Table of Contents

1. Introduction

This manual describes the operation of B&K Precision's ***Model 846 and Model 847*** *software driven device programmer. The information contained in this manual has been reviewed for accuracy, clarity, and completeness.*

Please report, in writing, any errors to:

B&K PRECISION

1031 Segovia Circle,

Placentia, CA 92870

TEL: (714) 237-9220

FAX: (714) 237-9214

WEB: www.bkprecision.com

B&K Precision reserves the right to use and distribute any information supplied without obligation.

ABOUT THIS MANUAL

The **Model 846/Model 847** User Guide explains how to install and run the programming software on your computer.

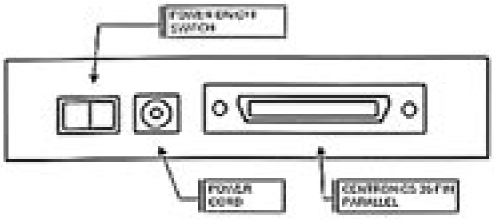
- Chapter 2 contains instructions for installing and running the **Model 846(847)**.
- Chapter 3 describes the basic operating procedures of the **Model 846(847)**.
- Chapter 4 is organized by main operating commands and gives detail instructions on each command including macro files and batch mode operation.
- Chapter 5 provides troubleshooting information for identifying and solving problems with your **Model 846(847)**. The detailed guidance for B&K Precision's technical support and return material procedures are described in this chapter.

This Manual assumes that you have a working knowledge of your personal computer and its operating conventions.

GENERAL DESCRIPTION

Model 846(847) is a software driven device programmer that supports a wide variety of programmable devices including: EPROM, EEPROM, Serial PROM, EPLD, PEEL, GAL, FPGA, and single chip microcontroller.

Model 846(847) easily connects to the parallel printer port of any IBM PC, and can operate with a full spectrum of IBM compatibles: PC XT, AT, 386, 486, Pentium, PS/2, portable(laptop), and clone computers.



The great advantage of Model 846(847) is the programming speed and the superior software. **Model 846(847)** is controlled via a host IBM PC computer. The operating software is user-friendly interface that includes window type pull-down menus, macro facility for batch file execution, and virtual memory management to deal with very large files.

The main capabilities of Model 846(847) with software installed on IBM PC are:

- selection of a device type and a manufacturer of source or target device,
- loading and saving data files in Binary, Intel Hex, Motorola S, Tektronix Hex, and JEDEC formats,
- distribution of 32- and 16- bit data into 8- bit portions,
- maintenance of file, device, and buffer offset addresses,
- blank checking, reading, writing, and verification of devices,
- security fuse blowing for PLD devices,
- provides parallel test vector application,
- Auto Device Selection of E(E)PROM and Flash memory devices.

- running DOS command without quitting Model 846(847) control program.

BASIC TECHNICAL SPECIFICATION:

Dimension:

Module: 8.5"(L) x 5"(W) x 1.5"(H) (Perfect size for briefcase)

Material: Heavy-duty Aluminum

Weight: 4.0 lb., 4.5 lb.(Model 847)

Interface:

Model 846 uses a standard parallel port (Lpt1, Lpt2, Lpt3)

Socket:

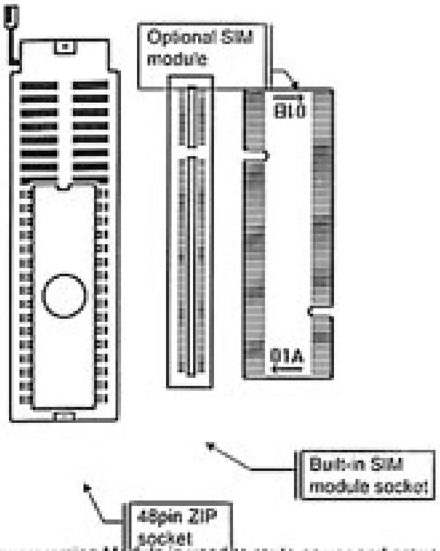
Model 846 - Textool 48 pin ZIF(300/600 mil) standard.
Model 847 - Textool 48 & three of 32 pin ZIF(300/600)

Optional adapter:

Over 50 different converters are available for special-package devices such as PLCC, SOP, TSOP, QFP and SDIP.

Optional module:

The **"PROGRAMMING MODULE"** allows for the re-routing of VCC, VPP, and group pins when changing to different device groups. The **"PROGRAMMING MODULES"** can be easily inserted on the SIM socket. The concept of "PROGRAMMING MODULES" provides support for a wide variety of devices and each module supports two groups of devices. B&K Precision will be able to provide future support, as new devices become available, for a minimal cost to the customer.



The Programming Module is used to route power and ground signals directly from pin drivers inside the Model 846(847) to the device. When you select a device, the correct module number for that device will show next to the device part number. The Programming Module MUST be installed correctly before any device programming, reading, or verifying can take place. Your Model 846/847 should include two Programming Modules, labeled 01A/B and 02A/B. If any of the two modules are not present in the shipped package, please let us know

Power Supply :

UL listed, CSA certified

Input: 120V 60 Hz 40W (USA) or
220V - 240V, 50/60 Hz 40W(EUROPE)

Output: AC 12V 2A

SYSTEM REQUIREMENT

TYPE: IBM PC XT, AT, 386, 486, Pentium, PS/2, Portable(notebook) or compatibles.

A hard disk drive is recommended for software installation.

RAM size: 512K

Host computer I/O : Either one of standard parallel ports.

OS: MS-DOS 2.1 or greater

Model 846 package contains:

- 48 pin ZIF socket programming module
- AC power transformer
  - Input: 110V or 240V
  - Output: 1.8AMP 12-16V AC
- #01 and #02 "PROGRAMMING MODULES"
- 6 feet long parallel printer cable.
- Installation diskette and manual.
- Registration and warranty card.

Model 847 package contains:

- Programming module(One-48 pin & three-32 pin ZIF sockets)
- AC power transformer
  - Input: 110V or 240V
  - Output: 1.8AMP 12-16V AC

- #1 and #2 "PROGRAMMING MODULES"
- 6 feet long parallel printer cable.
- Installation diskette and manual.
- Registration and warranty card.

NOTE: Be sure to complete and return the enclosed registration and warranty card so that we can continue to provide you with updated software, technical support, and new programmer developments.

TERMS AND SYMBOLS USED IN THE GUIDE

SAFETY NOTE CONVENTIONS

NOTE *assists the user in performing a task. It makes the job more easily understood.*

CAUTION *alerts the user that unexpected results or damages to a device may occur if an instruction is not followed.*

OTHER TERMS AND DEFINITIONS ARE AS FOLLOWS:

- Most operations are clearly indicated in bold print.
- **8467.EXE** is the main executable file for both Model 846 and Model 847 software that allows a user to read, program, and verify a device. From a DOS prompt, type 8467<Enter> to execute *8467.EXE* file.
- User Input Keystrokes are not case-sensitive. The software can accept either upper or lower-case characters.
- **Device** : The IC you are attempting to read, program, or verify.
- **Buffer** : The work area in your computer memory to execute Read, Save, Program, and Verify. The Buffer size may be from 64K to 16 MegaBytes.

Note: If the size of a device is bigger than the buffer size in your computer, 8467.EXE will use the hard disk space(swapping). For this reason, the Model 846 software can handle devices up to 16MBit E(E)PROMs with your standard memory space (minimum 512KB RAM memory is required).

- **<Enter>** is shown as the **ENTER** key.
- **ESC** refers to the **ESCAPE** key.

- ^X means pressing X key while **holding Control key.**
- ALT-X means pressing X key while **holding ALT key.**
- **Inserting a DIP Device**

Pin 1 on a DIP package is generally indicated by a notch on one end of the device. Pin 1 is also indicated by a stamped or recessed dot on one corner of the device.

In the illustration below, the notch and the dot are highlighted by arrows. The dot is on the left and the notch is on the right.

To Insert a DIP device into the ZIF socket:

CAUTION: Devices are static sensitive. Operate your programmer at an antistatic workstation. To avoid electric shock and damage to the devices, use an antistatic wrist strap containing a 1 mega-ohm (minimum) to 10 mega-ohm (maximum) isolating resistor.

1. Lift the socket lever to the open position.

2. With the notched end facing the top of the socket, place the device in the socket so that the bottom of the device is aligned with the bottom of the socket (bottom justified).

3. Lower the socket lever to lock the device into the socket.

4. When you done, select the device you just inserted.

- **Inserting a PLCC Device**

Note: In order to use PLCC devices, your programmer must have a proper adapter for the device. Consult your programmer's adapter list for the non-standard device.

Pin 1 on a PLCC package is generally indicated by a notch on one corner of the device. Pin 1 is also indicated by a stamped or recessed dot on one side of the device.

In the illustration below, the notched corner and the dot are highlighted by arrows. The notch is on the left and the dot is on the right.

To insert a PLCC device into an adapter:

CAUTION: Devices are static sensitive. Operate your programmer at an antistatic workstation. To avoid electric shock and damage to the devices, use an antistatic wrist strap containing a 1 mega-ohm (minimum) to 10 mega-ohm (maximum) isolating resistor.

1. Open the adapter 90 degrees, set its front edge under the two tabs at the front of the base opening, and lower its back edge into place.

2. Orient the device you want to use so that pin 1 is next to the retaining latch. Each Adapter has a small molded dot that represents pin 1 and a notched corner that can be used to align chamfered corners of devices.

3. Insert the device into the open adapter.

4. Close the Adapter, and press the retaining latch forward with your thumb until the latch snaps into place.

5. If you have not already, select the device you just inserted.

2. Getting Started / Installation

INSTALLATION REQUIREMENTS

Model 846 and Model 847 are designed to be operative with any IBM-PC XT/AT/386, and 486 based compatible computer running PC-DOS or MS-DOS 2.1 or greater. The Minimum of 512K bytes of RAM memory is required to program any device size and **NO EMS** memory is used for larger devices. The computer requires a double-sided disk drive, but a hard disk drive is also recommended

HARDWARE INSTALLATION

The following section details the procedure for accomplishing the hardware installation procedure.

INSTALLATION PROCEDURE

Model 846(847) easily connects to any of the parallel printer port in your computer. There are three different addresses for the parallel port. When you select an address from **LPT1, LPT2, LPT3**, one of them should be valid without a communication error message. Be sure that **Model 846(847)** recognizes your computer's parallel port address when you execute 8467.EXE file.

SOFTWARE INSTALLATION

The Model 846 software is distributed on a single 1.44M diskette. The diskette contains following files:

INSTALL.COM

BK8467.EXE

Using the COPY command of the DOS, make a backup copy of the **Model 846(847)** software. The software is not copy protected.

TO INSTALL MODEL 846 SOFTWARE ON YOUR HARD DISK:

1. Turn on the computer and get to a DOS prompt (usually C:\>).
2. Insert the installation diskette into floppy disk drive A and type:

   A: <Enter>

INSTALL <Enter>

This starts the install program from the source drive A to the destination drive C.

The **Model 846(847)** directory will contain following files:

8467.EXE	*Main Executable file*
8467.DAT	*Device Library file, Help File*

NOTE: 8467. CFG file will be generated and changed after executing and quitting the 8467.EXE.

STARTING Model 846

TO START MODEL 846:

1. When installation is complete, the DOS prompt will automatically be located in the **Model 846(847)** directory that is defined during the installation.
2. To start **Model 846(847)**, type:

   8467 <Enter>

NOTE: In order to run the Model 846 from any directory, you can add a PATH variable in the AUTOEXEC.BAT file as shown below:

PATH=C:\DOS;C:\BK8467(or path name of Model 846)

Therefore, whenever the AUTOEXEC batch file is invoked, you may type only 8467 to run the Model 846.

TROUBLE SHOOTING IN INSTALLATION

A communication error may occur on the screen if the hardware / software is not correctly installed.

BE SURE THE FOLLOWING STEPS ARE CHECKED:

- Make sure that the **Model 846(847)** is connected to your PC printer port directly. It will not work with multiple port connector.
- Be sure your printer cable is directly & firmly connected to your computer and **Model 846(847)** programmer.

- Plug in the AC power cord to your **Model 846(847)** and turn on the switch.

3. Quick Start Examples

If you are using the **Model 846(847)** for the first time, this section will help you to become familiar with the basic operating procedure. This section includes two examples of device programming with your **Model 846(847).**

PROGRAMMING AN EPROM

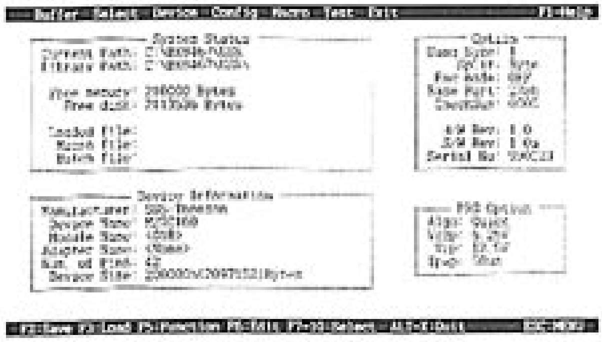
We selected an AMD 27C010 EPROM to show you how to program an EPROM. The 27C010 EPROM needs to be erased (blank) before this procedure commences.

NOTE: EPROMs has a quartz window which can be erased by exposing the EPROMs to ULTRAVIOLET(UV) light. Erasing an EPROM usually takes 10-30 minutes.

1. From your computer keyboard, type:

   CD\BK8467<Enter>
   8467   <Enter>

The Main Screen appears:

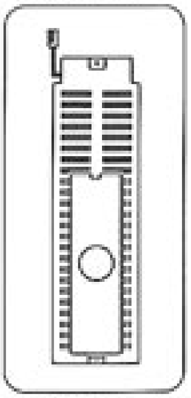


```
 Buffer  Select  Device  Config  Macro  Test  Exit                    F1-Help

     ------------ System Status ------------------          ---- Option ----
     Current Path: C:\BK8467\[illegible]                    Data Type: [illegible]
     Library Path: C:\BK8467\[illegible]                        Split: Byte
                                                            Bus Width: [illegible]
      Free memory: [illegible] Bytes                        Base Port: [illegible]
        Free disk: [illegible] Bytes                         Checksum: [illegible]

      Loaded file:                                            H/W Rev: 1.0
       Macro file:                                            S/W Rev: 1.0a
       Batch file:                                          Serial No: [illegible]

     ------------ Device Information -------------          -- PRG Option --
     Manufacturer: SGS-Thomson                              Algo: Quick
      Device Name: M27C160                                  Vcc: [illegible]
      Module Name: <DIP>                                    Vpp: [illegible]
     Adapter Name: <None>                                   Speed: [illegible]
     Num. of Pins: 42
      Device Size: 200000h(2097152)Bytes

 F2-Save  F3-Load  F5-Function  F6-Edit  F7-10-Select  ALT-X-Exit        [illegible]
```

2. Check the optional configuration before programming is commenced.

3. Select device. There are two different ways to select the target device from the menu. The device manufacturer and type can be chosen by using the arrow keys. You may also type the manufacturer and the device names on the Device Select line and *<Enter>*. After selecting the device, the DEVICE INFORMATION window can provide detail device information. Press F4 key for this information.
4. Load your file from floppy or hard disk drive into the buffer. Make sure that the file is located in your current directory; otherwise, you have to enter the directory path in the name entry box. Now, you may examine the contents of your buffer.
5. Select DEVICE FUNCTION menu in order to begin programming.
6. Now insert the 27C010 device into the ZIF socket. After inserting the part, make sure the socket handle is down(close) to secure the chip.

