attempted on a non-blank PLD, there are two possible responses:

1. If the PLD security fuse is blown, the program operation will halt immediately.

2. If the PLD security fuse is NOT blown, the program operation will continue, but fuses that have

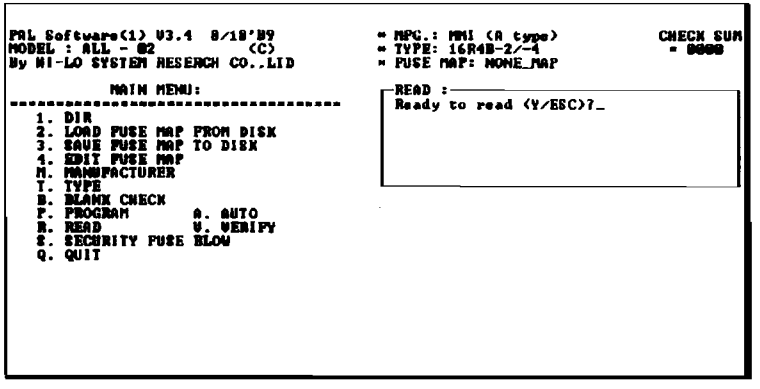
already been blown will remain in that state. This usually means that the function of the logic in the chip will be considerably different than what you had intended.



Q. QUIT

This option lets you quit the MOD-MUP software and return to DOS. If you are in one of the sub-menus, you must first exit that menu - usually by pressing <Esc> - before exiting the MOD-MUP software.

R. READ

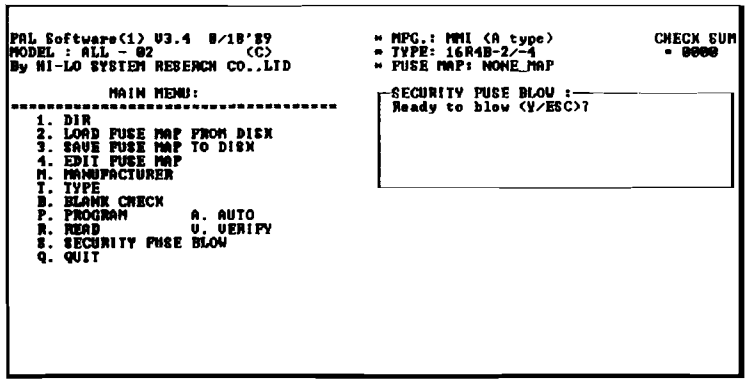


This option will read the contents of a PLD and transfer the data to the memory buffer. You will only be

able to read chips where the security fuse in not blown.

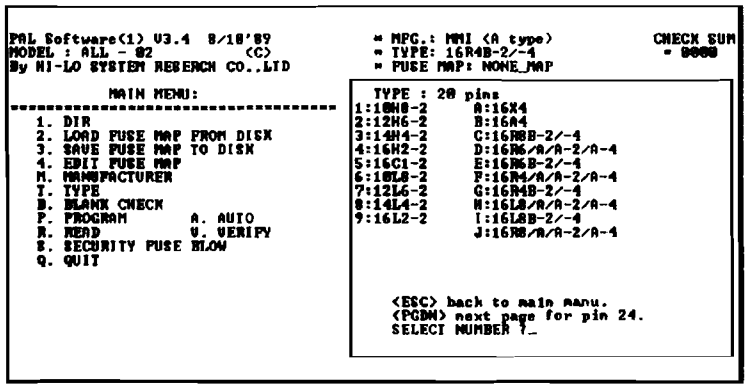
S. SECURITY FUSE BLOW

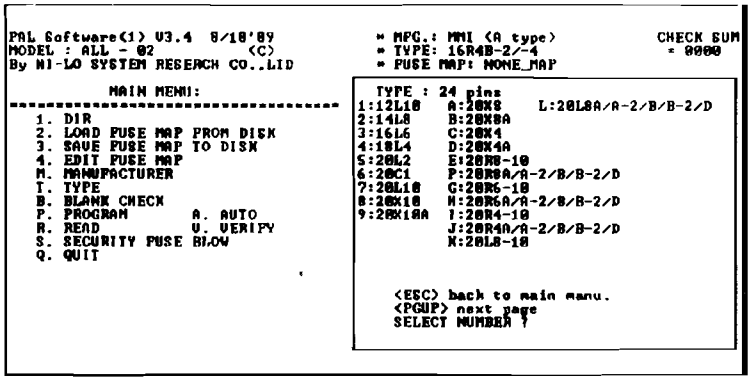
This option lets you blow the security fuse on the selected PLD. By blowing the security fuse, you preve all further read/write access to the PLD. This allows yc to prevent unauthorized copies of your PLD design to b made.



