VECTOR

CP/M-86

User's Manual

VECTOR 4 CP/M-86

USER'S MANUAL

P/N 73ØØ-ØØØ2
Rev. ØØ

May 2, 1983

Copyright 1983 by Vector Graphic Inc.

Made in U.S.A.

Copyright 1983 by Vector Graphic Inc.
All rights reserved.

Disclaimer

Vector Graphic makes no representations or warranties with respect to the contents of this manual itself, even if the product it describes is covered by a warranty or repair agreement. Furthermore, Vector Graphic reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Vector Graphic to notify any person of such revision or changes, except when an agreement to the contrary exists.

Revision Numbers and Updates

The date, revision and part number of each pages herein appears on the bottom line of the page. The revision number (which may range from ØØ to 99) is appended to the eight-digit Vector Graphic part number by a hyphen. The date and revision numbers on the Title Page correspond to that of the page most recently revised. In addition, there is included a sheet listing the latest revision level and date of each page in the manual. THIS MANUAL SHOULD BE USED ONLY WITH PRODUCT(S) IDENTIFIED ON THE TITLE PAGE.

TRADEMARK

CP/M-86 is a registered trademark of **Digital Research.**

PAGE REVISION LEVEL AND DATE

Page	Rev. Level		Page	Rev. Level		Page	Rev. Level	
i	ØØ	Ø5/Ø2/83	V-10	ØØ	Ø5/Ø2/83	IX-1	ØØ	Ø5/Ø2/83
ii	ØØ	Ø5/Ø2/83	V-11	ØØ	Ø5/Ø2/83	IX-2	ØØ	Ø5/Ø2/83
iii	ØØ	Ø5/Ø2/83	V-12	ØØ	Ø5/Ø2/83	IX-3	ØØ	Ø5/Ø2/83
iv	ØØ	Ø5/Ø2/83	V-13	ØØ	Ø5/Ø2/83	IX-4	ØØ	Ø5/Ø2/83
v	ØØ	Ø5/Ø2/83	V-14	ØØ	Ø5/Ø2/83	IX-5	ØØ	Ø5/Ø2/83
vi	Blank		V-15	ØØ	Ø5/Ø2/83	IX-6	ØØ	Ø5/Ø2/83
vii	ØØ	Ø5/Ø2/83	V-16	ØØ	Ø5/Ø2/83	IX-7	ØØ	Ø5/Ø2/83
viii	ØØ	Ø5/Ø2/83	V-17	ØØ	Ø5/Ø2/83	IX-8	ØØ	Ø5/Ø2/83
ix	ØØ	Ø5/Ø2/83	V-18	ØØ	Ø5/Ø2/83	IX-9	ØØ	Ø5/Ø2/83
x	Blank		V-19	ØØ	Ø5/Ø2/83	IX-1Ø	ØØ	Ø5/Ø2/83
			V-20	ØØ	Ø5/Ø2/83	IX-11	ØØ	Ø5/Ø2/83