See the illustration below:



7. Move the highlighted cursor to BLANK CHECK, press *<Enter>*

NOTE: If an EPROM is not erased completely, it will not pass the Blank Check. If an EPROM is damaged to begin with, it may not pass the blank check, although it has been erased for a long time.

8. Select PROGRAM entry, press *<Enter>*

CAUTION: Do not touch the device while the busy LED is on (During programming).

After programming a device, the part is automatically verified and the CheckSum is calculated. Check the CheckSum for the device in the OPTION menu. In order to verify your work, read the programmed part. The CheckSum value of the device appears on your screen. If you have the same CheckSum value, the 27C010 is programmed successfully.

DUPLICATING AN EPROM

The following is an instruction on duplicating a programmed device. In order to do so a source device and an **erased(blank)** target device are necessary.

Source device : AMD 27C256 (is programmed)

Target Device : INTEL 27C256 (is erased)

1. Make sure the Model 846(847) main menu is displaying without any communication error(refer to programming section).
2. Place the AMD 27C256 device into the ZIF socket.
3. Select the manufacturer and part names from the SELECT menu.
4. Read the source device. In order to make sure the device is read properly, select and execute the VERIFY function.
5. Remove the current chip from the socket and replace it with the blank INTEL 27C256 device. Select the appropriate device from SELECT menu on screen.

Note: You do not need to change the device information if you use the exact same chip as the source device.

6. Select Blank Check routine.
7. Program the target device. The part will be programmed and verified automatically. If no error messages appear during programming time, your duplicating work is a success. You will have a duplicated INTEL 27C256 part from AMD 27C256 chip.

4. Operations

This section describes the operations of the software. The standard system-menu is divided into five display areas: System Status, Device Information, Device address, Buffer address, and Option.

MENU SELECTION AND FUNCTION KEY

There are three ways to select commands.

- Arrow keys
- Highlighted capital letter (Hot key)
- Function keys - activates a command almost anywhere in the screen.

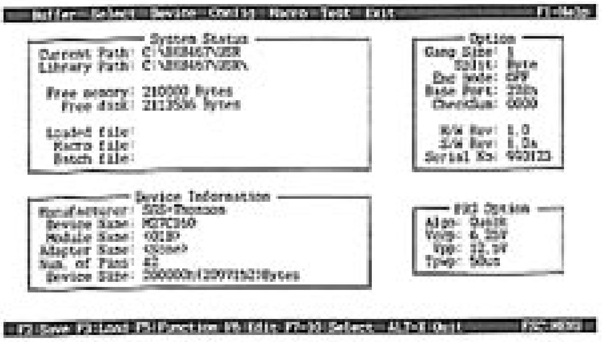
LIST OF FUNCTION KEYS

F1	Help
F2	Save
F3	Load
F4	Details of Device Information(activates in the SELECT DEVICE TYPE menu only)
F5	Function
F6	Buffer Edit
F7	Device Select (for E(E)PROM)
F8	Device Select (for PLD)
F9	Device Select (for Microcontroller)
F10	Auto device select

HOT-KEY COMBINATION

ALT-X	Quit to DOS
ALT-Q	End of Macro record
ALT-F9	Change to OS Shell
^V	Vector pattern edit(PLD only)
^F6	UES edit(GAL only)
^R	Start of Macro Record
^S	Save Macro
^B	Start of Batch Record
^L	Load macro

The main screen appears as following:



```
 Buffer  Select  Device  Config  Macro  Test  Exit                F1-Help

 ─────────── System Status ───────────────────       ───── Option ─────
  Current Path: C:\[illegible]                        Gang Size: 1
  Library Path: C:\[illegible]                            Split: Byte
                                                      [illegible] Mode: OFF
   Free memory: 210000 Bytes                          Base Port: 278h
     Free disk: 2113536 Bytes                          CheckSum: 0000

   Loaded file:                                          H/W Rev: 1.0
    Macro file:                                          S/W Rev: 1.0a
    Batch file:                                        Serial No: 993123

 ─────────── Device Information ──────────────       ─── PGM Option ───
  Manufacturer: SGS-Thomson                           Algo: [illegible]
   Device Name: M27C160                               Vcc: 6.25V
   Module Name: <0LB>                                 Vpp: 12.5V
  Adapter Name: <None>                                Tpw: 50us
  Num. of Pins: 42
   Device Size: 200000h(2097152)Bytes

 F2-Save F3-Load F5-Function F6-Edit F7-10-Select ALT-X-Quit    [illegible]
```

SYSTEM STATUS

The system status display area monitors the status of your computer peripherals after you run the Model 846(847) software.

Current Path	Current location of Model 846(847) working directory
Library Path	Current location of Model 846(847) library file directory
Free memory	Available memory size after executing Model 846(847)
Free Disk	Check the free disk space for a big size E(E)PROM programming.
Loaded File	Current file name in the buffer after loading the file
Macro File	Current Macro file name in Macro buffer
Batch File	Current Batch file name for MACRO command

DEVICE INFORMATION

The Device Information display area presents device information of the selected device. The complete device information is displayed by pressing F4 key at the specific device manufacturer and type during selecting a device.

Manufacture	Manufacturer name of the current device
Device Name	The current device number
Module Name	The SIM Programming Module Number
Adapter Name	Optional Adapter Name for Non-standard devices
Num. of Pins	Number of device pin
Chip Size	The size of device in Hex value. It is calculated by (End - Start + 1) in device address.

Device	Device Address
2716	0 - 7FF
2732	0 - FFF
2764	0 - 1FFF
27128	0 - 3FFF
27256	0 - 7FFF
27512	0 - FFFF
27010/1024	0 - 1FFFF
27020/2048	0 - 3FFFF
27040/4096	0 - 7FFFF

OPTION

The Option display area presents optional information for the programming environment.

Gang Size	Current socket size when Model 846 is used
Split	Current word format for split programming
Enc Mode	Enable or Disable Encryption mode for microcontrollers
Base Port	Current parallel port address
CheckSum	CheckSum number of the data in current buffer
H/W Rev	Hardware revision number for your programmer
S/W Rev	Current Model 846(847) software revision number
Serial No	Serial number of Model 846(847) hardware

NOTE: It is important to notify B&K Precision's tech support staff of the H/W Rev, S/W Rev, and Serial numbers when you contact them.

PRG OPTION <FOR NON PLD DEVICES ONLY>

The PRG Option display area presents programming information of the selected devices.

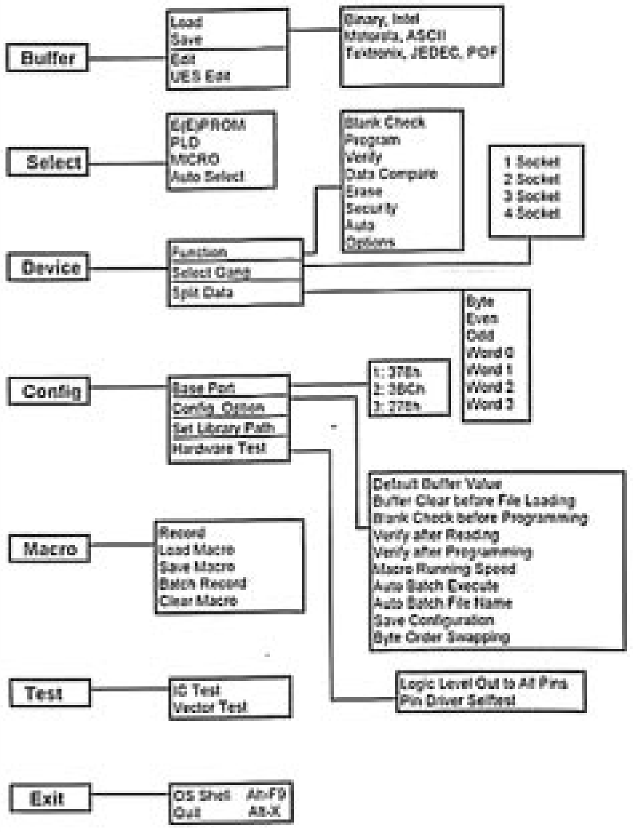
Algo	Programming Algorithm
Vccp	Main Power Supply Voltage
Vpp	Programming Power Supply Voltage
Tpwp	Programming Pulse Width

The Option display area presents optional information for the programming environment.

Model 846 MAIN MENU AND COMMANDS

The main menu consists of six sub menus: Buffer, Select, Device, Config, Macro, Test, and Exit. See the table of Model 846(847) Main Menu Structure for a more detailed description.

MODEL 846/MODEL 847 SOFTWARE MENU STRUCTURE



DATA FILE FORMATS

Model 846(847) uses four different file types: BINARY, INTEL HEX, MOTOROLA S HEX, and TEKTRONIX HEX.

INTEL HEX FORMAT

Intel Hex format files are text files which include the file information in hexadecimal.

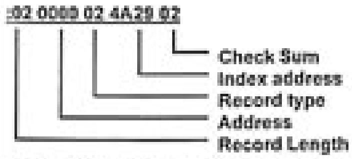
Position(byte)	Character	Remarks
1	:	A record mark
2 - 3	Byte	Record length in 2 digit Hex, Max 20 (64 in ASCII)
4 - 7	Address	4 digit Hex Field. Most significant byte first
8 - 9	Byte	2 digit field record type: 01 End of file 02 Extended address
10 - N	Data	Data field in Hex digits
N+1 - N+2	CheckSum	Two digit hex CheckSum character computed by two's complementing the sum of previous bytes except the ':'

EXAMPLE

:110000000444154414D414E2053332053455249414C73

:00000001FF

The extended address record specifies the index address that data will be loaded into. The Extended Address will continue to offset data record address until a new Extended Address record is specified.



- The Address field is blank because this record is not data.
- The record length is '02' for index address (2 Bytes).

NOTE: If the address for the data record is '2B56', the actual address will be 4A290 + 2B56 or 4CDE6(hex).

MOTOROLA S HEX FORMAT

The Motorola S format file is an ASCII-Hex file.

Position(Byte)	Character	Remarks
1	S	Letter S indicates start of record
2	0, 1, 2, 3, or 9	A single character indicates the type of record . 9: End-of-file 3: 32-bit address data record 2: 24-bit address data record 1: 16-bit address data record 0: Header
3-4	Byte	Byte COUNT in hex of rest-of-record(multiply by two for number of characters). This count includes the address, data, and CheckSum field.
5-X	Bytes	Memory Address for the current record. X will be: 8: 16-bit addressing for files less than 64K. 10: 24-bit addressing for files greater than 64K. 12: 32-bit addressing for files greater than 64K in length.
X+1-N	Bytes	Hex Data (two per byte)
N+1 - N+2	CheckSum	Two digit hex CheckSum character calculated by one's complement of Sum of DATA, ADDRESS and COUNT.

EXAMPLE

S1140000444154414D414E2053332053455249414C6F

S9030000FC

TEKTRONIX HEX FORMAT

The Tektronix hex format contains ASCII records, expressing bytes ASCII pairs.

Position	Character	Remarks
1	/	Slash character for start of line
2 - 5	2Bytes	Address, MSB first load
6-7	Byte	Number of data bytes (not checksums)
8-9	Byte	CheckSum of ADDRESS and COUNT by character in hex (not by byte)
10-N	Data	Data bytes as ASCII pairs
N+1-N+2	Byte	CheckSum of Data by character (not as bytes)

EXAMPLE

/00001102444154414D414E20533320534552494 14C8F

/01000001

BINARY FORMAT

Binary format does not specify the address or CheckSum of the file. The file contains the actual binary data. An example of this format is a DOS executable file with an .EXE or .COM extension. Binary format is generated for programmable memory devices. It is recommended to save your EPROM data as binary format in order to load the file as a standard file format later.

ASCII FORMAT

This selection generates an ASCII coded hex format for either 4-bit or 8-bit PROMs. Each record contains a four-digit hex address(16-bit) followed by 16 data elements. A 16-bit checksum is at the end of the file.

When this format is selected, the device base address must be specified. This address represents the lowest address in the device. The file created contains an entry for each location in this device. ASCII hex format can be created for programmable memory devices only.

JEDEC STANDARD < PLD DEVICES ONLY>

JEDEC(Joint Electronic Device Engineering Council) files are the standard method of describing PLD fuse patterns and test vectors. JEDEC files contain fuse data, test vectors, part numbers, and checksums. The checksum of file allows you to verify that a given file is intact and has not been unintentionally modified. JEDEC files normally use the extension(last 3 letters) ".JED."

Following is an example of a JEDEC file:

```
<STX>File for PLD 15S8 Created on 11-SEP-96 5:08PM
2754 memory decode 345-432-123
Seung Park PK Logic corp.
QP20* QF448* QV8*
F0*X0*
L00001111101111111111111111111111*
L00281011111111111111111111111111*
L00561110111111111111111111111111*
```

```
L01120101101101111011111111111111*
L02240111101110111101111111111111*
L03360101011101110111111111111111*
V0001000000XXXNXXXHHHLXXN*
V0002010000XXXNXXXHHHLXXN*
V0003100000XXXNXXXHHHLXXN*
V0004110000XXXNXXXHHHLXXN*
V0005111000XXXNXXXHLHHXXN*
V0006111010XXXNXXXHLHHXXN*
V0007111100XXXNXXXHHLHXXN*
V0008111110XXXNXXXLHHLXXN*
C124E*<ETX>8646
```

STX The fuse map begins with an ASCII STX character(02 hex)

Design Specification This item is user specified. While no format rules apply, certain information, such as user's name, company, design date, part designation, revision and device part number, should be entered This field is terminated by an asterisk(*).

QP This field specifies the number of pins in the devices.

QF This field specifies the number of JEDEC fuses in the devices.

L The fuse list fields contain the state of all fuse links in the devices. The starting fuse number follows the L specifying the field type. The fuse list that follows contains a zero(0) for each intact link and a one(1) for each blown link. An L field is generated for each product term in the device.

C The checksum field contains the 16-bit sum of the link states of the 8-bit words.

ETX The fuse map ends with an ASCII ETX character(03 hex).

Sum Check A 16-bit sum of the ASCII values of the characters from STX to ETX inclusive. The sum check follows the ETX.

For more information on the JEDEC standard, contact:

Global Engineering Documents Inc. at (800)854-7179

Electronic Industries Association at (202)457-4900.