T. TYPE

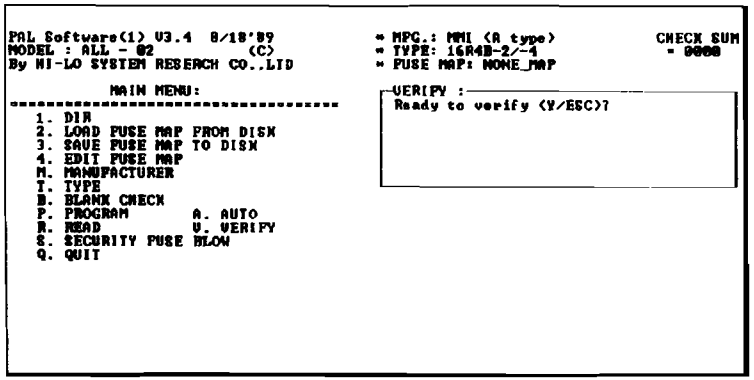
This option allows you to choose the type of PLD to be used. The screen will show the available types of chips for the current manufacturer.





V. VERIFY

This option lets you compare a portion of the contents of the chip with the a portion of the contents in the buffer. Usually this is done after a program procedure to insure that the buffer programmed the chip correctly.



Example

For example purposes, we are going to program a JEDEC fuse map (MAP.FUS) on the C: drive in a directory called CUPL into a Signetics 16L8A.

1. Making sure all cables are attached and your computer system is working properly, start the PAP02 program.

2. Insert the Signetics 16L8A into the ZIF socket and close the handle.

3. Press <M> to select the manufacturer. When the manufacturer menu comes up, press <5> to select Signetics. Do not press return. You will notice that the manufacturer displayed in the status area near the top of your screen changed to Signetics.

4. Press <Esc> to return to the main menu. Press <T> to access the type selection menu. Do not press <Return>.

5. Press <6> to select 16L8A.

6. Press <B> to blank check the target chip. If the chip passes the blank check, proceed to step 6, otherwise, get a blank chip and start over from step 1.

7. Press <2> to load a fuse map from disk. When prompted, enter the complete name of the fuse map file. Our example file is C:\CUPL\FUSE.MAP.

8. Press <P> to program the contents of the memory buffer into the target chip.

9. When programming is completed, press <V> to verify that the chip has been properly programmed.

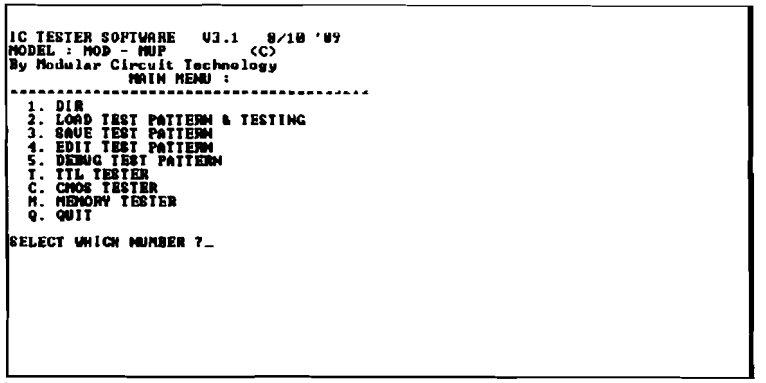
10. If you wish the PAL to be a secure device; i.e., no one wil be able to read or copy your PAL, press <S> to blow the security fuse.

IC/RAM Testing: DMT02.EXE

The DMT02.EXE program allows you to test the functionality of various types of chips.