I-1	ØØ	Ø5/Ø2/83	V-21	ØØ	Ø5/Ø2/83	IX-12	ØØ	Ø5/Ø2/83
I-2	ØØ	Ø5/Ø2/83				IX-13	ØØ	Ø5/Ø2/83
I-3	ØØ	Ø5/Ø2/83	VI-1	ØØ	Ø5/Ø2/83	IX-14	ØØ	Ø5/Ø2/83
I-4	ØØ	Ø5/Ø2/83	VI-2	ØØ	Ø5/Ø2/83	IX-15	ØØ	Ø5/Ø2/83
I-5	ØØ	Ø5/Ø2/83	VI-3	ØØ	Ø5/Ø2/83	IX-16	ØØ	Ø5/Ø2/83
I-6	ØØ	Ø5/Ø2/83	VI-4	ØØ	Ø5/Ø2/83	IX-17	ØØ	Ø5/Ø2/83
			VI-5	ØØ	Ø5/Ø2/83	IX-20	ØØ	Ø5/Ø2/83
II-1	ØØ	Ø5/Ø2/83	VI-6	ØØ	Ø5/Ø2/83	IX-19	ØØ	Ø5/Ø2/83
II-2	ØØ	Ø5/Ø2/83	VI-7	ØØ	Ø5/Ø2/83	IX-21	ØØ	Ø5/Ø2/83
II-3	ØØ	Ø5/Ø2/83	VI-8	ØØ	Ø5/Ø2/83	IX-22	ØØ	Ø5/Ø2/83
II-4	ØØ	Ø5/Ø2/83				IX-23	ØØ	Ø5/Ø2/83
II-5	ØØ	Ø5/Ø2/83	VII-1	ØØ	Ø5/Ø2/83	IX-24	ØØ	Ø5/Ø2/83
II-6	ØØ	Ø5/Ø2/83	VII-2	ØØ	Ø5/Ø2/83	IX-25	ØØ	Ø5/Ø2/83
			VII-3	ØØ	Ø5/Ø2/83	IX-26	ØØ	Ø5/Ø2/83
III-1	ØØ	Ø5/Ø2/83	VII-4	ØØ	Ø5/Ø2/83	IX-27	ØØ	Ø5/Ø2/83
III-2	ØØ	Ø5/Ø2/83	VII-5	ØØ	Ø5/Ø2/83	IX-28	ØØ	Ø5/Ø2/83
III-3	ØØ	Ø5/Ø2/83	VII-6	ØØ	Ø5/Ø2/83	IX-29	ØØ	Ø5/Ø2/83
III-4	ØØ	Ø5/Ø2/83	VII-7	ØØ	Ø5/Ø2/83	IX-3Ø	ØØ	Ø5/Ø2/83
III-5	ØØ	Ø5/Ø2/83	VII-8	ØØ	Ø5/Ø2/83	IX-31	ØØ	Ø5/Ø2/83
III-6	ØØ	Ø5/Ø2/83	VII-9	ØØ	Ø5/Ø2/83	IX-32	ØØ	Ø5/Ø2/83
III-7	ØØ	Ø5/Ø2/83	VII-1Ø	ØØ	Ø5/Ø2/83	IX-33	ØØ	Ø5/Ø2/83
III-8	ØØ	Ø5/Ø2/83	VII-11	ØØ	Ø5/Ø2/83	IX-34	ØØ	Ø5/Ø2/83
			VII-12	ØØ	Ø5/Ø2/83	IX-35	ØØ	Ø5/Ø2/83
IV-1	ØØ	Ø5/Ø2/83	VII-13	ØØ	Ø5/Ø2/83	IX-36	ØØ	Ø5/Ø2/83
IV-2	ØØ	Ø5/Ø2/83	VII-14	ØØ	Ø5/Ø2/83	IX-37	ØØ	Ø5/Ø2/83
IV-3	ØØ	Ø5/Ø2/83				IX-38	ØØ	Ø5/Ø2/83
			VIII-1	ØØ	Ø5/Ø2/83	IX-39	ØØ	Ø5/Ø2/83
V-1	ØØ	Ø5/Ø2/83	VIII-2	ØØ	Ø5/Ø2/83	IX-4Ø	ØØ	Ø5/Ø2/83
V-2	ØØ	Ø5/Ø2/83	VIII-3	ØØ	Ø5/Ø2/83	IX-41	ØØ	Ø5/Ø2/83
V-3	ØØ	Ø5/Ø2/83	VIII-4	ØØ	Ø5/Ø2/83	IX-42	ØØ	Ø5/Ø2/83
V-4	ØØ	Ø5/Ø2/83	VIII-5	ØØ	Ø5/Ø2/83	IX-43	ØØ	Ø5/Ø2/83
V-5	ØØ	Ø5/Ø2/83	VIII-6	ØØ	Ø5/Ø2/83	IX-44	ØØ	Ø5/Ø2/83
V-6	ØØ	Ø5/Ø2/83	VIII-7	ØØ	Ø5/Ø2/83	IX-45	ØØ	Ø5/Ø2/83
V-7	ØØ	Ø5/Ø2/83	VIII-8	ØØ	Ø5/Ø2/83	IX-46	ØØ	Ø5/Ø2/83
V-8	ØØ	Ø5/Ø2/83	VIII-9	ØØ	Ø5/Ø2/83	IX-47	ØØ	Ø5/Ø2/83
V-9	ØØ	Ø5/Ø2/83	VIII-1Ø	ØØ	Ø5/Ø2/83	IX-48	ØØ	Ø5/Ø2/83
			VIII-11	ØØ	Ø5/Ø2/83			