LOGIC COMPILERS FOR PLD DEVICES

Software is available to help the engineer develop designs using PLDs. Software Tools called logic assemblers or compilers translates a design file written in high-level language into a fuse pattern stored in a JEDEC file. JEDEC files are produced by almost all PLD development software and are accepted by **Model 846(847)** programmer.

There are many commercial software package available to help you design with PLDs.

Compiler Name	Maker	Contact
PALASM	AMD	800-538-5450
CUPL	Logical Devices Inc.	800-331-7766
OPAL	NS	408-734-8184
SNAP	SIGNETICS	800-234-7381
Max+Plus II	Altera	800-800-EPLD

BUFFER

NOTE: In order to handle your data properly, use the **File Offset** and **Buffer Start Address** when you use **BUFFER** commands. Press **<TAB>** key in **FUNCTION** when you need to change the device or buffer address before programming.

File Offset is subtracted from addresses of the file downloaded to the programmer. For example, if you set File Offset to 1000h, then the downloaded data minus 1000h would be placed into the buffer at the address specified by the **Buffer Start Address.**

Buffer Start Address is the address in the buffer where you want your downloaded data to start. For example, if you set Buffer Start Address to 800h, then the downloaded data only appears in the buffer beginning at address 800h.

BUFFER / LOAD

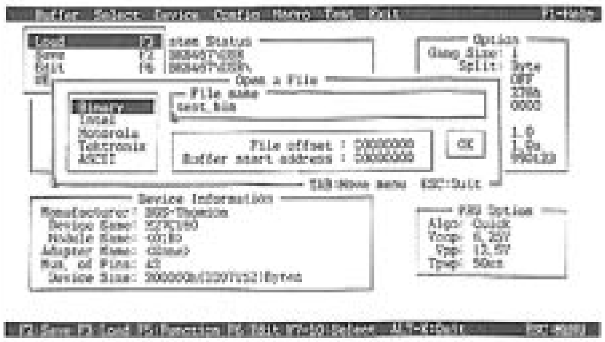
Data can be loaded into memory from a device or by opening a data file. Load fills your buffer memory with the data from storage for viewing or editing. This command loads the data from the selected file storage into the memory buffer. The file must be one of the file formats supported by **Model 846(847)**, such as **INTEL HEX**, **MOTOROLA S**, **TEKHEX**, or Binary formats. You can type the path which includes the drive, directory, and filename to load the source file. It accepts DOS wildcard such as "*.*".

To select one of the files mentioned above, choose the file by moving the cursor up or down and press *<Enter>*. When you have selected the desired file, press *<Enter>* again to load the file into the data buffer. If you are programming a PLD, you will want to load a JEDEC file. The procedure is identical to loading a data file, except the files in the current directory will have the JED extension. If your selected device is Altera MAX family then the file you should load is POF extension.

The **Model 846(847)** uses a RAM buffer to hold data. After loading a file into the buffer, you can edit buffer data. If you load a JEDEC file, you may use (the vector pattern edit) command to view or edit the fuse map and (test/vectors) for any test vectors that may have been in the JEDEC file.

EXAMPLE

*C:\BK8467\file\2764.hex or C:\BK8467\*.**



BUFFER / SAVE

Save the current data in your memory buffer to a disk storage by using one of the currently supported file formats.

Before saving a file, check the buffer and the file address ranges. The contents of the buffer through the specified range will be written into the new file, completely erasing any existing file with the same name. Before saving to disk, make sure that no file with the same name exists.

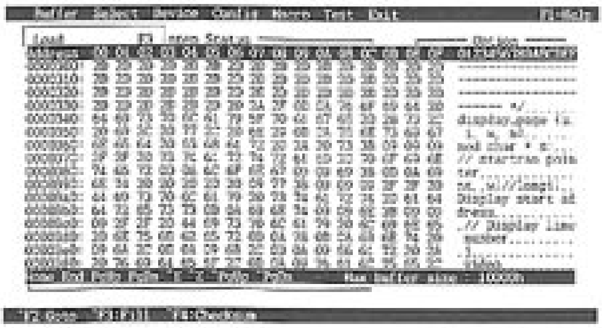
BUFFER / EDIT

Change the content of your current buffer. This command allows the user to examine and modify the contents of the memory buffer.

This section applies to a non-JEDEC file (PROM, EPROM, EEPROM, Microcontroller) or to a memory chip. If PLD is being loaded, see the (vector pattern edit) section. The data is presented in hex and ASCII formats. A highlighted cursor appears on the active side at the character that can be edited. The tab key allows you to move current cursor to either hex data area or ASCII data location. Changes are accepted as the hex characters: 0-9, 'A' to 'F', and 'a' to 'f '.

Data editing features can be invoked by using the F6 function key.

The [Pg Up] and [Pg Dn] keys allow the previous or next set of data bytes to be displayed. The [home] and [end] keys move the cursor to the beginning or end of the device pattern.



COMMAND LIST OF EDITOR

(^X means pressing X key while holding Control key)

^E	Scroll up
^Z	Scroll down
^PgUp	Go to the top of the file
^PgDn	Go to the end of the file
^F2	Goto new address
^F3	Fill the buffer with Default buffer value in config menu
^F4	Check sum
TAB	Switch between ASCII and HEX editor

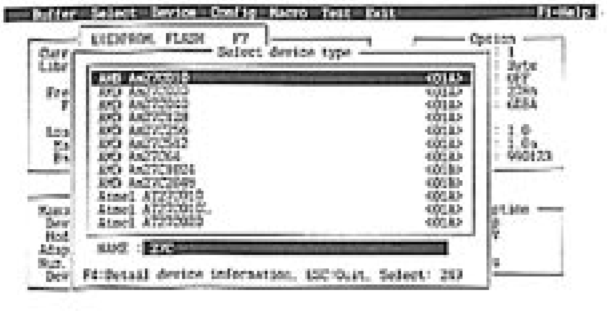
BUFFER / UES EDIT

The UES Edit command creates or changes the User's Electronic Signature(UES) array in GAL device. Each GAL device contains an electronic signature word consisting of 64 bits of reprogrammable memory. The electronic signature word can be programmed to contain any identification information desired by the user. Some uses include pattern identification labels, version numbers, dates, inventory control information, etc. These features give the user the ability to view and edit the UES data before programming a GAL device.

When the UES edit command is invoked, an editing data window appears. If the data fields are empty, you may create a new UES. You can enter the UES up to eight characters in the Hex or ASCII data area. If you see any data from the current UES window, it means the UES has been created and that you can modify the data for a different reason. The UES data is not secured when you execute the Function / Security command.

SELECT

During operation, the first step is usually to select a device. This Select command enables the user to define the manufacturer and the type of the device that will be used. After you select a device, you can insert a device into the programmer's device socket and conduct various device operations such as program and verify device data or read data from the device. The SELECT command contains manual and automatic methods of selecting device. If your device is not identified by the Auto Device Select menu, you can select the device list displayed in the Manufacturer & Device list. Scroll through the manufacturers and device numbers until you find the manufacturer and device you are looking for. You can use wildcards to help you "zoom" on the device you are looking for.



E(E)PROM, FLASH F7

Select device type

AMD Am27C010 <01A>
AMD [illegible] <01A>
AMD [illegible] <01A>
AMD Am27C128 <01A>
AMD Am27C256 <01A>
AMD Am27C512 <01A>
AMD Am27C64 <01A>
AMD [illegible] <01B>
AMD [illegible] <01B>
Atmel [illegible] <01A>
Atmel [illegible] <01A>
Atmel [illegible] <01A>

NAME : 27C

F4:Detail device information, ESC:Quit, Select: 2??

SELECT / E(E)PROM, FLASH

All EPROMs(27xxx), EEPROMs(28xxx), Serial E(E)PROMs(17xxx, 24xxx, 32xxx, 33xxx, 35xxx, 59xxx), and Flash EPROM(29xxx) of 24/28/32/40/42 and up to 48 pins (1 M bit, 2M bit, 4M bit, 8M bit, 16M bit, and up)

SELECT / PLD

EPLD, EEPLD, FPL, PEEL, GAL

SELECT / MICRO CONTROLLER

Intel 87xx..., Signetics 87C75x; Microchip PIC16C5x,17C4x;

Motorola 68HC705/711xx/; Zilog Z86Exx; NEC 8749H..

SELECT / AUTO DEVICE SELECT

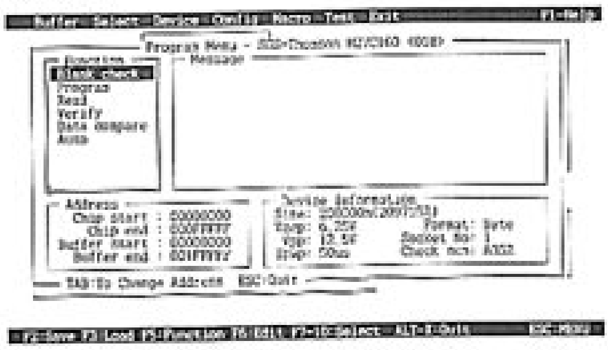
Identify the device that is mounted on the ZIF socket. This command can only be applied to memory and some microcontroller devices. Executing the command will enable the programmer to identify the ID on the device and will select the matching device in the library automatically.

Note: If you have "Device not found" message, manually select the device. If you have old devices or defective chip, Model 846(847) will not recognize the ID code from your device.

DEVICE

DEVICE / FUNCTION

This section presents main operation menu for the target device that is mounted on the ZIF socket. In order to process the following commands, make sure the device is correctly inserted into the ZIF socket and the latch is down.



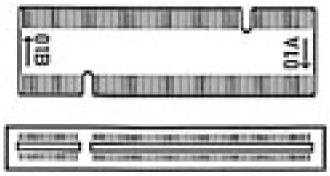
```
 Buffer  Select  Device  Config  Macro  Test  Exit                    F1-Help
         ┌ Program Menu - 2532>Thomson M27C160 <008> ──────────────────────┐
 ┌ Function ┐ ┌ Message ───────────────────────────────────────────────┐
 │Blank check│ │                                                        │
 │Program    │ │                                                        │
 │Read       │ │                                                        │
 │Verify     │ │                                                        │
 │Data compare│ │                                                       │
 │Auto       │ │                                                        │
 └───────────┘ └────────────────────────────────────────────────────────┘
 ┌ Address ─────────────────┐ ┌ Device Information ──────────────────┐
 │   Chip start : 00000000  │ │Size: 200000h(2097152)                 │
 │     Chip end : 000FFFFF  │ │Vcc: 6.25V           Format: Byte      │
 │ Buffer start : 00000000  │ │ Vpp: 12.5V       Socket No: 1         │
 │   Buffer end : 001FFFFF  │ │Prog: 50us        Check sum: A332      │
 └──────────────────────────┘ └───────────────────────────────────────┘
 ── TAB:To Change Address  ESC:Quit ──
 F2:Save F3:Load F5:Function F6:Edit F7-F10:Select ALT-X:Quit          ESC:Menu
```

Note: *<PROGRAMMING MODULE>*

The Programming Module is used to route power and ground directly from pin drivers inside the Model 846 to the device. When you select a device, the correct module number for that device will show next to the device part number.

<MOTOROLA MICROCONTROLLERS>

The window of windowed devices must be covered with an opaque label at all time during any operations.



*The Programming Module MUST be installed correctly before any device programming, reading, or verifying can take place. Your **Model 846/847** should include two Programming Modules, labeled 01A/B and 02A/B. If any of the two modules are not present in the shipped package, please let us know.*

DEVICE / FUNCTION / BLANK CHECK

The Blank Check function is used to verify whether or not a device is in an erased or unprogrammed state.

All EPROM(Erasable Programmable Read Only Memory) devices should be checked before programming. EEPROM(Electrical Erasable Programmable Read Only Memory) based parts do not need this command because EEPROM is erased automatically before programming.

PLD based parts are checked by verifying all of the fuses that are intact. Any erased PLD should pass this test.

NOTE: Erasing EPROMs. In order to clear data in an EPROM, the chip should be exposed to short wave UV(Ultra violet) light. Most erasers require between 5 and 30 minutes to erase an EPROM. Some types of chips take longer to erase than others. An EPROM based part(a PLD or Microcontroller) with a security bit feature is designed so that the security address is typically the last bit to be erased. If the window of a chip is not clear, try cleaning the window with alcohol or solvent. Erase chips if the chips are exposed to sunlight and fluorescent light for months or years; your chips can be erased. You should cover the window of the programmed chips with an opaque label to make the data permanent. Some EPROM based parts can't be erased because they do not have window. These chips are called one time programmable(OTP) EPROMs.

An EPROM has a quartz window located on the chip just above the die. Erasing an EPROM is done by exposing the EPROM to high-frequency ultra-violet light waves. Erasing an EPROM usually takes from 15-20 minutes, but may be shorter or longer, depending on the device. Many manufacturers make EPROM erasers. If you wish to purchase an eraser, call B&K PRECISION, (714) 237-9220. When an EPROM is not being erased, the window may be covered with an opaque label. Sometimes (over a period of years) an EPROM will start to erase due to room level fluorescent. Direct exposure to sunlight also has this effect, but happens much more rapidly.

NOTE: In order to check whether the device is blank or not, the user should read the target device and check the CheckSum. If the CheckSum is zero or buffer data is "FF", it means the device is in an erased or unprogrammed state; otherwise, the device is not erased.

DEVICE / FUNCTION / PROGRAM

Function/Program command will enable you to place a new data from memory buffer into the target device. The busy LED will be blinking during programming.

Make sure the device is correctly inserted into the ZIF socket and the latch is down. Then check the buffer device address range before you start. The values will default to the size of the device.

- Memory device

The target device must be blank checked unless the part is electrically erasable. Although most of EEPROMs and Flash Memory have the ERASE function in the menu, some EEPROMs such as AT28CXXX or AT29CXXX don't have the ERASE function. Note that EEPROMs without the ERASE function are automatically erased before programming.

- Programmable Logic Device operation

After programming is complete, verification should be performed according to the semiconductor manufacturer's specifications. In order to test vectors, a vector test should be performed (See vector test under the **TEST** menu). Finally, the part may be secured so that its content can no longer be examined or modified. The security function will not execute if the device fails to verify or pass the vector test properly.

- 28C256, 28C010, ect:

28CXXX family devices support Software Data Protection. After programming the device, the **Model 846(847)** software automatically protects the data using the software data protection routine. Therefore, in order to program the protected device, it must be unprotected before re-programming it. To protect the data, do RST protect. To obtain more information about RESET protection, please consult device's manufacture specification.

- Microchip PIC devices:

Microchip PIC series is different from other microcontrollers in that it has an EPROM area as well as a CONFIGURATION FUSE. The configuration fuse in the PIC family is used to setup **Oscillator Type, Memory Code Protection, Watchdog Timer, or Processor Mode** etc. After programming the EPROM portion, change the modes of the items listed under OPTION. Then you must program the configuration option in the OPTION menu.