To start the DMT02 program, type in the following:

[d:] [path] **DMT02** <Enter>



1. DIR

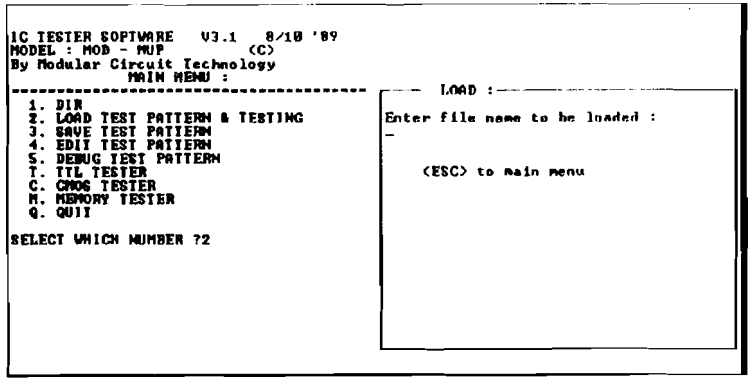
This menu choice lets you view a directory. The format is very similar to the DOS DIR command:

[d:] [path] [filename] [/P] [/W]

If none of these parameters are specified, the directory will be that of the current drive and directory.

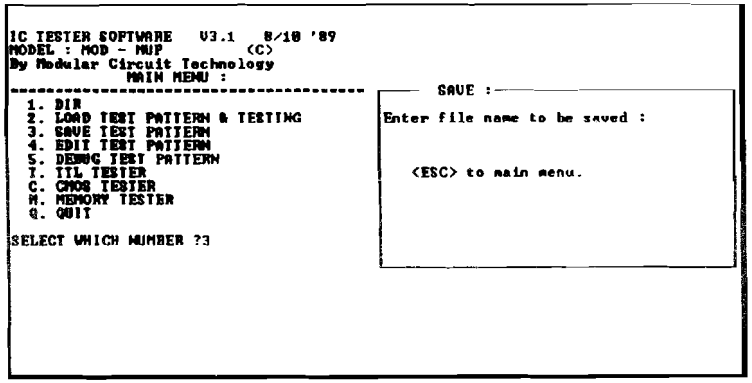
2. LOAD TEST PATTERN & TESTING

This option lets you load a test pattern from a disk file into the memory buffer. When prompted, enter the complete filename of the test pattern.



3. SAVE TEST PATTERN

This option lets you save a test pattern to a disk file. The program will ask you for the file name. Type in the complete name, including drive and path if you are not using the current directory.



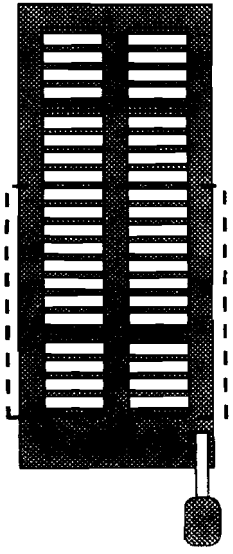
4. EDIT TEST PATTERN

This option uses the DOS DEBUG utility to edit the test pattern. This program does not check the path when searching for DEBUG.EXE, so you have to make sure there is a copy of DEBUG.EXE in the same directory

as you are working in.

The MOD-MUP allows you to customize test patterns. Although a complete test can be done by using the information on a chip's spec sheet, you can save time by testing only certain functions or addresses.

The testing utility will test up to 24 pin chips. The ZIF socket is divided into 3 bi-directional I/O ports: PA, PB and PC. See Figure 2 below for details.



I/O Port	Pin	Pin	I/O port
PC4	1	24	PC7 or V_{CC}
PC5	2	23	PC6 or V_{CC}
PC0	3	22	PC3 or V_{CC}
PC1	4	21	PC2 or V_{CC}
PA0	5	20	PB7 or V_{CC}
PA1	6	19	PB6 or V_{CC}
PA2	7	18	PB5
PA3	8	17	PB4
PA4	9	16	PB3
PA5	10	15	PB2
PA6	11	14	PB1
PA7 or GND	12	13	PB0

Figure 2
Port Locations on the ZIF Socket for Editing Test Patterns

To create a test pattern, you must do the following:

1. Select V_{CC} and GND code.
2. Set I/O Port direction.
3. Set I/O Port test pattern(s).

4. Determine total number of test patterns.
5. Load test pattern(s) into the memory buffer.
6. Test chip(s).

TECH TIP

Perform steps 1-4 on paper prior to actual testing. This makes it much easier to find errors in your test patter.