PAGE REVISION LEVEL AND DATE

Page	Rev. Level	
IX-49	ØØ	Ø5/Ø2/83
IX-5Ø	ØØ	Ø5/Ø2/83
IX-51	ØØ	Ø5/Ø2/83
IX-52	ØØ	Ø5/Ø2/83
X-1	ØØ	Ø5/Ø2/83
X-2	ØØ	Ø5/Ø2/83
X-3	ØØ	Ø5/Ø2/83
X-4	ØØ	Ø5/Ø2/83
X-5	ØØ	Ø5/Ø2/83
X-6	ØØ	Ø5/Ø2/83
X-7	ØØ	Ø5/Ø2/83
X-8	ØØ	Ø5/Ø2/83
X-9	ØØ	Ø5/Ø2/83
X-1Ø	ØØ	Ø5/Ø2/83
X-11	ØØ	Ø5/Ø2/83
X-12	ØØ	Ø5/Ø2/83
X-13	ØØ	Ø5/Ø2/83
X-14	ØØ	Ø5/Ø2/83
X-15	ØØ	Ø5/Ø2/83
X-16	ØØ	Ø5/Ø2/83
X-17	ØØ	Ø5/Ø2/83
X-18	ØØ	Ø5/Ø2/83
X-19	ØØ	Ø5/Ø2/83
XI-1	ØØ	Ø5/Ø2/83
XI-2	ØØ	Ø5/Ø2/83
XI-3	ØØ	Ø5/Ø2/83
XI-4	ØØ	Ø5/Ø2/83
XI-5	ØØ	Ø5/Ø2/83
XI-6	ØØ	Ø5/Ø2/83
XI-7	ØØ	Ø5/Ø2/83
XII-1	ØØ	Ø5/Ø2/83
XII-A-1	ØØ	Ø5/Ø2/83
XII-A-2	ØØ	Ø5/Ø2/83
XII-A-3	ØØ	Ø5/Ø2/83
XII-A-4	ØØ	Ø5/Ø2/83
XII-A-5	ØØ	Ø5/Ø2/83
XII-B-1	ØØ	Ø5/Ø2/83
XII-B-2	ØØ	Ø5/Ø2/83
XII-B-3	ØØ	Ø5/Ø2/83
XII-B-4	ØØ	Ø5/Ø2/83
XII-B-5	ØØ	Ø5/Ø2/83
XII-B-6	ØØ	Ø5/Ø2/83
XII-C-1	ØØ	Ø5/Ø2/83
XII-C-2	ØØ	Ø5/Ø2/83
XII-C-3	ØØ	Ø5/Ø2/83
XII-C-4	ØØ	Ø5/Ø2/83
XII-C-5	ØØ	Ø5/Ø2/83
XII-C-6	ØØ	Ø5/Ø2/83
XII-C-7	ØØ	Ø5/Ø2/83
XII-C-8	ØØ	Ø5/Ø2/83
XII-C-9	ØØ	Ø5/Ø2/83
XII-C-1Ø	ØØ	Ø5/Ø2/83
XII-C-11	ØØ	Ø5/Ø2/83
XII-C-12	ØØ	Ø5/Ø2/83
XII-C-13	ØØ	Ø5/Ø2/83
XII-C-14	ØØ	Ø5/Ø2/83
XII-C-15	ØØ	Ø5/Ø2/83
XII-C-16	ØØ	Ø5/Ø2/83
XII-D-1	ØØ	Ø5/Ø2/83
XII-D-2	ØØ	Ø5/Ø2/83
XII-D-3	ØØ	Ø5/Ø2/83
XII-D-4	ØØ	Ø5/Ø2/83
XII-E-1	ØØ	Ø5/Ø2/83
XII-E-2	ØØ	Ø5/Ø2/83
XII-E-3	ØØ	Ø5/Ø2/83
XII-E-4	ØØ	Ø5/Ø2/83
XII-E-5	ØØ	Ø5/Ø2/83
XII-E-6	ØØ	Ø5/Ø2/83
XII-E-7	ØØ	Ø5/Ø2/83
XII-E-8	ØØ	Ø5/Ø2/83
XII-E-9	ØØ	Ø5/Ø2/83
XII-E-1Ø	ØØ	Ø5/Ø2/83
XII-F-1	ØØ	Ø5/Ø2/83
XII-F-2	ØØ	Ø5/Ø2/83
XII-F-3	ØØ	Ø5/Ø2/83
XII-F-4	ØØ	Ø5/Ø2/83
XII-F-5	ØØ	Ø5/Ø2/83
Index-1	ØØ	Ø5/Ø2/83
Index-2	ØØ	Ø5/Ø2/83
Index-3	ØØ	Ø5/Ø2/83
Index-4	ØØ	Ø5/Ø2/83

Page	Rev. Level

FOREWORD

Audience The **User's Manual for Vector 4 CP/M-86** is intended for first-time and experienced users of Vector 4 computer systems and the CP/M-86 Operating System.

Scope The **User's Manual** gives complete information on use of the Vector 4 and CP/M-86. It tells how to install and begin working with the computer, describes system commands, and presents an overview of CP/M-86 and Vector Graphic's Disk Partition System.

Organization The **User's Manual** has twelve sections. These fall into two main parts:

* Sections I through VI are for the first-time user of computers or of the Vector 4 system. They introduce the Vector 4 system, show how to set up it up, and describe the most frequently used CP/M-86 commands.

* Sections VII through XII are for more experienced users. They give further information about CP/M-86 and Vector Graphic's Disk Partition System, provide a complete guide to system commands, and show how to install an operating system on the hard disk. Section XII contains appendices, including information on the use of diskettes and on troubleshooting.

The **User's Manual** is accompanied by the **User's Guide to CP/M-86** by Digital Research, Inc., the creator of CP/M-86. That manual describes all basic CP/M-86 programs. Because CP/M-86 has been customized for the Vector 4, it differs in important ways from what is described in the Digital Research manual. In particular, Vector Graphic has created universal versions of CONFIG, FORMAT, SYSGEN, and COPYDISK commands to be used in place of the standard CP/M-86 programs and also for any other operating system installed on Vector 4 systems.

Refer to this **User's Manual** as your guide to CP/M-86 for the Vector 4, and see the Digital Research **User's Guide** for further information about CP/M-86 commands, including the utilities for advanced programming that are not covered here.