Do the following procedure:

1. **Program the main memory**
2. **Set the configuration in OPTION menu**
3. **Program the configuration fuse**

You may also read the state of the Configuration Fuse under the OPTION selection.

In order to obtain more information about programming the configuration fuse, contact Microchip technology at 602-786-7200 or consult the appropriate data book.

- Serial EEPROMs:

These devices are electrically erasable, but they operate serially rather than in parallel.

- Atmel or Xilinx 17xxx:

You need to set the POLARITY FUSE with this family via the Option menu . After programming the main MEMORY, go to the OPTION menu and make the appropriate change. On the OTP(One Time Programmable) devices, the POLARITY FUSE status cannot be reversed once it has been changed. Even on some of the windowed 7xxx family devices(excluding Xilinx 17xxD/L & 17128), the POLARITY FUSE cannot be toggled. Consult device's manufacture for further instruction how to handle the Polarity FUSE.

CAUTION: Do not touch or remove a device during an operation.

DEVICE / FUNCTION / READ

Read the data in the source device mounted on the ZIF socket into the buffer for examination.

The CheckSum will be displayed on the CheckSum line. The buffer may be edited, saved to disk, or used to duplicate the chip.

CAUTION: Reading the device into the buffer destroys the buffer contents through the specified range. Make sure everything in the buffer that is needed has been saved.

PLD test vectors are not stored in a logic device. Therefore, they cannot be read. The test vector buffer will be empty after reading the PLD.

NOTE: Devices that have been secured cannot be read properly. Secured chips may appear all blank, fully programmed, or scrambled.

DEVICE / FUNCTION / VERIFY

Assures that data in the device matches data in the memory buffer. If your device has the security fuse blown, a verification error is detected. The verify operation requires that the exact data pattern or file that was used to program the device be resident in the memory buffer.

DEVICE / FUNCTION / DATA COMPARE

Compares the data in device to the data in buffer and saves any difference into the COMPARE.8467 file. When you have a verify error during the <Function/Verify> operation, the <Function/Data Compare> command will be useful. It will detect a difference between the device content and the buffer content and write the difference into the COMPARE.8467 file under the **Model 846(847)** directory. You may view the file using an editor utility software.

DEVICE / FUNCTION / ERASE

This option erases the data in your socket before programming it. This operation is valid for only limited devices such as EEPROM, Flash

Memory, GAL, PEEL devices. EPROMs that comes with a window on it should be erased by UV EPROM eraser externally *(see* NOTE "Erasing EPROMs" in page 33)

DEVICE / FUNCTION / SECURITY

Secures a PLD or a microcontroller so that their content can no longer be examined or modified. Security is confirmed when valid data can no longer be read or verified against a previously read pattern. To ensure that the security fuse has been blown, the <Function/Security> operation is preceded by a read of the device and followed by a verify.

NOTE: Usually, on a UV erasable PLD or a microcontroller, a secured device may take longer to erase because the security bit address is designed to erase last.

Securing a device prevents the programmed data pattern into the device from unauthorized access. This command appears only when the selected device supports it. Some microcontrollers and PLDs can be secured by programming a special address location. The security bit will be cleared when the device is erased. Once a device is secured, it cannot be unsecured to read, verify, or duplicate. Also the secured device is seen as a blank chip even though it is not actually blank.

DEVICE / FUNCTION / ENCRYPTION

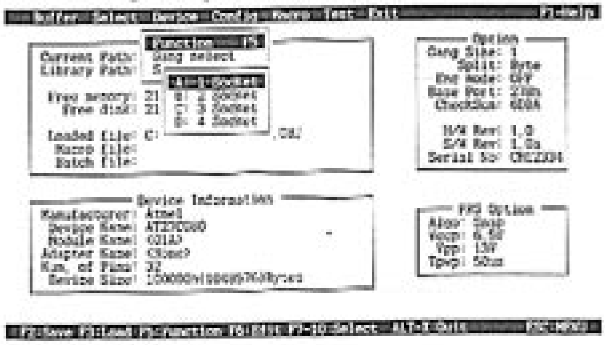
The encryption table is a feature of the 87C51/87C52 family microcontroller devices. The Encryption array of the Microcontroller is initially unprogrammed (all '1's). In order to protect the code from being easily read by anyone other than the programmer, this feature allows to program the encryption table that is exclusive NORed with the program code data as it is read out. You have to know its content in order to correctly decode the program code data. Thereafter you will have to use the same displayed encryption array any time you need to read back the device.

DEVICE / GANG SELECT

Model 847 programs multiple E(E)PROMs up to four devices at the same time. In order to program multiple E(E)PROMs you must use **Model 847**. The Model 847 is especially useful when it is necessary to program many devices with the same data simultaneously. The Model 847 is designed for multiple programming and it dr es not support set programming.

CAUTION: The 48-pin socket should be used for universal device programming. The rest of the 32-pin sockets are not supported for universal device programming. 4-socket option programming is valid for limited device programming such as E(E)PROMs and Flash PROMs. In order to program multiple devices using the 4-gang socket, you should use ICs with the same manufacturer and device type. All of the devices should be inserted into the socket with notch in the same direction as pictured on case.

EXAMPLE: Programming four E(E)PROMs as follows:



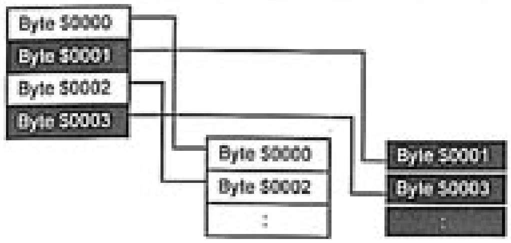
```
 Buffer  Select  Device  Config  Macro  Test  Exit                     F1-Help

                      Function       F5                        Option
  Current Path:   Gang select                           Gang Size: 1
  Library Path:   S                                          Split: Byte
                      A: 1 Socket                          End mode: OFF
  Free memory:  21    B: 2 Socket                         Base Port: 238h
    Free disk:  21    C: 3 Socket                          CheckSum: 6E0A
                      D: 4 Socket
   Loaded file:  C:                   .CAL                   H/W Rev: 1.0
    Macro file:                                              S/W Rev: 1.0a
    Batch file:                                            Serial No: CHC2304

                  Device Information                       PPS Option
  Manufacturer:  Atmel                                  Algo: Smap
   Device Name:  AT27C080                               Vccp: 6.5V
   Module Name:  021A?                                   Vpp: 13V
  Adapter Name:  <None>                                 Tpwp: 50us
   Num. of Pins: 32
   Device Size:  100000h(1048576)Bytes

 F2-Save F3-Load F5-Function F6-Edit F7-10-Select ALT-X-Exit          ESC-MENU
```

DEVICE / SPLIT DATA

When programming devices for a 16-bit or 32-bit environment, you will need to split your data onto two or four devices.

SPLIT PROGRAMMING EXAMPLES

EXAMPLE 1: Programming two 8-bit EPROM as follows:



1. Load a 16-bit file into the buffer.

2. Select the target device from menu.

3. Insert the target device (#1) into the ZIF socket.

4. Invoke EVEN in Split data menu.

5. Program the device (#1).

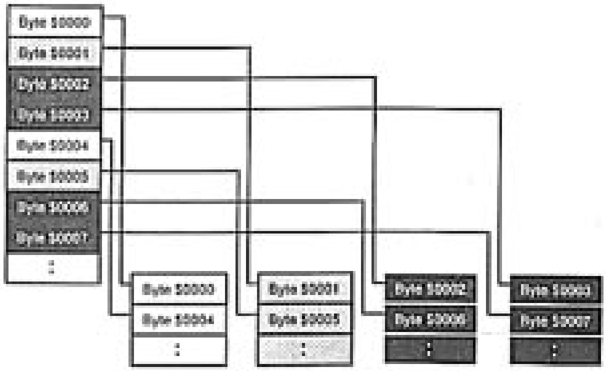
6. Remove the device (#1) and insert the second device (#2) into the ZIF socket.

7. Invoke ODD.

8. Program the second device.

Now, you have two 8-bit EPROMs that have been programmed. The first EPROM (#1) contains all the even address or low bytes and the second (#2) device contains all the odd address or high bytes.

EXAMPLE 2: Programming four 8-bit EPROMs as follows:



1. Select the target EPROM.

2. Load the hex file (32-bit file) into the buffer.

3. Insert the first EPROM (#1) into the socket.

4. Invoke Word 0 in Split Data menu.

5. Program the mounted device.

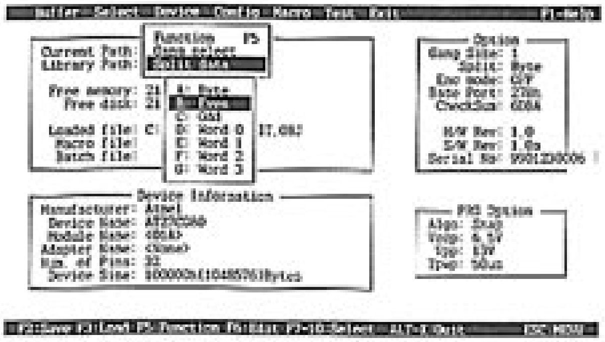
6. Remove the programmed device (#1) and insert the second device (#2) into the socket.

7. Follow the same steps as the above.

After programming the 4th EPROM with Word 3, you will have four 8-bit programmed EPROMs. The original file (32-bit) is split into four EPROMs that contains 8-bit data in each device.

EXAMPLE: Splitting a 32-bit file into four 8-bit EPROMs.

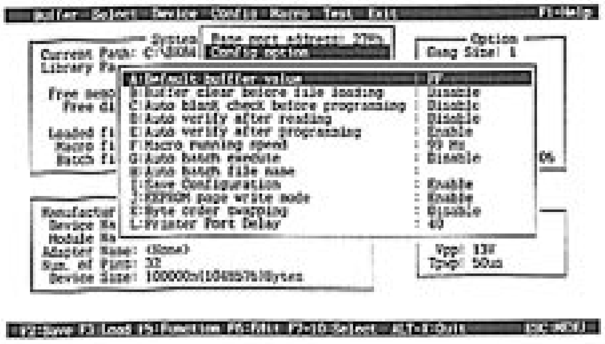
Original hex file (32-bit)



```
 Buffer  Select  Device  Config  Macro  Test  Exit                     F1-Help

  Current Path:  Function     F5                               Option
  Library Path:  Gang select                             Gang Size: 1
                 Split Data                                  Split: Byte
                                                          Enc mode: OFF
   Free memory: 21  A: Byte                              Base Port: 278h
     Free disk: 21  B: Even                               CheckSum: 608A
                    C: Odd
   Loaded file: C:  D: Word 0   17.0BJ                     H/W Rev: 1.0
    Macro file:     E: Word 1                              S/W Rev: 1.0a
    Batch file:     F: Word 2                            Serial No: 9901230006
                    G: Word 3

              Device Information                            PRG Option
  Manufacturer: Atmel                                   Algo: Intel
   Device Name: AT27C080                                Vcc: 6.5V
   Module Name: <DL8>                                   Vpp: 13V
  Adapter Name: <None>                                 Tpw: 50us
  Num. of Pins: 32
   Device Size: 100000h(1048576)Bytes

 F2-Save F3-Load F5-Function F6-Edit F7-10-Select  Alt-X-Quit          ESC-MENU
```

CONFIG

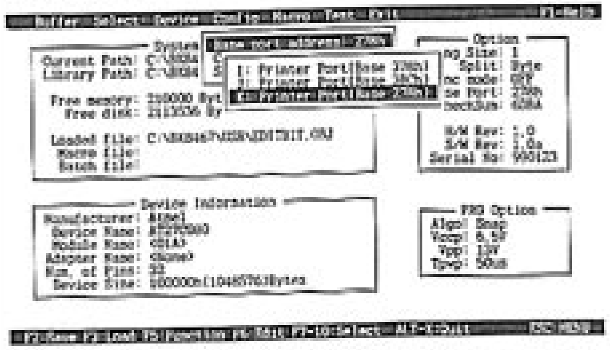
This menu allows you to specify different programming options. Setting the configuration option properly, the operation of Model 846(847) will assist your application needs. Be sure that you understand all of the optional configuration before you set each option.



```
 Buffer  Select  Device  Config  Macro  Test  Exit                     F1-Help

               System  Base port address: 278h               Option
  Current Path: C:\DOS Config option                   Gang Size: 1
  Library Pa
              A:Default buffer value                    : FF
   Free memo  B:Buffer clear before file loading        : Disable
     Free di  C:Auto blank check before programming     : Disable
              D:Auto verify after reading               : Disable
   Loaded fi  E:Auto verify after programming           : Enable
    Macro fi  F:Macro running speed                     : 99 Ms
    Batch fi  G:Auto batch execute                      : Disable       0%
              H:Auto batch file name                    :
              I:Save Configuration                      : Enable
              J:EEPROM page write mode                  : Enable
  Manufactur  K:Byte order swapping                     : Disable
   Device Na  L:Printer Port Delay                      : 40
   Module Na
  Adapter Name: <None>                                  Vpp: 13V
  Num. of Pins: 32                                     Tpw: 50us
   Device Size: 100000h(1048576)Bytes

 F2-Save F3-Load F5-Function F6-Edit F7-10-Select  Alt-X-Quit          ESC-MENU
```

CONFIG / BASE PORT ADDRESS

Select a parallel port from three available port addresses. One of them is chosen for your parallel port address.



```
 Buffer  Select  Device  Config  Macro  Test  Exit                 F1-Help

        ─── System  Base port address: 278h            ─── Option ───
  Current Path: C:\8008                                 ng Size: 1
  Library Path: C:\8008   1: Printer Port(Base 378h)      Split: Byte
                          2: Printer Port(Base 3BCh)    nc mode: OFF
   Free memory: 288000 Byt 3: Printer Port(Base 278h)   se Port: 278h
     Free disk: 2113536 By                              heckSum: 428A

   Loaded file: C:\8008467\USB\2DIT317.OBJ               H/W Rev: 1.0
    Macro file:                                          S/W Rev: 1.0a
    Batch file:                                        Serial No: 990123

        ─── Device Information ───                     ─── PRG Option ───
  Manufacturer: Atmel                                   Algo: Snap
   Device Name: AT27C080                                Vccp: 6.5V
   Module Name: <DIA>                                    Vpp: 13V
  Adapter Name: <None>                                  Tpvp: 50us
  Num. of Pins: 32
   Device Size: 100000h(1048576)Bytes

 F2:Save F3:Load F5:Function F6:Edit F7-10:Select ALT-X:Quit       ESC:MENU
```

CONFIG / CONFIG OPTION / DEFAULT BUFFER VALUE

Fill the buffer value (hexadecimal) with the initial data that you type in this field. This feature helps the user who wants to have different initial value ('00' or 'FF') in the buffer. Once you have made this change, exit to DOS, and restart 8467 to get the result of the change.