NOTE

The IC used for example purposes is a 24-pin logic device. It is not a real chip.

1. Select V_{cc} and GND code.

The V_{cc} and GND codes are determined by the number of pins on the chip. Logic, CMOS logic and memory chips have the V_{cc} and GND pin locations standardized. Use the table below to set the V_{cc} and GND code:

# of pins	V_{cc}/GND code
14	00
16	01
18	02
20	03
22	04
24	05

2. Set I/O Port direction.

Each pin on the tester socket is matched with a pin on the chip, and each pin has a given function. Each

function is assigned a code. Use the table below to aid you in setting a direction code:

Function	Code
V_{cc}	0
GND	0
Input	1
Output	0
Don't care (X)	1

The direction code is determined by examining one port at a time; you must determine the the pin function for each pin. Using each I/O Port line as a binary digit, form a binary number with Px0 as the least significant bit and Px7 as the most significant bit. Convert the resulting number to HEX.

An example

PORT A

PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
GND	OUT	IN	IN	OUT	IN	IN	X
0	0	1	1	0	1	1	1 =37h

DIR A = 37

PORT B

PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
X	V_{cc}	IN	IN	OUT	IN	IN	OUT
1	0	1	1	0	1	1	0 =B6h

DIR B = B6

PORT C

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
IN	IN	IN	IN	OUT	IN	IN	OUT
1	1	1	1	0	1	1	0 =F6h

DIR C = F6

3. Set I/O Port test patterns.

The next step is to determine combinations of inputs and desired outputs. Use the table below to help you in selecting the state code.

State		Code
High	(≥ 2.4VDC)	1
Low	(≤ 0.7VDC)	0
GND		0
VCC		1
Don't care (X)		0

Again, examine one port at a time. For each pin designated as an input, use a combination of HIGH and LOW signals. Then, at the output pins, use the expected output. Next, determine the codes for the pins designated V_{cc}, GND, and Don't care. Form a binary number using the same method we used for determining I/O Port direction. See the example below and on the next page.

An Example

For this example, we will use I/O Port A only. We will use the same chip as our example for "Set I/O Port direction".

	PA7 GND	PA6 OUT	PA5 IN	PA4 IN	PA3 OUT	PA2 IN	PA1 IN	PA0 X
input	0	0	0	0	0	0	0	0
output	0	1	0	0	1	0	0	0
number	0	1	0	0	1	0	0	0

= 48h

Repeat this process for all desired combinations.

	PA7 GND	PA6 OUT	PA5 IN	PA4 IN	PA3 OUT	PA2 IN	PA1 IN	PA0 X
input	0	0	0	1	0	0	1	0
output	0	1	0	0	1	0	0	0
number	0	1	0	1	1	0	1	0

= 5Ah

	PA7 GND	PA6 OUT	PA5 IN	PA4 IN	PA3 OUT	PA2 IN	PA1 IN	PA0 X
input	0	0	1	0	0	1	0	0
output	0	1	0	0	1	0	0	0
number	0	1	1	0	1	1	0	0

= 6Ch

	PA7 GND	PA6 OUT	PA5 IN	PA4 IN	PA3 OUT	PA2 IN	PA1 IN	PA0 X
input	0	0	1	1	0	1	1	0
output	0	0	0	0	0	0	0	0
number	0	0	1	1	0	1	1	0

= 36h

The complete test pattern for this port consists of a set of test patterns. The above example has 4 test patterns in its set, so TPA = 48 5A 6C 36.

4. Determine total number of test patterns.

The total number of test patterns is the number of input combinations used. In the previous examples, we only created a set of test patterns for I/O Port A. We will use an assumed set of patterns for I/O Ports B & C. So at this point, you should have the following codes and patterns derived:

VCC (V_{cc} & GND code)	05
DIRA (direction A)	37
DIRB (direction A)	B6
DIRC (direction A)	F6
TPA (test pattern A)	48 5A 6C 36
TPB (assumed)	49 5B 6D 76
TPC (assumed)	49 4C 6D 7A
NO. (# of test patterns)	4

5. Load test patterns into the memory buffer.

After all the test pattern codes are determined, the next step is to load them into the memory buffer. This is done via the DOS DEBUG program. See your DOS manual for instructions on using the DEBUG program.