NOTE

In Sections I through VI, it is assumed that CP/M-86 has been installed on Vector 4s with a hard disk. Normally, installation is done by your dealer or by an experienced member of your staff.

If CP/M-86 has not been installed, consult your dealer or see Section XI for information on converting your system.

TABLE OF CONTENTS

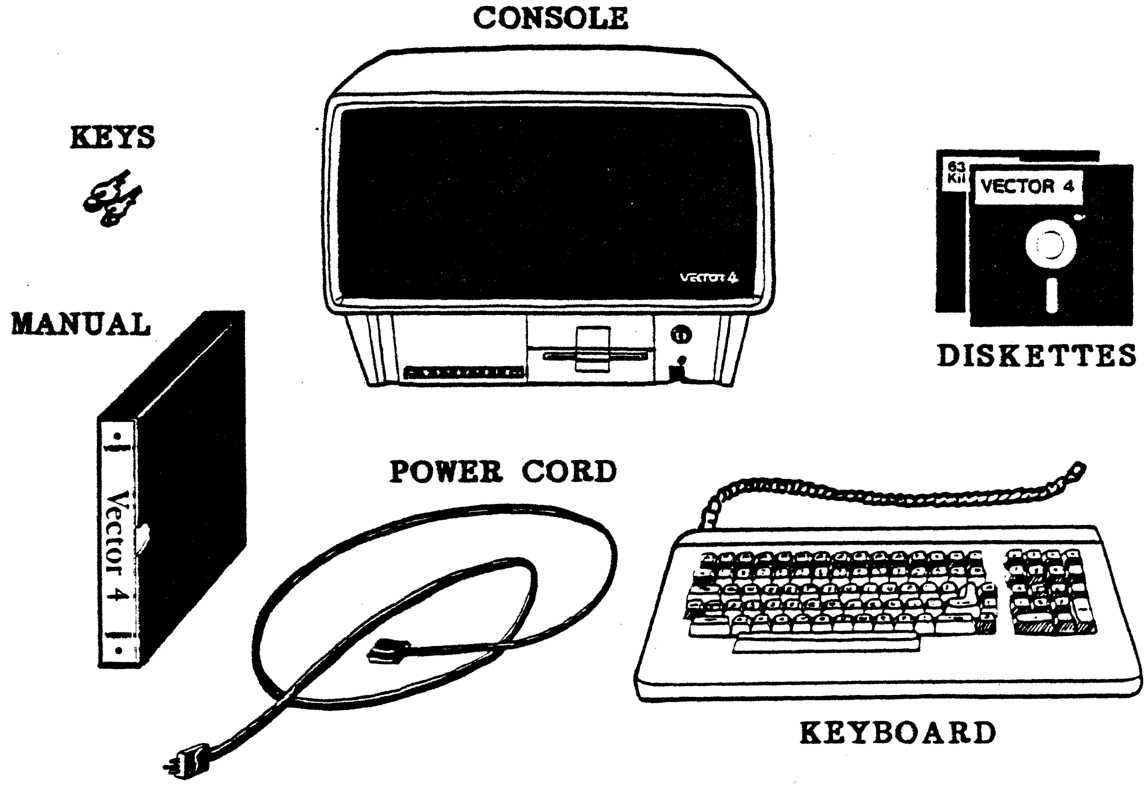
SECTION I

MEET YOUR VECTOR 4

The Components Of Your System

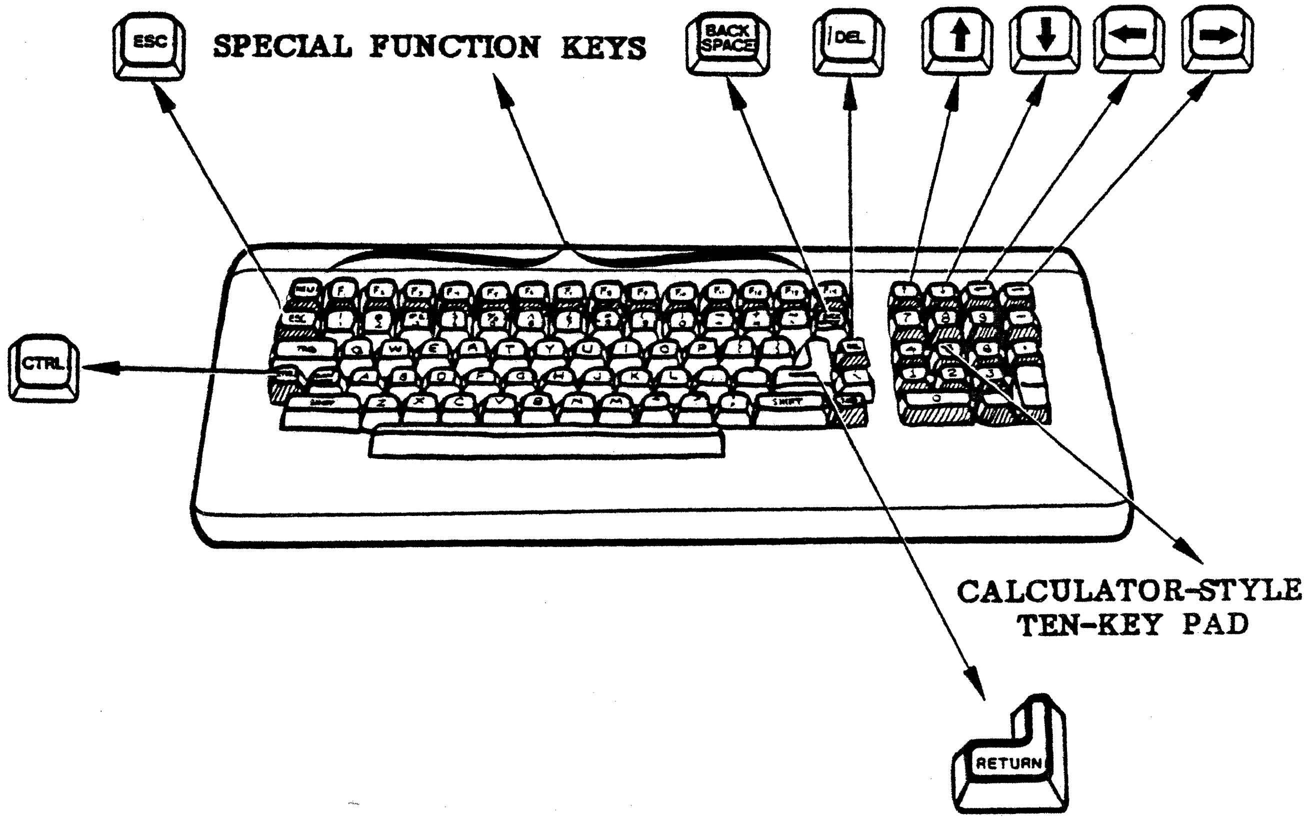
Your Vector 4 computer system is made up of several components that work together to process and store information. Before you go on, take a moment to look at what these components do.

The console is the part of the Vector 4 with the video screen. Inside the console are the many electronic parts that make up the system, including the microprocessor. Beneath this console are two disk drives. In front of the console is a separate keyboard.