CONFIG / CONFIG OPTION / BUFFER CLEAR BEFORE FILE LOADING

When loading a file into the buffer, executing the ENABLE option fills the buffer with the data that is defined in Default Buffer Value before the file is loaded into the buffer. When you load a file that is smaller than the current buffer size, the unfilled buffer will contain the Default Buffer Value so that you may examine the buffer data more conveniently. DISABLE option keeps the same data for the unfilled buffer area after Buffer Load command is executed.

NOTE: The Buffer Clear means that the current buffer will be filled with the Default Buffer Value. It can be any data of Hexadecimal values.

CONFIG / CONFIG OPTION / AUTO BLANK CHECK BEFORE PROGRAMMING

Enabling Auto Blank Check Before Programming verifies whether the device is erased before programming. Disabling Auto Blank Check Before Programming prevents this check from occurring.

CONFIG / CONFIG OPTION / AUTO VERIFY AFTER READING

Setting the configuration menu to ENABLE will allow you to verify whether the device data is the same as the data in your current buffer after reading the source device.

CONFIG / CONFIG OPTION / AUTO VERIFY AFTER PROGRAMMING

Setting the option to ENABLE will allow you to verify whether the device data is the same as the one in your current buffer after programming a device.

CONFIG / CONFIG OPTION / MACRO RUNNING SPEED

Specify the time interval between the command executions when you run a macro file.

CONFIG / CONFIG OPTION / AUTO BATCH EXECUTE

Whenever 8467 is invoked, enabling the Auto Batch Execute will execute the macro batch file that is located in Auto Batch File Name automatically. To stop this execution, press any key. In order to execute this command, you may exit from the current screen display. Disabling the Auto Batch Execute will let you execute the commands manually.

CONFIG / CONFIG OPTION / AUTO BATCH FILE NAME

For the Auto Batch Execute, you can type a name of macro batch file that is in this field. The file will be placed in your current file directory.

CONFIG / CONFIG OPTION / SAVE CONFIGURATION

ENABLE will save any changed value to configuration to the 8467.CFG file in your **Model 846(847)** file directory. DISABLE will prevent any change in configuration and lose any change after quitting to DOS shell.

CONFIG / CONFIG OPTION / EEPROM PAGE WRITE MODE

In order to program EEPROM devices fast, most of programming algorithms will support Page Write Mode. Enabling this option allows an EEPROM to be written by pages. *If your computer is not fast enough to write a page in proper time, this option is not useful because of the computer speed.* For the user who has XT or AT computer with lower clock speed, DISABLE(byte mode) option is recommended in order to program an EEPROM that supports Page Write Mode according to IC manufacturer specification.

CONFIG / CONFIG OPTION / BYTE ORDER SWAPPING

This option applies only to 16-bit wide (E)EPROMs or Flash Memory.

User data is displayed in the buffer according to the Intel convention with the default value set at **Disable.** Enabling this option allows you to use data according to the Motorola convention during PROGRAM and VERIFY operations under the DEVICE/FUNCTION menu. However, **the data in the buffer is not physically swapped.**

When enable, the MSB(Most Significant Byte) of data is located to EVEN addresses(0,2,4,...) and the LSB(Least Significant Byte) of data is located to ODD addresses(1,3,5,...).

Byte swap is useful for example, if an assembler creates a file in Intel format, in which the low byte is read before the high byte, but the file must be in Motorola format, in which the high byte is read before the low byte.

Sample data file (Motorola EXORmacs Format, Code 87):

```
S00B00004441544120492F4FF3

S1130000012345678 9ABCDEF00112233445566 7750

S9030000FC
```

Data file opened with format 87 and displayed in the editor (8-bit addressing mode):

```
CURSOR AT LOCATION: 00000000    8 BIT ADDRESSING
            HEXADECIMAL                                      ASCII
ADDRESS  -0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -A -B -C -D -E -F  0123456789ABCDEF
00000000 01 23 45 67 89 AB CD EF 00 11 22 33 44 55 66 77  #Eg ...  "3Dul w
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Example #1: Programming one 16-bit device (Data word width = 16, Odd/even byte swap = disabled)

The user data is allocated as follows:

		Device	
		MSB	LSB
Device Address :	0	23	01
	1	67	45
	2	AB	89
	3	EF	CD

Sample data file (Motorola EXORmacs Format, Code 87):

```
S00B00004441544120492F4FF3

S1130000012345678 9ABCDEF00112233445566 7750

S9030000FC
```

Data file opened with format 87 and displayed in the editor (8-bit addressing mode):

```
CURSOR AT LOCATION: 00000000    8 BIT ADDRESSING
            HEXADECIMAL                                      ASCII
ADDRESS  -0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -A -B -C -D -E -F  0123456789ABCDEF
00000000 01 23 45 67 89 AB CD EF 00 11 22 33 44 55 66 77  #Eg ..  "3DUf w
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Example #2: Programming one 16-bit device (Data word width = 16, Odd/even byte swap = Enabled)

ENABLE will save any changed value to configuration to the 8467.CFG file in your **Model 846(847)** file directory. DISABLE will prevent any change in configuration and lose any change after quitting to DOS shell.

CONFIG / CONFIG OPTION / EEPROM PAGE WRITE MODE

In order to program EEPROM devices fast, most of programming algorithms will support Page Write Mode. Enabling this option allows an EEPROM to be written by pages. *If your computer is not fast enough to write a page in proper time, this option is not useful because of the computer speed.* For the user who has XT or AT computer with lower clock speed, DISABLE(byte mode) option is recommended in order to program an EEPROM that supports Page Write Mode according to IC manufacturer specification.

CONFIG / CONFIG OPTION / BYTE ORDER SWAPPING

This option applies only to 16-bit wide (E)EPROMs or Flash Memory.

User data is displayed in the buffer according to the Intel convention with the default value set at **Disable.** Enabling this option allows you to use data according to the Motorola convention during PROGRAM and VERIFY operations under the DEVICE/FUNCTION menu. However, **the data in the buffer is not physically swapped.**

When enable, the MSB(Most Significant Byte) of data is located to EVEN addresses(0,2,4,...) and the LSB(Least Significant Byte) of data is located to ODD addresses(1,3,5,...).

Byte swap is useful for example, if an assembler creates a file in Intel format, in which the low byte is read before the high byte, but the file must be in Motorola format, in which the high byte is read before the low byte.

Sample data file (Motorola EXORmacs Format, Code 87):

```
S00B00000444154412049 2F4FF3
S1130000012345 6789ABCDEF00112233445566 7750
S9030000FC
```

Data file opened with format 87 and displayed in the editor (8-bit addressing mode):

```
CURSOR AT LOCATION: 00000000    8 BIT ADDRESSING
           HEXADECIMAL                                 ASCII
ADDRESS  -0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -A -B -C -D -E -F  0123456789ABCDEF
00000000 01 23 45 67 89 AB CD EF 00 11 22 33 44 55 66 77  .#Eg . . . '3Du' w
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Example #1: Programming one 16-bit device (Data word width = 16, Odd/even byte swap = disabled)

The user data is allocated as follows:

		Device	
		MSB	LSB
Device Address :	0	23	01
	1	67	45
	2	AB	89
	3	EF	CD

Sample data file (Motorola EXORmacs Format, Code 87):

```
S00B000004441544120492F4FF3
S1130000012345 6789ABCDEF001122334455667750
S9030000FC
```

Data file opened with format 87 and displayed in the editor (8-bit addressing mode):

```
CURSOR AT LOCATION: 00000000    8 BIT ADDRESSING
           HEXADECIMAL                                 ASCII
ADDRESS  -0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -A -B -C -D -E -F  0123456789ABCDEF
00000000 01 23 45 67 89 AB CD EF 00 11 22 33 44 55 66 77  .#Eg . . '3Du' w
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Example #2: Programming one 16-bit device (Data word width = 16, Odd/even byte swap = Enabled)

The user data is allocated as follows:

		Device	
		MSB	LSB
Device Address :	0	01	23
	1	45	67
	2	89	AB
	3	CD	EF

CONFIG / CONFIG OPTION / PRINTER PORT DELAY

Since the ISA-bus clock speed is not fast enough with that of the CPU when you use a fast computer such as Pentium 90/133/166 Mhz, we designed this option to facilitate the problem. The default value is 0. For computers that have CPU speed of greater or equal to 133 Mhz, we recommend that you set the Port Delay to 40. In most cases, this option will help to solve the communication problem between your PC and the Model 846/847.

CONFIG / SET LIBRARY PATH

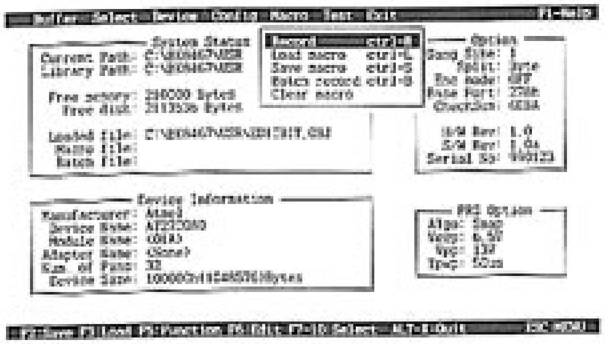
This option allows you to enter the current path of 8467.DAT file. Be sure that the path is same as the current location of 8467.EXE file when you move the Model 846(847) file to other location in your hard disk.

CONFIG / HARDWARE TEST

This option is useful to determine your Model 846/847 programmer hardware operating status. There are two kinds of tests: a "Logic Level test" and a "Pin Driver Self test" . The "Logic Level test" is usually performed at the programmer's manufacturer facility. Users can use the "Pin Driver Self test" to perform a hardware diagnosis if they are suspecting a hardware problem that may have occurred with the programmer.

MACRO

This section explains how the MACRO function is operated for automation. To speed up your repetitive work cycle or customize routine job, this feature is extremely convenient.



```
Record          ctrl-R
Load macro      ctrl-L
Save macro      ctrl-S
Batch record    ctrl-B
Clear macro
```

```
System Status
 Current Path: C:\[illegible]
 Library Path: C:\[illegible]

  Free memory: 280000 Bytes
    Free disk: [illegible] Bytes

  Loaded file: C:\[illegible]
   Macro file:
   Batch file:
```

```
Option
 Gang Size: 1
     Split: Byte
  Exe mode: OFF
 Base Port: 278h
  CheckSum: 4E3A

   H/W Rev: 1.0
   S/W Rev: 1.0x
 Serial No: 990123
```

```
Device Information
 Manufacturer: Atmel
  Device Name: AT27C080
  Module Name: (DIA)
 Adapter Name: (None)
 Num. of Pins: 32
  Device Size: 100000h(1048576)Bytes
```

```
PRG Option
 Algo: [illegible]
 Vddp: 6.5V
  Vpp: 13V
 Tpwp: 50us
```

MACRO / RECORD

Store a sequence of commands and assign them to any single key or key combinations(except Shift and Alt keys) from your keyboard. The sequence of commands is placed in the memory buffer temporarily so that you may save the sequence of commands in a macro file permanently for your later usage.

After choosing this command, start recording by defining a macro key. A macro key may be any single key or key combination except hot-keys that are preoccupied by Model 846(847). Any of the following key strokes will be recorded as a macro sequence. Finish the recording by hitting Alt-Q.

CAUTION: Always start with any hot-key for recording Macro sequence. It is important not to start with arrow keys. You can use any key from second key stroke of Macro Record. Keep this in mind for accurate usage of macro.

MACRO / LOAD MACRO

Loads a macro file into the buffer from the directory which includes the macro file. Then, you are ready to invoke the macro key which corresponds to the macro file that has been saved.

NOTE: Remember the key name that corresponds to the recorded and saved macro file name when loading and using the macro file. A macro key is always saved and loaded with a corresponding macro file name.

MACRO / SAVE MACRO

Saves a sequence of commands in a macro file. A Macro Record procedure should be completed before a Save Macro command is executed. A file name entry box appears. Type a file name with a path.

NOTE: The macro files can be deleted with a DOS command. The current file that appears in the file name entry box will not be deleted when a new macro file name is entered into the box.

MACRO / BATCH RECORD

Store a sequence of commands in the Auto Batch File Name that is located in Config menu. The batch macro file is saved automatically when the record routine is ended. This routine will be invoked whenever the 8467 is executed as long as the Auto Batch Execute is Enabled in Config menu. Hit any key to interrupt the batch execution.

NOTE: Using the batch file, the user can quit to DOS and all sequences are saved in batch file; however, the Exit to DOS cannot be recorded with Macro/Record.

MACRO / CLEAR MACRO

Clear the sequence of commands in the current memory buffer.

EXAMPLE FOR USING MACRO COMMAND

CREATE A MACRO FILE

1. Invoke Macro Record (^R) in Macro menu.
2. Choose any key from your keyboard to define a macro key and memorize the key.
3. Enter the commands sequentially using the arrow keys but you must start with a hot-key first.
4. Enter <ALT-Q> to quit the Record command at any location.

NOTE: The EXIT(DOS shell or Quit) commands are not available for Macro/Record because the Exit (DOS command) will not be recorded in the macro buffer (The Exit command in DOS switch to Model 846 software from DOS mode)

SAVE THE MACRO KEY TO A MACRO FILE

5. Invoke the Save Macro in MACRO menu. Type a file name in the name entry box.
6. Unless you quit the 8457or clear the current macro buffer, you may need to use the macro key that has been recorded in the buffer at any time.

USING THE MACRO FILE THAT WAS SAVED BEFORE

7. Invoke Macro Load command. Enter a macro file name that was saved for a Macro Key. The sequence of commands will be loaded into the buffer. In order to use the right macro key, you must use the same key that you have defined for the current macro file.
8. Invoke the Macro Key. Then the sequence of commands is played back automatically.

EXAMPLE FOR MACRO BATCH COMMAND

1. Invoke the Batch Command from MACRO menu.
2. Type a batch file name in the name entry box.
3. Using arrow keys, move the curse bar and enter the sequence of commands. Again, you have to use a hot-key as the starting key.
4. The recording process will be terminated in two different ways. Typing <ALT-Q> at any location or invoking the Exit ("QUIT") command will quit the Macro/Batch record. The sequence of commands will be saved automatically and the batch file name will appear in Auto Batch File Name that is located in Config menu.

NOTE: The Exit command is available for the Macro/Batch Record. It means that you can start and finish the 8467 with only one command in DOS (After typing 8467 in DOS, the Auto Batch will automatically execute all commands including EXIT). The Batch Mode is designed for the user who may need to program a device *while other In-Circuit Emulator is being operated with Model 846 programmer.*

USEFUL MACRO INSTRUCTION

There are two different ways to use MACRO.

1. For one time batch only, use BATCH RECORD. It will work just like AUTOEXEC.BAT in the DOS command. It will automatically execute when you run 8467.EXE.
2. For certain programming, use RECORD to create the macro procedures. In this case, be sure that you start the macro with hot-key for recording the MACRO sequence. Read the LIST OF FUNCTION KEYS and HOT-KEY combination under the "OPERATIONS" table in the manual.