The chart below shows the buffer addresses for the different codes and patterns:

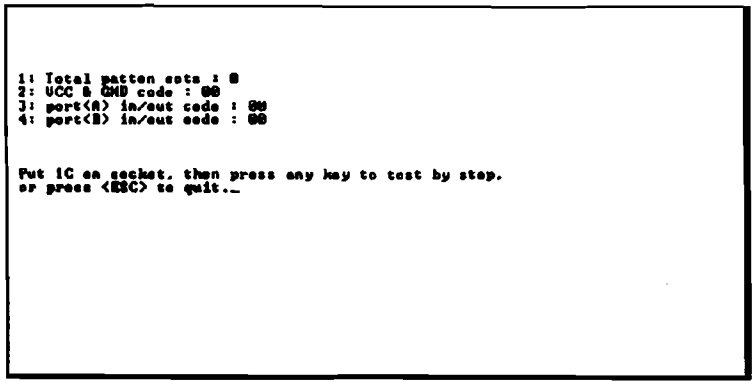
Address	Assignment
000h - 07Fh	TPA
080h - 0FFh	TPB
100h - 17Fh	TPC
180h	DIRA
181h	DIRB
182h	DIRC
183h	VCC
184h	NO.

6. Test chip.

See "5. DEBUG TEST PATTERN".

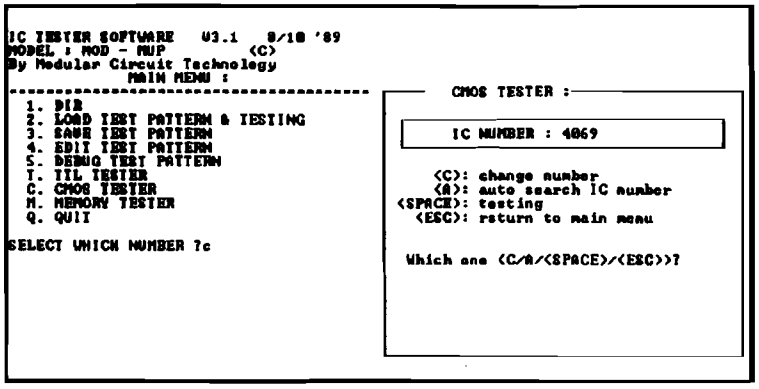
5. DEBUG TEST PATTERN

Although this option is named "DEBUG TEST PATTERN", it has nothing to do with the DOS debug command. This option will test the chip currently in the ZIF socket with the test pattern currently in the memory buffer. The software will display the test pattern set as it tests. A message will be displayed if an error occurs.



C. CMOS TESTER

This option will test a CMOS logic chip. The program will ask for the last digits (5 digits max.) of the generic IC number. If you wish to change the IC number, press <C>. If you are unsure of the IC number, press



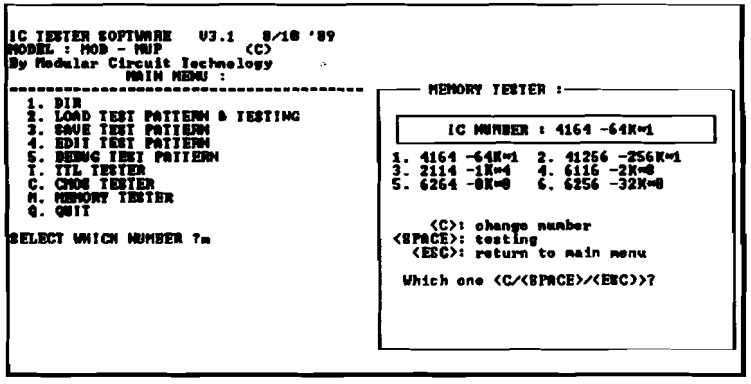
<A> (for Autosearch), and the program will search the chip and find the IC number. Press the spacebar to begin the test.

M. MEMORY TESTER

This option will test a memory chip. The software will ask for the type of memory chip. Select the type of chip you wish to test and press the spacebar to begin the test.

NOTE

The MEMORY TEST operation will only test for functionality, not performance. This means you can test whether or not a memory chip works, but not if the chips performs at the specified speed.