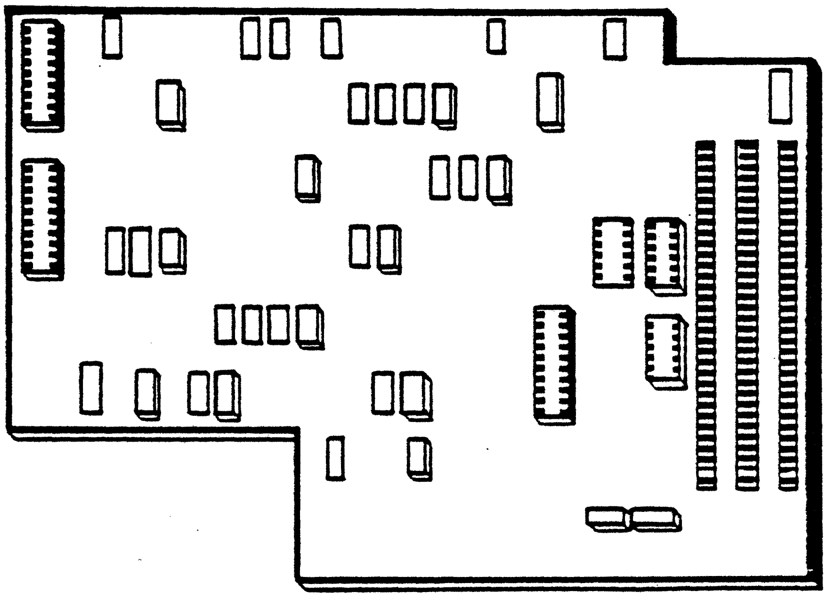
The Keyboard

You will use the keyboard to give instructions to your Vector 4. The layout of the keyboard is very much like that of an electric typewriter. There are the usual typewriter keys and, in addition, a ten-key calculator pad for rapid entry of numbers, 15 special function keys, and several special keys such as [DEL] and [ESC]. All these extra keys will help you in working with the computer. They will be explained later.



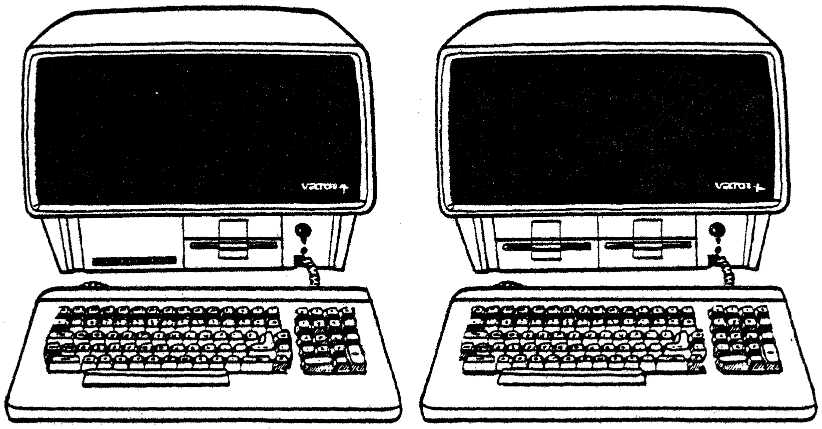
The Microprocessor

The microprocessor "chip" is the most important component of the computer. Your Vector 4 has and can use either of two different microprocessor chips. Whichever one is in use is responsible for making decisions, performing calculations, and executing the instructions or commands that you give it. The microprocessors, along with other electronic devices, are mounted on a printed circuit board inside the console.