<for Example the following steps>

- Select AMD 27C215
- File load
- Blank check
- Program
- Clear buffer

<Macro Instruction>

1. Run 8467.EXE
2. Move the cursor to MACRO / RECORD and Enter, or ^R(Ctrl R)
3. Select the Macro Key such as **Alt+1** for your certain procedure and remember the Key(Alt 1) for using the MACRO later.
3. Enter **F7** to select device type.
4. Type **AMD 27C512** : Select the device
5. enter F3 & load file with your selection(Format, Name)
6. enter F5 & do Blank check, Program.
7. enter F6 & ^F3 and enter "FF" into the "Fill Char=" line: Buffer clear indirectly as fill the buffer with "FF" value.
8. Do Alt+Q to finish the Record

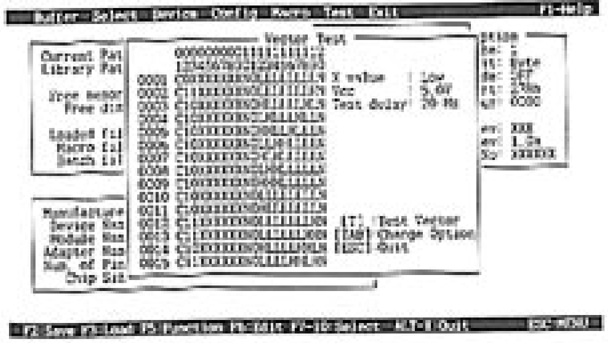
9. Move the cursor to MACRO / SAVE
10. In order to run the Macro that you just created above, insert an erased EPROM into the ZIF socket and invoke Alt1(Enter ALT and "1" together). It will process each step that we described above and stop after the buffer clears.

Since the MACRO record is in MACRO buffer, it will run when you type the ALT+1 key. You may delete your Macro using the Clear Macro. For other usage, you may create as many Macro Keys as you need.

NOTE: The Model 846 Macro invokes only the sequence of the key that you save to MACRO files. In order to make a starting point from any place in the menu when you record the macro sequences in the MACRO file, the MACRO key should start with a function key. Then the MACRO file can start at the same point(Function) regardless of your current cursor point in the Model 846 menu.

TEST

TEST / VECTOR TEST



Verifies that the PLD(PAL, GAL EPLD, etc.) behaves currently without having to prototype a circuit. In order to perform test vectors, test vectors should be in the JEDEC file when the file is loaded. Most PLD development software will generate valid test vectors automatically. Test vectors may be examined and modified with **Vector Pattern Edit/^F6** command in the buffer menu screen.

Note: Due to hardware's limitation, Vector Test is only implemented on 24-pin or less devices.

During the vector test, **Model 846(847)** applies high and low signals to the input pins of a tested PLD and observe signals at the output pins. The output results are compared to the expected results from the test vectors. Any difference will show up as an error message.

←→↑↓	Move cursor.
Home	Goto start of line
End	Goto end of line
Ctrl Home	Goto start of screen
Ctrl End	Goto end of screen
Ctrl PgUp	Goto top of buffer
Ctrl PgDn	Goto end of buffer

The following are valid characters for test vectors:

0	Apply input logic low(Vil) to an input pin
1	Apply input logic high(Vih) to an input pin
C	Clock an input pin(Vil, Vih, Vil)
F	Float pin
N	Power pin or untested output pin
V	VCC pin
X	Don't care: output values are not tested
G	GND pin
K	Clock an inverted input pin (Vih, Vil, Vih)
H	Expected result on output pin is Vih
L	Expected result on output pin is Vil
Z	Test for high impedance

Optional Operation

X value	Optional value of "don't care"
Vcc	Test Vcc value on Vcc pin
Test Delay	Test period of each vector in milli-second
[T]	Run test vector

[TAB]	To toggle between Vector Edit and Optional Operation
[ESC]	Exit the Editor

EXIT

EXIT / OS SHELL

Exit the **8467** program and return immediately to DOS command mode. You may run any DOS commands after invoking this command. The contents of programming buffer will keep the same data. To get back to programmer's menu, type 'exit' in DOS shell.

EXIT / QUIT

Quit the **Model 846(847)** program and return to DOS shell. The current data in buffer will be lost.

5. Trouble Shooting & Technical Support

This section provides customer support information such as the return material authorization policy as well as methods of obtaining B&K Precision's technical assistance and software updates.

The Model 846(847) is designed to require a minimum of technical support for both hardware and software. Making the product in U.S.A., we are supplying qualified programmers as trouble-free as possible.

1. REGISTRATION

A registration card is located in the user guide manual with the software diskette. Complete the card and return it to B&K Precision to become eligible for:

- Customer support, warranty service and technical assistance
- Notification and special pricing on new products and upgrades

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

2. SOFTWARE UPDATES

The new software updates are available at no charge via our website www.bkprecision.com, the file name is BK8467.EXE Use the new software if you have any other incorrect programming result.

3. TESTING THE HARDWARE

Make sure that your programmer works properly before you call us for a technical assistance. Refer to Hardware Test section in Config menu.

4. QUICK SELF-DIAGNOSTICS

In order to provide accurate and fast technical assistance, we recommend you to check the following information before you call our technical support department.

- Be sure that the programmer is connected properly to your computer.

- Be sure that you use the same **AC transformer** that comes with original package.
- Be sure that you use the latest software version for any programming error or minor software bug. For Int'l customers, contact your local B&K Precision's distributor.
- You may need to change your printer port even though it is working fine with your printer since the Model 846/847 communicates with your computer via the printer port in a bi-directional mode.
- If you run your computers under Windows 95, please remember that your Model 846/847 software should be run under the DOS mode within Windows 95. Model 846/847 has not yet been designed to work under Windows NT.
- Remember that your programmer is not functional without the corresponding MODULE for the device that you select from the menu.

5. CALLING CUSTOMER SUPPORT

B&K Precision provides telephone technical assistance during normal business hours(9:00 AM to 5:00 PM, Pacific Standard Time).

- Please call our Technical Support Department or your local B&K Precision's distributor while you are at your computer and be prepared to repeat the sequence of steps leading up to the problem
- Have the following information ready when you call :

- The invoice number for the user who bought the Model 846(847) from B&K Precision.
- The distributor's name and the purchased date
- The model & serial number in the back side of programming module.
- Your hardware software revision number from PRG. Option menu on the main screen .
- Description of problem with error message.
- The exact part number you were working to program.

Back to EPROM Product Page

Models 846 & 847 Device List

Additional Devices are being added continuously.

Altera

EP330 <06B>
EP600 <06B>
EP600I <06B>
EP610 <06B>
EP610I <06B>
EP900 <06B>
EP900I <06B>
EP910 <06B>
EP910I <06B>
EP1800LC P31 <06B>
EP1810JC P31 <06B>
EP1830JC P31 <06B>
EPC1 <09B-1>
EPC1213 <09B-1>
EPC1064 <09B-1>

MBM29F002T-pd cic <05A>
MBM29F002T-pftn cic <05A>
MBM29F002B-pd cic <05A>
MBM29F002B-pftn cic <05A>
MBM29LV002T/B cic <05A>
MBM29LV400T/B cic <05A>
MBM29LV004T/B cic <05A>
MBM29F200AT cic <05A>

PIC16C58A/B(OTP) <03A>
PIC16C61 <03A>
PIC16C62 <03A>
PIC16C62A <03A>
PIC16C63 <03A>
PIC16C621 <03A>
PIC16C622 <03A>
PIC16C64 <03B>
PIC16C64A <03B>
PIC16C65 <03B>
PIC16C65A <03B>
PIC16C66 <03B>
PIC16C67 <03B>
PIC16C71 <03A>
PIC16C710 <03A>
PIC16C711 <03A>

TS27128A
TS27256 <
TS2764A <
TS27C256
TS27C64 <
TS27C64A
TS28C16A
TS28C17A
ST62E20 <
ST62E20B
ST62E25 <
ST62E25B
ST62E60 <
ST62E60B
ST62E65 <
ST62E65B

Am27C040 <01A> *
Am27C128 <01A> *
Am27C256 <01A> *
Am27C512 <01A> *
Am27C64 <01A> *
Am27H010 <01A> *
Am27H256 <01A> *
Am27HB010 <01A> *
Am27C1024 <01B>
Am27C2048 <01B>
Am2817A <01A> *
Am2864A <01A> *
Am2864AE <01A> *
Am2864B <01A> *
Am2864BE <01A> *
Am28F010 <01A> *
Am28F020 <01A> *
Am28F256 <01A> *
Am28F512 <01A> *
AM29F010 <05B>
AM29F040 <05B>
AM29F040B <05B>
AM29F080 <05A>
AM29F016 <05A>
AM29F100B/T cic <05A>
AM29F200B/T cic <05A>
AM29F200AB <05A>
AM29F200AT <05A>
AM29F400AB cic <05A>
AM29F400AT cic <05B>
AM29F400B/T cic <05B>

28C16 <01A> *
28C17 <01A> *
28C17F <01A> *
28C64 <01A> *
28C64A <01A> *
28C64AX <01A> *

Generic-Algorithm

27128<21V> <01A> *
27128A<12.5V> <01A> *
2716<25V> <01A> *
27256<12.5> <01A> *
2732<25V> <01A> *
2732A<21V> <01A> *
2732B<12.5V> <01A> *
27512<12.5V> <01A> *
2764<21V> <01A> *
2764A<12.5V> <01A> *
27C128<12.5V> <01A> *
27C128<21V> <01A> *
27C16<25V> <01A> *
27C256<12.5V> <01A> *
27C32<25V> <01A> *
27C512<12.5V> <01A> *
27C64<12.5V> <01A> *
27C64<21V> <01A> *

LG-Semi.

M5L27256K <01A> *
M5L2732 <01A> *
M5L27512K <01A> *
M5L2764K <01A> *
M5M27128P <01A> *
M5M27256P <01A> *
M5M27512P <01A> *
M5M2764P <01A> *
M5M27C101K <01A> *
M5M27C100K <01A> *
M5M27C101P <01A> *
M5M27C128K <01A> *
M5M27C201K <01A> *
M5M27C256AK <01A> *
M5M27C256K <01A> *
M5M27C256P <01A> *
M5M27C401K <01A> *
M5M27C512AK <01A> *
M5M27C512AP <01A> *
M5M28C64AP <01A> *
M5M28C64P <01A> *
M5M27C102K <01B>
M5M27C102P <01B>
M5M27C202K <01B>
M5M27C402K <01B>
M5M29F016VP cic <01B>
M5M29FB800 cic <01B>
M5M29FT800 cic <01B>

Mostek

ET2716 <01A> *

GAL16V8/
GAL16V8/
GAL16V8/
GAL16V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/

Sharp

LH57128 <
LH57256 <

Am27C040 <01A> *
Am27C128 <01A> *
Am27C256 <01A> *
Am27C512 <01A> *
Am27C64 <01A> *
Am27H010 <01A> *
Am27H256 <01A> *
Am27HB010 <01A> *
Am27C1024 <01B>
Am27C2048 <01B>
Am2817A <01A> *
Am2864A <01A> *
Am2864AE <01A> *
Am2864B <01A> *
Am2864BE <01A> *
Am28F010 <01A> *
Am28F020 <01A> *
Am28F256 <01A> *
Am28F512 <01A> *
AM29F010 <05B>
AM29F040 <05B>
AM29F040B <05B>
AM29F080 <05A>
AM29F016 <05A>
AM29F100B/T cic <05A>
AM29F200B/T cic <05A>
AM29F200AB <05A>
AM29F200AT <05A>
AM29F400AB cic <05A>
AM29F400AT cic <05B>
AM29F400B/T cic <05B>

28C16 <01A> *
28C17 <01A> *
28C17F <01A> *
28C64 <01A> *
28C64A <01A> *
28C64AX <01A> *

Generic-Algorithm

27128<21V> <01A> *
27128A<12.5V> <01A> *
2716<25V> <01A> *
27256<12.5> <01A> *
2732<25V> <01A> *
2732A<21V> <01A> *
2732B<12.5V> <01A> *
27512<12.5V> <01A> *
2764<21V> <01A> *
2764A<12.5V> <01A> *
27C128<12.5V> <01A> *
27C128<21V> <01A> *
27C16<25V> <01A> *
27C256<12.5V> <01A> *
27C32<25V> <01A> *
27C512<12.5V> <01A> *
27C64<12.5V> <01A> *
27C64<21V> <01A> *

LG-Semi.

M5L27256K <01A> *
M5L2732 <01A> *
M5L27512K <01A> *
M5L2764K <01A> *
M5M27128P <01A> *
M5M27256P <01A> *
M5M27512P <01A> *
M5M2764P <01A> *
M5M27C101K <01A> *
M5M27C100K <01A> *
M5M27C101P <01A> *
M5M27C128K <01A> *
M5M27C201K <01A> *
M5M27C256AK <01A> *
M5M27C256K <01A> *
M5M27C256P <01A> *
M5M27C401K <01A> *
M5M27C512AK <01A> *
M5M27C512AP <01A> *
M5M28C64AP <01A> *
M5M28C64P <01A> *
M5M27C102K <01B>
M5M27C102P <01B>
M5M27C202K <01B>
M5M27C402K <01B>
M5M29F016VP cic <01B>
M5M29FB800 cic <01B>
M5M29FT800 cic <01B>

Mostek

ET2716 <01A> *

GAL16V8/
GAL16V8/
GAL16V8/
GAL16V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/
GAL20V8/

Sharp

LH57128 <
LH57256 <

AM29F400B/T cic <05B>
AM29F400BB cic <05B>
AM29F400BT cic <05B>
AM29F800B cic <05A>
AM29F800T cic <05A>
AM29LV800B cic <05A>
AM29LV800T cic <05A>
AM29F800AB cic <05A>
AM29F800AT cic <05A>
AM29F016 P59 <05A>
AM29F002T <05B>
AM29F002NT <05B>
AM29F002NB <05B>
AM29F002B <05B>
AM29LV200B <05B>
AM29LV200T <05B>
AM29LV400B <05B>
AM29LV400T <05B>
AM29LV002T <05B>
AM29LV002B <05B>
AM29LV004T <05B>
AM29LV004B <05B>
AM29LV008T <05B>
AM29LV008B <05B>
PALCE16V8H <02B>
PALCE16V8Q <02B>
PALCE16V8Z <02B>
PALCE20V8H-25 <02B>
PALCE20V8Q-25 <02B>
PALCE20V8H-10/4

LG-Semi.