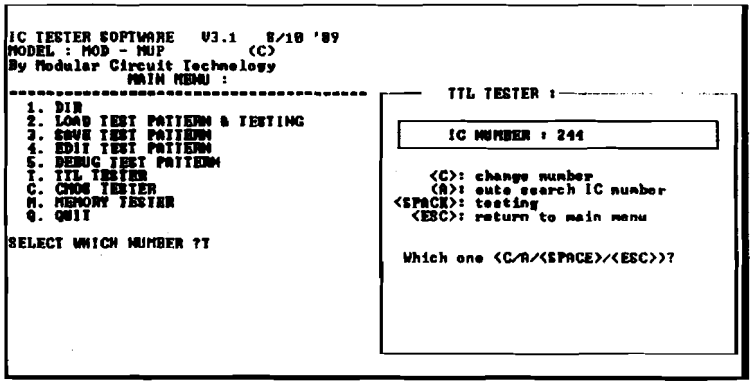


Q. QUIT

This option lets you quit the MOD-MUP software and return to DOS. If you are in one of the sub-menus, you must first exit that menu - usually by pressing <Esc> - before exiting the MOD-MUP software.

T. TTL TESTER

This option will test a TTL logic chip. The program will ask for the last digits (3 digits max.) of the generic IC number. If you wish to change the IC number, press <C>. If you are unsure of the IC number, press <A> (for Autosearch), and the program will search the chip and find the IC number. Press the spacebar to begin the test.



Example

For our example, we will test a 4164 memory chip.

1. Making sure all cables are attached and your computer system is working properly, start the DMT02 program.

2. Insert the 4164 into the ZIF socket and close the handle.

3. Press <M> to select the memory tester. When the menu comes up, press <1> to select the 4164 type chip.

4. Press <spacebar> to perform the test.

NOTES

4. Technical Reference

Devices Supported

EPROM

2716	2716B	2732	2732A
2732B	2764	2764A	27128
27128A	27256	27256A	27512
27512A	27010	27011	27100
271024			

CMOS EPROM

27C16	27C32	27C64	27C128
27C256	27C512	27C010	27C011
27C301	271024		

E²PROM

2816A	2817A	2864A

PAL (20-pin)

10H8	10L8	12H6	12L6
12L10	14H4	14L4	14L8
16A4	16C1	16L2	16H2
16L6	16L8	16P8	16R4
16R6	16R8	16X4	16L8B
16RP4 16RP6	16RP8 168A		
16R6B 16R8B	20L8	20R4	20R6
20R8			

PAL (24-pin)

12L10	14L8	16L6	18L4
20L2	20C1	20L8	20L10
20R4	20R6	20R8	20RA10
20RS4 20RS8	20RS10	20X4	120X8
20X4A 20X8B	20X10	20X10A	

FPL (Signetics)

PLS153	PLS173

GAL

16V8 20V8

S-GAL

PLC16V8 PLC20V8

PEEL

18V8

Misc PLD

20G10 22V10 A18P8

Memory

2114 4164 41256 6116
6264

TTL Logic (74xx, 74Hxx, 74HCxx, 74Lxx, 74LSxx, 74Sxx)

00	01	02	03	04	05	06	07
08	09	10	11	12	13	14	15
16	17	18	19	20	21	22	24
25	26	27	28	30	32	33	37
38	40	42	45	46	47	48	49
51	54	55	74	75	83	85	86
89	90	92	93	95	96	107	109
112	113	114	125	126	128	132	133
136	137	138	139	140	145	147	148
151	152	153	155	156	157	158	160
161	162	163	164	165	166	170	173
174	175	183	189	190	191	192	193
194	195	240	241	242	243	244	245
246	247	248	249	251	253	257	258
259	260	261	266	273	276	279	280
283	289	290	293	295	298	299	323
348	352	353	365	366	367	368	373
374	375	377	378	386	390	399	425
426	490	572	640	645	668	669	670

CMOS Logic

4000	4001	4002	4009	4010	4011	4012	4013
4015	4016	4017	4020	4022	4023	4025	4027
4028	4030	4035	4040	4041	4043	4044	4049
4050	4051	4076	4077	4078	4082	4501	4502
4503	4506	4510	4512	4516	4518	4519	4529
4532	4555	4558	4573				