The Disk Drives

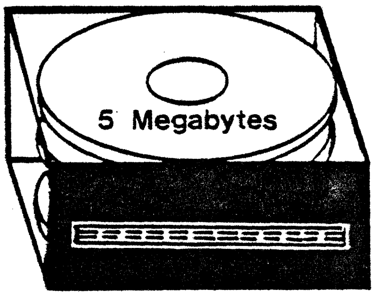
Depending on its configuration (the arrangement of its parts), your Vector contains one or two kinds of disk drives. Look at the lower part of you computer's console (the cabinet that holds the screen). It will look like on of the two illustrations below:



The system on the right has two floppy-diskette drives The diskette drive to the right is called Drive A; the drive to the left is called Drive B.

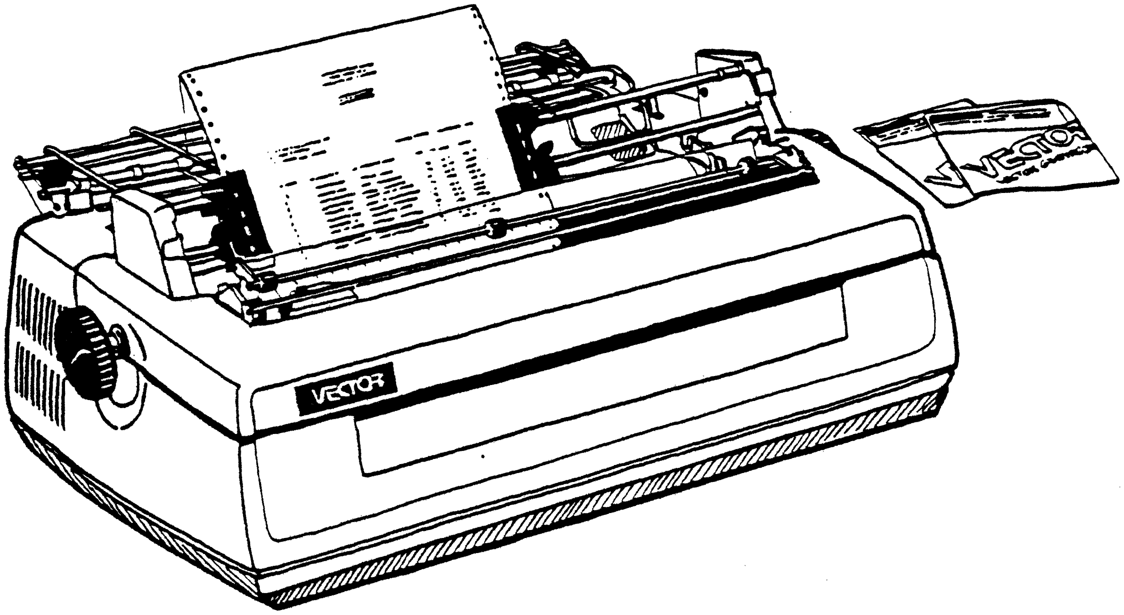
The system on the left has one floppy-diskette drive and one hard-disk drive. The hard disk holds very large quantities of information and, as you will learn later, it can be "divided" into parts that the computer "thinks" about it as if they were several separate disks.

Both kinds of drives are used for storing permanent copies of your work.



The Printer

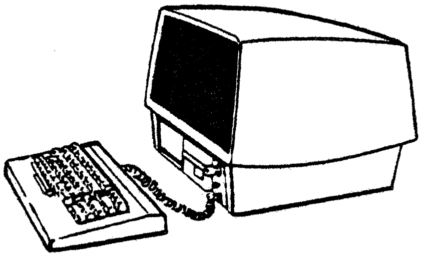
A printer provides permanent copies of your work. Although not truly part of your computer—it is really an accessory—the printer is an important part of your system. The printer allows you to produce "hard copy" of your work.

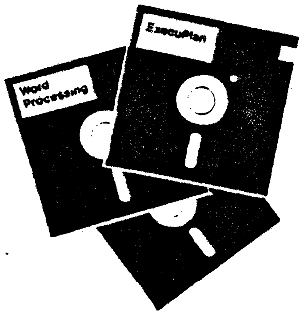


The Hidden Component

The hidden component of your computer system is its software. (What w have been describing so far is called hardware.) Software means programs, th instructions that tell the hardware components what to do. These instruction direct the hardware to "print this character at the printer," "display thi number on the screen," or "look at the keyboard to see if anything has bee typed"—and much more.