GMS80C701 <01A>

Gould

PEEL18CV8 <02B>
PEEL20CG10 <02B>
PEEL22CV10 <02B>

Greenwich

GR27128 <01A> *
GR27256 <01A> *
GR27512 <01A> *
GR2764 <01A> *
GR281 <01A>
GR881 <01A>
GR3281 <01A>
GR12882 <01A>
GR12883 <01A>

Hitachi

HN27128A <01A> *
HN27128AG <01A> *
HN27128AP <01A> *
HN27256 <01A> *
HN27256G <01A> *
HN27256P <01A> *
HN27512 <01A> *
HN27512G <01A> *
HN27512P <01A> *

ET2716 <01A> *
ETC2716 <01A> *
ETC2732 <01A> *
MK2716 <01A> *
MK2764 <01A> *

Motorola

MCM2716 <01A> *
MC68705P3 <2B-1>
MC68705P5 <2B-1>
MC68705R3 <12A>
MC68705R5 <12A>
MC68705U3 <12A>
MC68705U5 <12A>
MC68HC11A1-P <10A>
MC68HC11A8-P <10A>
MC68HC11E11-FN A52 <10A>
MC68HC11E9-FN A52 <10A>
MC68HC11E20-FN A52 <10A>
MC68HC711D3-FN P05<10A>
MC68HC711D3-P <10A>
MC68HC811E2-FN A52<10A>
MC68HC711E9-FN A52<10A>
MC68HC711L6-FN

LH5764 <0
LH28F016
LH28F032
LH28F400

Signetics

8582C <02
8582D <02
8582E <02
8592 <02A
PLC18V8Z
27C010 <0
27C256 <0
27C512 <0
27C64A <0
27C210 <0
27C240 <0
87C51 <01
87C51FA <
87C51FB <
87C51FC <
87C51GB P
87C52 <01
87C524 <0
87C528 <0
87C550 P2
87C552 P2
87C575 <0
87C576 <0

PALCE20V8H-10/4 <02B>
PALCE20V8H-15/4 <02B>
PALCE20V8H-25/4 <02B>
PALCE20V8Q-15/4 <02B>
PALCE20V8Q-25/4 <02B>
PALCE22V10H <02B>
PALCE22V10Q <02B>
PALCE22V10Z <02B>
PALLV22V10 <02B>
PALLV22V10Z <02B>
PALCE20RA10H <02B>
PALCE20RA10Q <02B>
PALCE26V12H <02A>
PALCE26V12H/4 <02A>
MACH110 P17 <02A>
MACH111 P17 <02A>
MACH111-SP P17 <02A>
MACH120 P20 <02A>
MACH130 P18 <02A>
MACHLV210 <02A>
MACH210 P17 <02A>
MACH211 P17 <02A>
MACH211-SP P17 <02A>
MACH215 P17 <02A>
MACH220 P20 <02A>

HN27512P <01A> *
HN27C101AG <01A> *
HN27C101AP <01A> *
HN27C101G <01A> *
HN27C101P <01A> *
HN27C256 <01A> *
HN27C256AG <01A> *
HN27C256G <01A> *
HN27C256HG <01A> *
HN27C301G <01A>*
NH27C301P <01A>*
HN27C4001G <01A> *
HN27C512AG <01A> *
HN27C64G <01A> *
HN462716 <01A> *
HN462716G <01A> *
HN462732G <01A> *
HN48016P <01A> *
HN4827128G <01A> *
HN4827128P <01A> *
HN482732AG <01A> *
HN482764G <01A> *
HN482764P <01A> *
HN58064 <01A> *
HN58C256P(Page) <01A> *
HN58C256P(Single) <01A> *
HN58C65P <01A> *
HN58C1001 <01A>

MC68HC711L6-FN A68<10A>
MC68HC711K4-FN A84<10A>
MC68HC711E20-FN A52<10A>
MC68HC11A1-FN A52<10A>
MC68HC11A8-FN A52<10A>
MC68HC705C4A-FN P05<10B>
MC68HC705C4-P <10B>
MC68HC705C5-P <10B>
MC68HC705C5-FN P05<10B>
MC68HC705C8-FN P05<10B>
MC68HC705C8-P <10B>
MC68HC705C8A-P <10B>
MC68HC705C8A-FN P05<10B>
MC68HC705C8A-P <10B>
MC68HC705C9-FN P05<10B>
MC68HC705C9-P <10B>
MC68HC705C9A-FN P05<10B>
MC68HC705C9A-P <10B>
MC68HC705D9-FN P05<10B>
MC68HC705D9-P <10B>

87C592 P2
87C652 <0
87C654 <0
87C750 <0
87C751 <0
87C752 <0
P3Z22V10
P5Z22V10

Siliconia

SM8203 <0

SMOS

SPM27128
SPM27128
SPM27C25
SPM27C64
SPM27C64
SPM2864C

SST

PH28EE01
PH28EE01
PH29EE02
28SF040 <
PH29LE01
PH29EE51

Synertek

MACH220 P20 <02A>
MACH230 P18 <02A>
MACH435 P18 <02A>
8751H <01A>
8753H <01A>
87C51 <01A>
87C521T2 <01A>
87C521 <01A>
87C541 <01A>

Atmel

AT17C65 <02A>
AT17C128 <02A>
AT17C256 <02A>
AT24C01 <02A>
AT24C01A <02A>
AT24C02 <02A>
AT24C04 <02A>
AT24C08 <02A>
AT24C128 <02A>
AT24C256 <02A>
AT24C164 <02A>
AT24C16 <02A>
AT24C32 <02A>
AT24C64 <02A>
AT25C02 <02A>
AT25010 <02A>
AT25020 <02A>
AT25040 <02A>
AT25080 <02A>

HN27C1024HG <01B>
HN27C4096G <01B>

Hyundai

HY2764 <01A> *
HY27C64A <01A> *
HY93C46 <02A>
HY18CV8 <02B>

ICT

27CX010 <01A> *
93C46 <02A>
93C46A <02A>
93C56A <02A>
93C66A <02A>
93CX46 <02A>
93CX56 <02A>
93CX66 <02A>
PEEL18CV8 <02B>
PEEL20CG10 <02B>
PEEL20CG10A <02B>
PEEL22CV10 <02B>
PEEL22CV10A <02B>
PEEL22CV10A+ <02B>
PEEL22CV10Z <02B>
PA7024 <02B>
PA7128 <02A>
PA7140 <02A>

Intel

MC68HC705D32A-P <10B>
MC68HC705D32A-FN
P05<10B>
MC68HC705J2-P <10B>
MC68HC705B5-FN
B52<12A>
MC68HC705B16-FN
B52<12B>
MC68HC705B32-FN
B52<12A>
MC68HC811E2-P <10A>
MC68HC705P6 <10B>
MC68HC705P6A-P <10B>
MC68HC705P9-P <10B>
MC68HC805P18-P P62<10B>
MC68HC805P18-DW <10B>
MC68HC705X32-FU
P63<7A>
MC68HC705X4-DW B64
<10A>
MC68HC705X16-FU
B64<10A>

(All DIP packages work w/o adapters)

National-Semiconductor

MM2716 <01A> *
MM2716E <01A> *
NM27C010 <01A> *

SY2716 <01

Texas-Ins

SMJ2516 <0
SMJ27C25
SMJ27C51
TMS2516 <
TMS27128
TMS2732A
TMS2764 <
TMS27C01
TMS27C02
TMS27C01
TMS27C04
TMS27C12
TMS27C25
TMS27C32
TMS27C51
TMS27C64
TMS27P32
TMS27P64
TMS27PC0
TMS27PC1
TMS27PC2
TMS27PC3
TMS27PC5
TMS27PC6
TMS27C21
TMS27C21

AT25080 <02A>
AT25160 <02A>
AT25320 <02A>
AT25640 <02A>
AT25128 <02A>
AT27BV256 <01A> *
AT27BV512 <01A> *
AT27BV010 <01A> *
AT27BV020 <01A> *
AT27BV040 <01A> *
AT27BV1024 <01A>
AT27BV4096 <01A>
AT27C010 <01A> *
AT27C010L <01A> *
AT27C020 <01A> *
AT27C040 <01A> *
AT27C080 <01A> *
AT27C128 <01A> *
AT27C256 <01A> *
AT27C256R <01A> *
AT27C512 <01A> *
AT27C512R <01A> *
AT27C1024 <01B>
AT27C1024L <01B>
AT27C4096 <01B>
AT27C516 <01B>
AT27HC1024 <01B>
AT27HC64 <01A> *
AT27HC256 <01A> *
AT27LV256A <01A>*

Intel

27010 <01A> *
27128 <01A> *
27128A <01A> *
27128B <01A> *
2716 <01A> *
27256 <01A> *
2732 <01A> *
2732A <01A> *
27512 <01A> *
2764 <01A> *
2764A <01A> *
27C010 <01A> *
27C010A <01A> *
27C020 <01A> *
27C040 <01A> *
27C128 <01A> *
27C256 <01A> *
27C100 <01A>
27C400 <01B>
27C512 <01A> *
27C64 <01A> *
2816A <01A> *
2817A <01A> *
2864 <01A> *
2864A <01A> *
28F010 <01A> *
28F020 <01A> *
28F256-###P1C2

NM27C010 <01A> *
NM27C020 <01A> *
NM27C040 <01A> *
NM27C128 <01A> *
NM27C512 <01A> *
NM27C64 <01A> *
NMC2732 <01A> *
NMC27C010 <01A> *
NMC27C020 <01A> *
NMC27C128B <01A> *
NMC27C128BN <01A> *
NMC27C128BQ <01A> *
NMC27C128C <01A> *
NMC27C128CQ <01A> *
NMC27C16 <01A> *
NMC27C16BQ <01A> *
NMC27C16H <01A> *
NMC27C16HQ <01A> *
NMC27C16Q <01A> *
NMC27C256 <01A> *
NMC27C256B <01A> *
NMC27C256BN <01A>*
NMC27C256C <01A> *
NMC27C32 <01A> *
NMC27C32B <01A> *
NMC27C32E <01A> *
NMC27C32EH <01A> *
NMC27C32H <01A> *
NMC27C32Q <01A> *
NMC27C512 <01A> *

TMS28C64
TMS27C24
TMS28F20
TMS28F20
TMS28F20
TMS28F20
TMS28F20
TMS28F20

Toshiba

TC541000J
TC541000P
TC54256A
TC54256A
TC544000P
TC54512A
TC54H102
TC571000
TC571000
TC57256 <
TC57256A
TC57256A
TC57256D
TC574000
TC574000
TC57512A
TC57H100
TC57H256
TMM24128

AT27LV256R <01A>*
AT27LV512A <01A>*
AT27LV512R <01A>*
AT27LV010 <01A>*
AT27LV010A <01A>*
AT27LV010L <01A>
AT27LV020 <01A>*
AT27LV020A <01A>*
AT27LV040 <01A>*
AT27LV040A <01A>*
AT27LV080 <01A>*
AT27LV1024 <01B>
AT27LV4096 <01B>
AT28BV16 <01A> *
AT28BV64 <01A> *
AT28C16 <01A> *
AT28C17 <01A> *
AT28C64 <01A> *
AT28C256 <01A> *
AT28HC16 <01A> *
AT28HC16L <01A> *
AT28HC256 <01A> *
AT28HC64 <01A> *
AT28HC64B <01A> *
AT28C010 <01A> *
AT28MC010 <01A> *
AT28MC040 <01A> *
AT28C040 <01A>
AT28PC64 <01A> *

28F256-###P1C2 <01A> *
28F256-###P2C2 <01A> *
28F256A <01A> *
28F512 <01A> *
28F001BX-B G6 <01A>
28F001BX-T G6 <01A>
28F002BX-B CIC <05A>
28F002BX-T CIC <05A>
28F004BX-B cic <05A>
28F004BX-T cic <05A>
28F800BV-T cic <05A>
28F800BV-B cic <05A>
28F800CV-B cic <05A>
28F800CV-T cic <05A>
28F800CE-B cic <05A>
28F800CE-T cic <05A>
28F016SA cic <05A>
28F016SV cic <05A>
28F200BX-B cic <05A>
28F200BX-T cic <05A>
28F400BX-B cic <05A>
28F400BX-T cic <05A>
28F400BV-B cic <05A>
28F400BV-T cic <05A>
28F400CE-B cic <05A>
28F400CE-T cic <05A>

NMC27C512 <01A> *
NMC27C512A <01A> *
NMC27C512AN <01A> *
NMC27C64 <01A> *
NMC27C64B <01A> *
NMC27C64BN <01A> *
NMC27C64N <01A> *
NMC27CP128 <01A> *
NMC9306 <02A>
NMC9307 <02A>
NMC9313B <02A>
NMC9314B <02A>
NMC9346 <02A>
NMC93C06 <02A>
NMC93C46 <02A>
NMC93C56 <02A>
NMC93C66 <02A>
NMC93C86 <02A>
NMC93CS06 <02A>
NMC93CS46 <02A>
NMC93CS56 <02A>
NMC93CS66 <02A>
NMC27C1024 <01B>
NMC27C2048 <01B>
GAL16V8/A <02B>
GAL20RA10 <02B>
GAL22V10 <02B>

NEC

uPD27128 <01A> *

TMM24256
TMM24256
TMM24512
TMM24512
TMM2464
TMM27128
TMM27128
TMM27128
TMM27128
TMM27128
TMM27128
TMM27256
TMM27256
TMM27256
TMM27256
TMM27256
TMM27256
TMM27256
TMM27256
TMM2732
TMM27512
TMM27512
TMM27512
TMM27512
TMM27512
TMM2764
TMM2764
TMM2764
TMM2764

AT28PC64E <01A> *
AT28LV64 <01A> *
AT28LV64B <01A>*
AT28LV256 <01A>*
AT28C1024 <01B>
AT29C010 <01A> *
AT29C010A <01A>*
AT29LV010 <01A>
AT29LV010A <01A>
AT29C020 <01A> *
AT29LV020 <01A>*
AT29C040A <01A>
AT29LV040A <01A>
AT29C040 <01A> *
AT29C1024 <01A>
AT29C256 <01A> *
AT29C257 <01A> *
AT29C512 <01A> *
AT32C16 <01B>
AT49F010 <01A>*
AT49F020 <01A>*
AT49F040 <01A>*
AT49F1025 <01B>
AT59C11 <02A>
AT59C22 <02A>
AT59C13 <02A>
AT93C46 <02A>
AT93C56 <02A>
AT93C66 <02A>
ATF16V8B/L <02B>

28F400CE-T cic <05A>
28F008SA <05A>
87C64 <01A>
8755A <01A>
P27128A <01A> *
P27128B <01A> *
P27256 <01A> *
P27512 <01A> *
P2764A <01A> *
P27C128 <01A> *
P27C256 <01A> *
P27C64 <01A> *
27210 <01B>
27C210 <01B>
27C220 <01B>
27C240 <01B>
8741AH <01A>
8742 <01A>
8742AH <01A>
8748 <01A>
8748H <01A>
8749H <01A>
8751BH <01A>
8752BH <01A>
8751H(1ms) <01A>
8751H(50ms) <01A>
87C51 <01A>
87C51FA <01A>
87C51FB <01A>
87C51FC <01A>
87C54 <01A>

uPD27128 <01A> *
uPD2716 <01A> *
uPD27256 <01A> *
uPD2732 <01A> *
uPD2732A <01A> *
uPD2764 <01A> *
uPD27C1000 <01A> *
Upd27c1000A <01A> *
uPD27C1001A <01A> *
uPD27C1001D <01A> *
uPD27C2001 <01A> *
uPD27C256 <01A> *
uPD27C256A <01A> *
uPD27C4001 <01A> *
uPD27C512 <01A> *
uPD27C64 <01A> *
uPD27C8001 <01A> *
uPD27C8000 <01B>
uPD28C64 <01A> *
uPD28C64C <01A> *
uPD27C1024 <01B>
uPD27C1024A <01B>
uPD27C4096 <01B>
8749H <01A>