8748 Microprocessors

8741(A, AH)	8742(H, AH)	8748(H, AH)	8749(H,AH)
8750H	8041(A)	8042(H)	8048(H)
8049H	8050H		

8751 Microprocessors

8744	8751(H, BH)	8742	87C51(FA)
87C252	8752BH		

AMD BPROM

27S12	27S13	27S18	27S19
27S20	27S21	27S28	27S29
27S32	27S33	27S41	27S43
27S180	27S184	27S185	27S190
27S191	27S281	27PS41	27PS185

Fujitsu BPROM

7111	7112	7113	7114
7115	7116	7117	7118
7121	7122	7123	7124
7127	7128	7131	7132
7137	7138	7141	7142
7151	7152		

MMI BPROM

63S080	63S081	63S140	63S141
63S240	63S241	63S280	63S281
63S440	63S441	63S480	63S481
63S840	63S841	63S880	63S881
63S1640	63S1641	63S1681	63S3281

National Semicontuctor BPROM

74S188	74S287	74S288	74S387
74S471	74S472	74S473	74S570
74S572	74S573	87S180	87S181
87S184	87S185	87S190	87S191
87S195	87S280	87S281	87S290
87S291	87S321	87S421	

Signetics BPROM

82S23	82S123	82S126	82S129
82S130	82S131	82S135	82S136
82S137	82S180	82S181	82S184
82S185	82S190	82S191	82S195
82S321	82S2708		

Texas Instruments BPROM

24S10	24S81	24SA10	24SA81
28L22	28L42	28L86	28L166
28LA22	28LA42	28S41	28S42
28S86	28S166	28S2708	28SA41
28SA42	28SA86		

NOTE

The list of supported devices was current at the time this manual was written, and is subject to change without notice. Print the PARTS02.LST file for a current list of supported devices.

5. Trouble-Shooting

"When I turn my computer on, I get no beeps, the fan doesn't spin, nothing happens!"

1. The power cord may be disconnected from the computer or the wall. Check the power cable.

2. Your MOD-MAC may be grounding improperly. Take the card out of the system and try it again.

3. Your power supply may not have sufficient power to drive both your system and the MOD-MAC.

"When I try to use a modular programmer, I get communication error messages!"

1. You may not have the I/O port set correctly for your programmer. Double check the I/O port assignment.

2. You may not have a chip correctly inserted in the ZIF socket. Make sure your chip is correctly installed and the handle is down.

3. There may not be good connection between the MOD-MAC and the MOD-MUP. Double check the cable connection.

4. Your system may be running too fast. Try slowing your system down as much as possible, or try using an IBM XT or compatible.

5. The bus speed on your system may be too fast. The MOD-MAC will not run with bus speeds > 8MHz.

"When I install the MOD-MAC, some of my other peripherals start behaving strangely!"

1. You are probably experiencing an I/O port conflict. Double check the I/O port assignments on all your peripherals, including the MOD-MAC.

"When I try to select a PAL, I get a 'file not found' message!"

1. The PALFORM directory must be a subdirectory of the directory that contains PAP02.

10 Things to do Before Calling Your Dealer

1. Reboot the computer and try again.

2. If you change switches or jumpers, write down the original settings.

3. Repeat all the steps, following the instructions in this manual.

4. Make sure all cards and cables are firmly attached.

5. Remove any memory resident programs from memory.

6. See if your problem is listed in the Trouble-Shooting section.

7. Try it on another system.

8. Compare system requirements with your configuration.

9. Ask your in-house "guru" (every office has one).

10. Ask whoever installed the product.

6. Glossary

BASE ADDRESS

1. The address in memory where a BIOS extension starts.

2. The amount of memory physically present on the motherboard; including EXTENDED memory on AT-compatible systems

BASE MEMORY

Memory available to the OS. In an IBM or compatible running DOS, this is usually 640K.

BIOS

BASIC INPUT OUTPUT SYSTEM. A collection of instructions and data that controls communication between the CPU and its peripherals. Usually located on a ROM; PROM, EPROM or E²PROM chip.

BIT

Binary digit. The smallest usable unit of data. Consists of a single binary digit that can take the value of 0 (FALSE or OFF) or 1 (TRUE or ON).