Software is stored on the hard disk in the console and on diskettes. Diskette can be easily exchanged, so you can change software as easily as changing record.



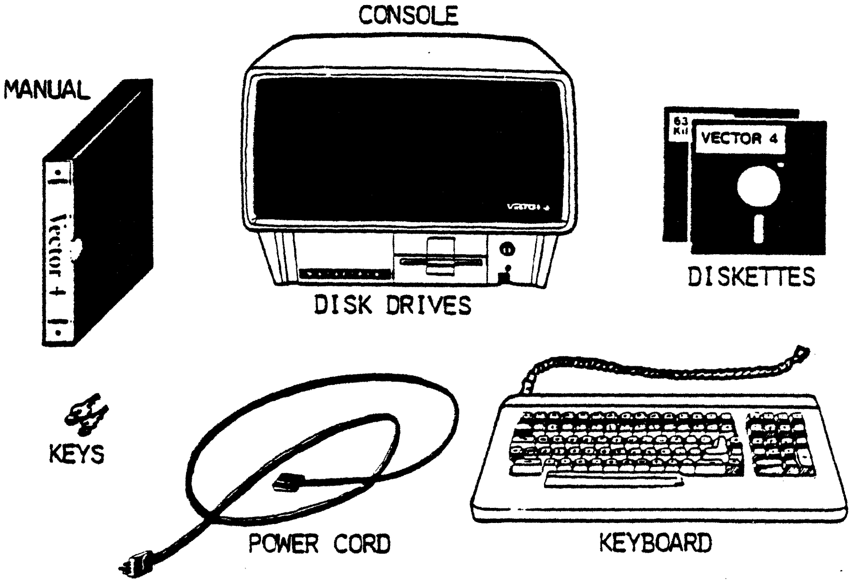


SECTION II

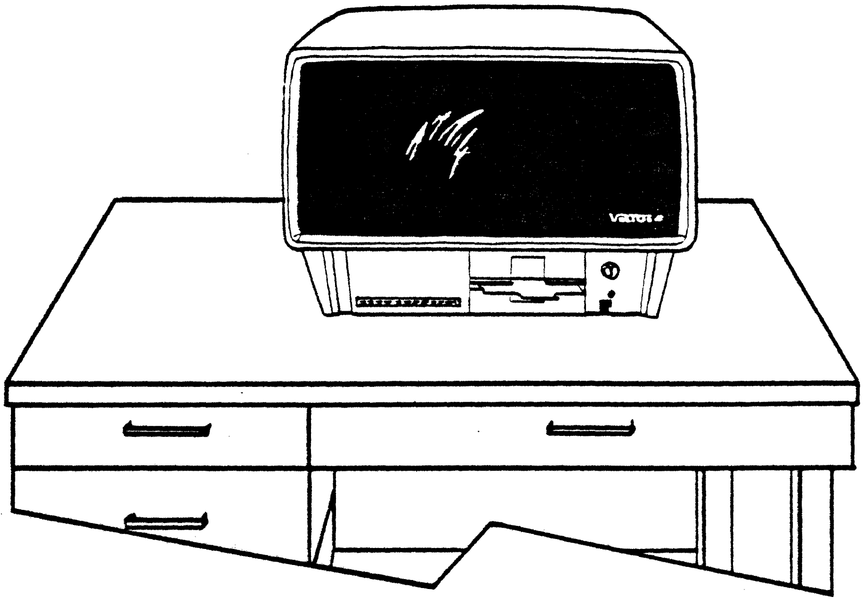
INSTALLATION INSTRUCTIONS

Connecting Your Vector 4 Components

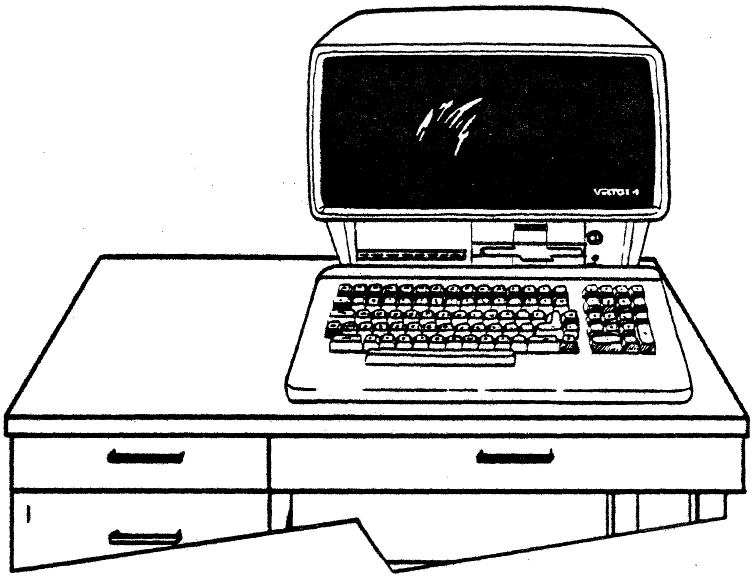
Your Vector 4 packing box should contain the following components:



