TO,READ,VERIFY,COMPARE will be referred to the TARGET ZONE area. On the other hand, the contents between TARGET start and end address will be programmed to the EPROM from EPROM start address. Please ref. to the terms definition at the beginning of this chapter.

NOTE

Be sure to check if the range of EPROM is correct every time you change the type number.
The normal range of each EPROM is shown below.

2716	0	---	7FF
2732	0	---	FFF
2764	0	---	1FFF
27128	0	---	3FFF
27256	0	---	7FFF
27512	0	---	FFFF
27101	0	---	1FFFF
27301	0	---	1FFFF

Q. Quit

Under main menu selection,enter Q will exit from EPPGMx.EXE to DOS.

EVEN & ODD Operating

Function P, A, R, V and C not only operate on all contiguous address of buffer, but also operate on EVEN or ODD address as shown below.

All address mapping : (by pressing key "Y")

buffer start	+0	------	device start	+0
	+1	------		+1
	+2	------		+2
	+3	------		+3
	etc.			etc.

Even address mapping : (by pressing key "E")

buffer start	+0	------	device start	+0
	+2	------		+1
	+4	------		+2
	+6	------		+3
	etc.			etc.

Odd address mapping : (by pressing key "O")

buffer start	+1	------	device start	+0
	+3	------		+1
	+5	------		+2
	+7	------		+3
	etc.			etc.

One key and One line character operating

Most of these functions need to be answered by pressing one key only. But some need to be answered by entering one line character, such as Function 2(LOAD), Function 3(SAVE) and Z(modify target zone).
One line characters answer need to be terminated by pressing <cr> after the last character.
If you want to neglect the answer and go back to the main menu please press key <ESC>.

3.2 TESTING and ADJUSTING section

This section describes the method of hardware testing and adjustment. You can skip this section if the Programmer works perfectly.

If this EPROM programmer does not work properly,please execute this software by typing key 'X' under main menu.
Follow the testing procedure step by step. Then write down the result of error point and contact us.

NOTE

Before executing this software,be sure that you have checked contacts of cable,test socket box,programmer card,TEST-SOCKET and PC slot. In our experience, many problems are due to poor contact or bad contact.

4. UTILITY PROGRAM USAGE

This section describes the invoking method of the utility program which is included in software diskette.

There are 5 utility programs available now. If you need some special utility, please let us know so that we can write it for you.

HEXBIN.EXE	: HEX to BIN converter file.
HEXBIN2.EXE	: Extended HEX to BIN converter file.
DUMP.EXE	: Binary file dump to console utility.
SPLIT2.EXE	: 2 ways file splitter for 16 bits wide file.
SPLIT4.EXE	: 4 ways file splitter for 32 bits wide file.

4.1 HEX to BINARY ROM code converter

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

This Programmer is not a stand alone programmer. We use direct I/O control technic to control it by the CPU on PC,and also share the large memory resources on PC. We write the Programmer software to load any type of file without converting anything during file loading. It loads file content byte by byte contiguously into software buffer for programming use.
We let HEX converter convert your HEX format file into an executable ROM code that is meaningful to your target CPU.

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

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

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

Max. converted size is 64 K bytes

HEXBIN2.EXE can convert INTEL,MOTOROLA extended hex format file.

Maximum conversion size is 256K bytes.
The starting address can be specified on HEXBIN2, and the leading garbage will be skipped out to maintain a small size of output binary file

The input command under DOS command prompt is :

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

[] : option, <CR> : return key or enter key.

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

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

start address : HEXACIMAL digit

1. In INTEL HEX format,the start address represents the start segment address. All data lying between start segment X 16 and next 256 K bytes will be picked up. Segment range is 0 to FFFF.
2. In MOTOROLA S HEX format,the start address represents the actual start address. All data lying between start address and next 256 K bytes will be picked up. Address range is 0 to FFFFFFFF.

for ex.:

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


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

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

```
A> HEXBIN2 <CR>

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

4.2 File Dump to Console

Not many ROM code BINARY files can be displayed on screen by DOS TYPE command. DUMP.EXE can convert BINARY file to HEXADECIMAL character and display them to the console or printer. Although it is meaningless on paper, some designers need to keep a copy for later use.

The input command under DOS command prompt is :

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

[] : option, <CR> : return key or enter key.

[^P] : Ctrl + P
It will connect PC with printer, and data displayed on screen will be outputed to printer.

FILENAME : standard file name that is specified by DOS.

start address : HEXADECIMAL digit
Start dumping from this address to end of the file.

4.3 2-way or 4-way file splitter

SPLIT2.EXE can split a 16 bits source file into two 8 bits files. One is the collection of data lying on LOW

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

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

The input command under DOS command prompt is :

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

[] : option, <CR> : return key or enter key.

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

APPENDIX A. EPROM MANUFACTURER,TYPE,VPP

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

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

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

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

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

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