BOOT

From the phrase "pulled up by ones bootstraps". This is the computer starting itself, either from turning on the power, pressing a reset switch, or issuing a software command (<Ctrl><Alt><Del>).

BUS

1. A series of parallel conductors that form a major interconnection between the CPU and its peripheral sub-systems.

Depending on its design, a bus may carry data, power, clock pulses, and other related signals.

Some common bus designs are ISA, EISA, MCA; SCSI, and S-100.

2. The expansion slots located on most motherboards.

BYTE

A sequence of bits taken as a unit. Usually 8 bits in length.

CMOS

1. COMPLEMENTARY METAL OXIDE SILICON or SEMICONDUCTOR. A technique for making ICs which allows faster performance and lower power consumption.

2. An IC manufactured with the above method.

CMOS SETUP

A memory area set aside on an AT-type system which holds configuration information, such as the number and type of floppies, hard drives, etc.

CONTROLLER

A circuit that interprets data from th bus to a form that the device being controlled can use.

CPU

CENTRAL PROCESSING UNIT. The main chip of a computer. In an IBM or compatible, this is usually an 8086, 8088, 80286, 80386 or 80486 (Intel 80x86 family). Other common CPU families are Z-80, 6800 and 6502.

CRC

CYCLIC REDUNDANCY CHECK or CHECKSUM. A data integrity check. Used primarily in disk drives and data transfer.

DRAM

DYNAMIC RAM. A form of RAM that utilizes 2 transistors per bit stored. Requires a periodic refresh to keep the data in storage.

DRIVE

Usually refers to a hard disk drive or floppy disk drive, but can also refer to a Bernoulli box, a tape drive, CD-ROM, etc.

EISA

EXTENDED INDUSTRY STANDARD ARCHITECTURE. A 32-bit extention specification to ISA, which maintains compatibility with "standard" expansion cards.

HARDWARE

If you can touch it, it's hardware. Computer equipment, as opposed to the programs that run on a computer.

Hz

HERTZ. A term used to describe the number of vibrations, or cycles, per second. I. e., 60 Hz = 60 vibrations per second.

I/O PORT

A "channel" of communication between the CPU and peripheral devices.

IRQ

INTERRUPT REQUEST. A signal along an ISA, EISA or MCA bus indicating that a piece of equipment needs attention.

ISA

INDUSTRY STANDARD ARCHITECTURE. The structure and signal specification for an IBM XT-compatible and/or IBM AT-compatible bus.

K

1. x 1024 (2^{10}).

2. Kilobyte.

3. x 1000 (traditional).

KILOBYTE

1024 bytes.

M

1. x 1,048,576 (2^{20}).

2. Megabyte.

3. x 1,000,000 (traditional).

MEGABYTE

1,048,576 bytes (1024 Kilobytes).

MHz

MEGAHERTZ. 1 Million Hz. Usually used to describe system speed or oscillator speed. See Hz.

MICROSECOND (µS or uS)

One millionth of a second.

MILLISECOND (mS)

One thousandth of a second.

MOTHERBOARD

The main board of a computer. The motherboard usually contains the CPU, the BIOS chips, RAM, and expansion slots. Other configurations do exist.

NANOSECOND (nS)

One billionth of a second

OS

OPERATING SYSTEM. The interface between the computer proper and the user (you). The common microcomputer OSs of today are DOS, OS/2, Macintosh OS and Unix/Xenix.

PARITY

A system of error checking based on the total number of bits in a byte, plus another bit called the parity bit. For example, in an even parity scheme, the total number of bits set to ON, or 1, must be even.

PORT

Short for I/O PORT.

POST

POWER ON SELF TEST. A set of diagnostic procedures that a system runs on itself every time it is booted.

RAM

RANDOM ACCESS MEMORY. Memory that does not have to be read serially, or from front to back, but can be accessed at any point or in any order desired.

RESET

A signal sent to the CPU which causes the computer to start itself over, or to boot again.

RESPONSE TIME

The time it takes to read from or write to a device. Usually associated with RAM chips and measured in NANOSECONDS.

ROM

READ ONLY MEMORY. Memory that cannot be written to. This usually contains the BIOS.

WARM BOOT

Reseting the computer without going through the POST. Usually done via <Ctrl><Alt><Del>.

MOD-MUP

Revision 2.0

4/1/91 VK