OKI

MSM271000 <01A> *
MSM271000AS <01A> *
MSM271000ZB <01A> *
MSM27128A <01A> *
MSM27128AS <01A> *

TMM2764
TMM2764
TC541001J
TC541001P
TC571024
TC571001
TC571001
TC57H100
TC574096
TC574200
TC578200
TC57H102
TC57H102
TC57H102

VLSI

VT27C256
VT27C512
VT27C64 <

Wafersca

WS27C010
WS27C040
WS27C128
WS27C256
WS27C256
WS27C512
WS27C512
WS27C64F
WS27C64L

ATF16V8C <02B>
ATF16V8CZ <02B>
ATF20V8B/L <02B>
ATF22V10B/L <02B>
ATF1500 <06A>
ATF1500A <06A>
AT89C51 <01A>
AT89C52 <01A>
AT89S53 <01A>
AT89C55 <01A>
AT89S8252 <01A>
AT89LV51 <01A>
AT89LV52 <01A>
AT89C1051 <02B>
AT89C2051 <02B>
AT90S1200 <02B>
AT90S2313 <02B>
AT90S4415 <02A>
AT90S8515 <02A>

Catalyst

CAT24C02 <01A>
CAT24C04 <01A>
CAT24C08 <01A>
CAT24C16 <01A>
CAT27010 <01A> *
CAT27128A <01A> *
CAT27256 <01A> *
CAT27512 <01A> *
CAT2764A <01A> *
CAT27C210 <01B> *
CAT27HC256 <01A> *

87C251SB <01A>
87L51FA(PLCC) P03 <01A>
87L51FB(PLCC) P03 <01A>
87L51FC(PLCC) P03 <01A>
87C196CA E68 <01B>
87C196JQ C52 <01B>
87C196JR C52 <01B>
87C196JT C52 <01B>
87C196JV C52 <01B>
87C196KB D68 <01B>
87C196KC D68 <01B>
87C196KD D68 <01B>
87C196KQ C68 <01B>
87C196KR C68 <01B>
87C196KS C68 <01B>
87C196KT C68 <01B>
87C196MC C84 <01B>
87C196MD C84 <01B>
87C196MH C84 <01B>

ISSI

IS27HC256 <01A> *
IS27HC512 <01A> *
IS27HC010 <01A> *
IS28F010 <01A> *
IS93C46 <02A>
IS93C66 <02A>

Lattice

MSM27128AS <01A> *
MSM27128AZB-RS <01A> *
MSM2716 <01A> *
MSM27256 <01A> *
MSM27256AS <01A> *
MSM27256ZB-RS <01A> *
MSM2732 <01A> *
MSM2732A <01A> *
MSM27512 <01A> *
MSM27512AS <01A> *
MSM27512ZB-RS <01A> *
MSM2764 <01A> *
MSM2764A <01A> *
MSM2764AS <01A> *
MSM2764AZB-RS <01A> *
MSM2764RS <01A> *
MSM27C128AS <01A> *
MSM27C2000 <01A> *
MSM27C256 <01A> *
MSM27C256H <01A> *
MSM27C256HZB <01A> *
MSM27C256ZB <01A> *
MSM27C64AS <01A> *
MSM2816ARS <01A> *
MSM271024 <01B>
MSM271024AS <01B>
MSM271024ZB <01B>
MSM27C1024 <01B>
MSM27C1024AS <01B>
MSM27C2048 <01B>
MSM27C1652CZ <01B>
MSM27C402ZB <05A>

WS57C128
WS57C256
WS57C512
WS57C64F
WS27C210

Winbond

W27CE010
W27C010 <
W27E257 <
W27E512 <
W27E040 <
W27E4096
W27F512 <
W27F010 <
W27F256 <
W29EE010
W29EE011
W29EE021
W29EE512
W29C020 <

Xicor

X2402 <02
X2404 <02
X24C01 <0
X24C01A <
X24C02 <0
X24C04 <0
X24C08 <0
X2416 <02

CAT27HC256 <01A> *
CAT28C16A <01A> *
CAT28C17A <01A> *
CAT28C256 <01A> *
CAT28C64A <01A> *
CAT28C64B <01A> *
CAT28C65A <01A> *
CAT28F010 <01A> *
CAT28F020
CAT33C101 <02A>
CAT33C104 <02A>
CAT33C116 <02A>
CAT33C116I <02A>
CAT33C204 <02A>
CAT35C102 <02A>
CAT35C104 <02A>
CAT35C116 <02A>
CAT35C116I <02A>
CAT35C202 <02A>
CAT35C204 <02A>
CAT59C11 <02A>
CAT93C46 <02A>
CAT93C46A <02A>
CAT93C56 <02A>

Cypress

PALCE16V8 <02B>

Dallas

DS87C520 <01A>
DS87C530 <01A>

Lattice

GAL16V8/A/B/C/D <02B>
GAL20V8/A/B/C <02B>
GAL18V10B <02B>
GAL20RA10 <02B>
GAL20RA10B <02B>
GAL20XV10 <02B>
GAL20XV10B <02B>
GAL22V10 <02B>
GAL22V10B <02B>
GAL22V10C <02B>
GAL26CV12 <02A>
GAL26CV12B <02A>
GAL6001 <02B>
GAL6001B <02B>

Macronix

MX27C256 <01A> *
MX27C64 <01A> *
MX27C1000 <01A> *
MX27C2000 <01A> *
MX28F1000 <01A>
MX28F2000 <01A>
MX28F2100B <05A>
MX27C1001 <01A>
MX27C2048 <01B>
MX27C2100 <01B>
MX27C4000 <01A> *
MX27C4096 <01B>
MX27C4100 <01B>

MSM27C402ZB <05A>

Quick-Pulse-Algorithm

27010 <01A> *
27040 <01A> *
27128A <01A> *
27256 <01A> *
27512 <01A> *
2764A <01A> *
27210 <01B>

Ricoh

RD27C256 <01A> *
RD27C64 <01A> *

Rockwell

R2764 <01A> *
R2764C <01A> *
R2764P <01A> *
R27C64 <01A> *
R2816 <01A> *

RUSSIAN

573RF2/5 <01A> *
57RF4-21 <01A> *
573RF4-21.5 <01A> *
573RF4-23 <01A> *

Samsung

KM2816A <01A> *
KM2816AI <01A> *
KM2817A <01A> *

X24164 <0
X24C16 <0
X24320 <0
X24325 <0
X25020 <0
X25021 <0
X25080 <0
X25160 <0
X25320 <0
X25640 <0
X25128 <0
X25F008 <
X25F016 <
X25F032 <
X25F064 <
X25F128 <
X25F047 <
X25F087 <
X25057 <0
X25097 <0
X25C02 <0
X24C44 <0
X2804A <0
X2816A <0
X2816B <0
X2816C <0
X28256 <0
X2864A <0
X2864B <0
X2864H <0
X28C010 <

EXEL

XLS2804A <01A> *
XLS2816A <01A> *
XLS2864A <01A> *
XLS2865A <01A> *
XLS93C46 <02A>
XLS93C56 <02A>
XLS93C66 <02A>
XLS93CS46 <02A>
XLS93LC46 <02A>
XLS93LC56 <02A>
XLS93LC66 <02A>

Fujitsu

MBM27128 <01A> *
MBM2716 <01A> *
MBM2716H <01A> *
MBM27256 <01A> *
MBM2732 <01A> *
MBM2732A <01A> *
MBM2764 <01A> *
MBM27C1001 <01A> *
MBM27C1001A <01A> *
MBM27C128 <01A> *
MBM27C128P <01A> *
MBM27C256 <01A> *
MBM27C256A <01A> *
MBM27C256A-##-W <01A> *
MBM27C256AP <01A> *
MBM27C256H <01A> *
MBM27C512 <01A> *
MBM27C512P <01A> *
MBM27C64 <01A> *
MBM28C64 <01A> *
MBM28C65 <01A> *
MBM28C256 <01A> *
MBM27C1000 <01A>
MBM27C1024 <01B>
MBM27C2048 <01B>
MBM27C4096 <01B>
MBM28F010 <01A> *

24LC02A <02A>
24LC04A <02A>
24LC16 <02A>
24LC164 <02A>
27256 <01A> *
27C128 <01A> *
27C256 <01A> *
27C512 <01A> *
27C64 <01A> *
27HC256 <01A> *
27HC256L <01A> *
27HC64 <01A> *
27HC64L <01A> *
28C04 <01A> *
28C04A <01A> *
28C04AF <01A> *
28C04F <01A> *
28C16 <01A> *
28C16A <01A> *
28C16AF <01A> *
28C16F <01A> *
28C17 <01A> *
28C17A <01A> *
28C17AF <01A> *
28C17F <01A> *
28C64 <01A> *
28C64A <01A> *
28C64AF <01A> *
28C64AX <01A> *
37LV36 <09B>
37LV65 <09B>
37LV128 <09B>
93C06 <02A>
93C46 <02A>
93C56 <02A>
93C66 <02A>
93LC46 <02A>
93LC56 <02A>
93LC66 <02A>
PIC12C508 <11A-1>
PIC12C509 <11A-1>
PIC16C52 <03B>
PIC16C52(OTP) <03A>
PIC16C54/A/B <03A>
PIC16C54/A/B(OTP) <03A>

28C256 <01A> *
28C256A <01A> *
28C256AH <01A> *
28C64 <01A> *
28C64A <01A> *
28C65 <01A> *
28HC256 <01A> *
28HC256H <01A> *
28HC64 <01A> *
E28C64 <01A> *
M28C64 <01A> *
M28C010 <01A> *
M28C010A <01A> *

SGS-Thomson

ET2716 <01A> *
ETC2716 <01A> *
ETC2732 <01A> *
M27128A <01A> *
M2716 <01A> *
M27256 <01A> *
M2732A <01A> *
M27512 <01A> *
M2764 <01A> *
M2764A <01A> *
M27C1000 <01A> *
M27C1001 <01A> *
M27C2001 <01A> *
M27C256B <01A> *
M27C4001 <01A> *
M27C405 <05B>
M27C512 <01A> *
M27C64A <01A> *
M27C801 <01A> *
M27C800 <01B>
M27C1024 <01B>
M27C4002 <01B>
M27C160 <01B>
M28C64 <01A> *
M28F256 <01A> *
M28F256A <01A> *
M28F512 <01A> *
M28F201 <01A>*
M28F101 <01A> *
M29F040 <05B>

Zilog

Z86E02 <1
Z86E02(18
Z86E03 <1
Z86E04 <1
Z86E04(18
Z86E06 <1
Z86E08 <1
Z86E21 <1
Z86E30 <1
Z86E31 <1
Z86E40 <1
Z86E61 <1
Z86E63 <1

DS1220AB <01A>
DS1220AD <01A>
DS1220Y <01A>
DS1225AB <01A>
DS1225AD <01A>
DS1225Y <01A>
DS1230AB <01A>
DS1230Y <01A>
DS1235YW <01A>
DS1243Y <01A>
DS1245Y <01A>
DS1245AB <01A>
DS1248Y <01A>
DS1643 <01A>
DS1643Y <01A>
DS1644 <01A>
DS1645AB <01A>
DS1645Y <01A>
DS1650AB <01A>
DS1650Y <01A>
DS1658AB <01B>
DS1658Y <01B>
DS1248Y <01A>
DS1245AB <01A>
DS1248Y <01A>

Dense-Pac

DPV27C101 <01A> *
DPV27C256 <01A> *
DPV27C512 <01A> *

MX27C4111 <01B>
MX27C512 <01A> *
MX27C8100 <01B>
MX27L1000 <01A> *
MX27L4000 <01A> *
MX27L256 <01A> *
MX27L512 <01A> *

Microchip

24AA01 <02A>
24AA02 <02A>
24AA04 <02A>
24AA08 <02A>
24AA16 <02A>
24AA164 <02A>
24AA65 <02A>
24C01A <02A>
24C02A <02A>
24C04A <02A>
24C08B <02A>
24C16B <02A>
24C32 <02A>
24C65 <02A>
24LC32 <02A>
24LC65 <02A>
24LC01A <02A>
24LC02B <02A>
24LC04B <02A>
24LC08B <02A>
24LC16B <02A>
24LC02A <02A>

KM2817A <01A> *
KM2817AI <01A> *
KM2864A <01A> *
KM2864AH <01A> *
KM2865A <01A> *
KM2865AH <01A> *
KM28C16 <01A> *
KM28C17 <01A> *
KM28C256 <01A> *
KM28C64A <01A> *
KM28C65A <01A> *
KM29C010 <01A> *
KM93C06 <02A>
KM93C07 <02A>
KM93C46 <02A>
KM93C56 <02A>
KM93C66 <02A>

Seeq

27128 <01A> *
27256 <01A> *
2764 <01A> *
27C256 <01A> *
2804A <01A> *
2816A <01A> *
2816AH <01A> *
2817A <01A> *
2817AH <01A> *
2864 <01A> *
2864H <01A> *
28C010 <01A> *

X28C256 <
X28C512 <
X28C64 <0
X28HC256
XM28C010
X88C64 <0
X68C64 <0
X68257 <0
X88257 <0

Xilinx

XC1718D
XC1718L <
XC1736D
XC1765D
XC1765L <
XC17128D
XC17128L
XC17256D
XC17256L
XC7236A P
XC7272A(
XC7272A(
XC7318 P0
XC7354(44
XC7354(68
XC7336 P0
XC7372(68
XC7372(84
XC73108A

<03A>

PIC16CR54A <03A>

PIC16CR54A(OTP) <03A>

PIC16C55 <03A>

PIC16C554 <03A>

PIC16C556 <03A>

PIC16C558 <03A>

PIC16C55(OTP) <03A>

PIC16C56/B <03A>

PIC16C56/B(OTP) <03A>

PIC16C57 <03A>

PIC16C57(OTP) <03A>

PIC16CR57B(OTP) <03A>

PIC16CR58A(OTP) <03A>

PIC16C58/A/B <03A>

M28F410 cic <05A>

M28F420 cic <05A>

M28V410 cic <05A>

M28V420 cic <05A>

M87C257<01A>

M9306 <02A>

M9346 <02A>

M93C06 <02A>

<nnA/B> indicates module number, each module has both A/B.

* supports for 4-gang programming; others are programmed by 48-pin socket(1-gang)

Pxx: Adapter, call us for adapter or look at the adapter list.