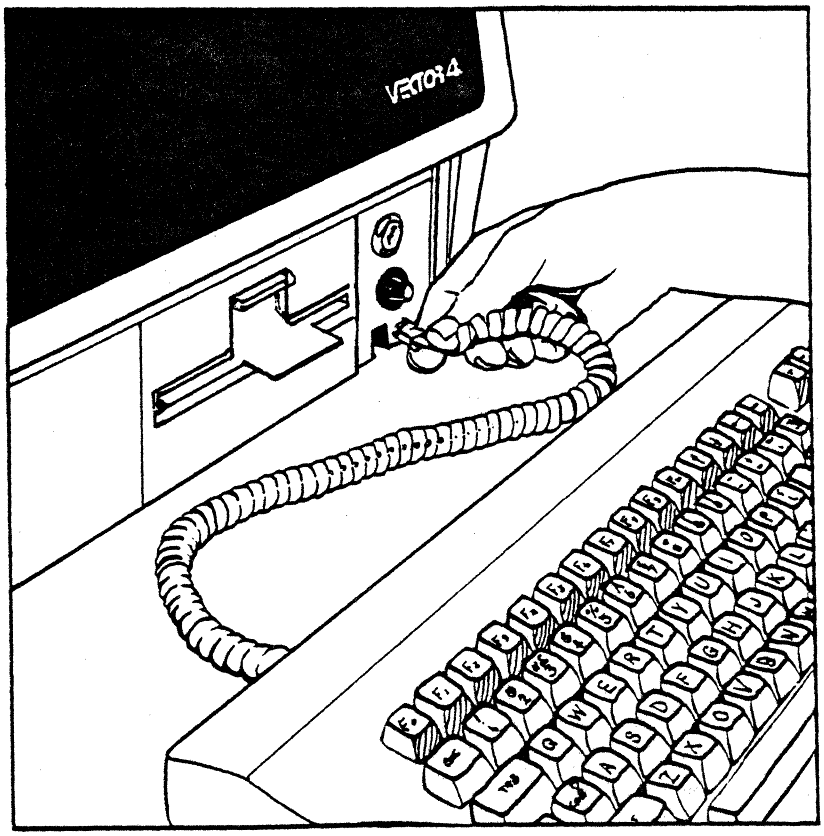
1. Place the console on a flat and sturdy surface.



2. Place the keyboard in front of the console.



3. Push the end of the keyboard connector into the socket on the lower right front of the console. Listen for the connector to "click" into place.



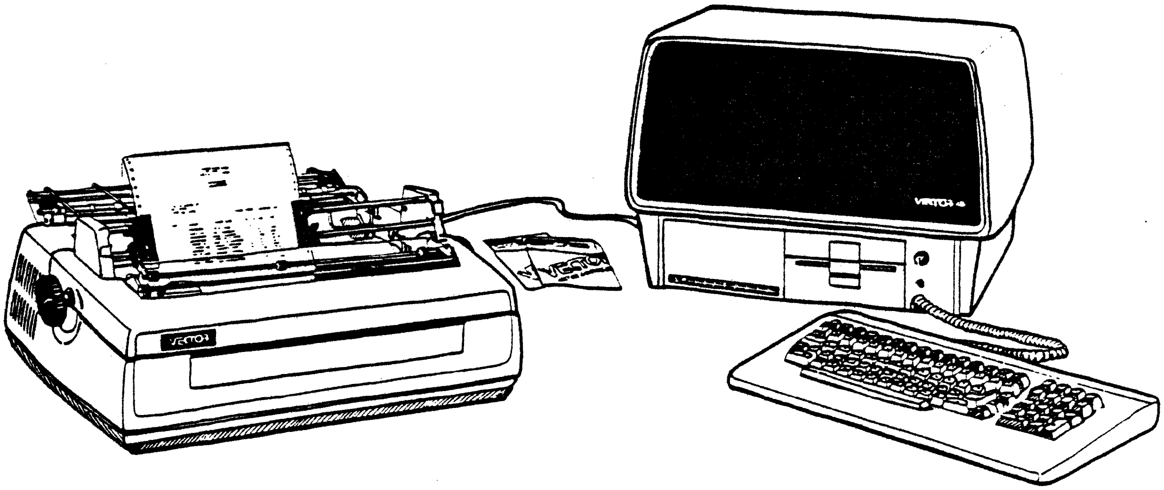
Connecting The Printer To Your Vector 4

Your Vector 4 is designed to use either a Vector 7700 or Vector 3500 Letter Quality Printer. The first step in setting up your printer is connecting its cables. After that, you need to configure your computer system (make adjustments to its software) to allow it to operate the printer you have attached. See Section V, "Choosing Printers," for information.

If you will not be installing a printer, go to the next heading, "Connecting Your Vector 4 to Power."

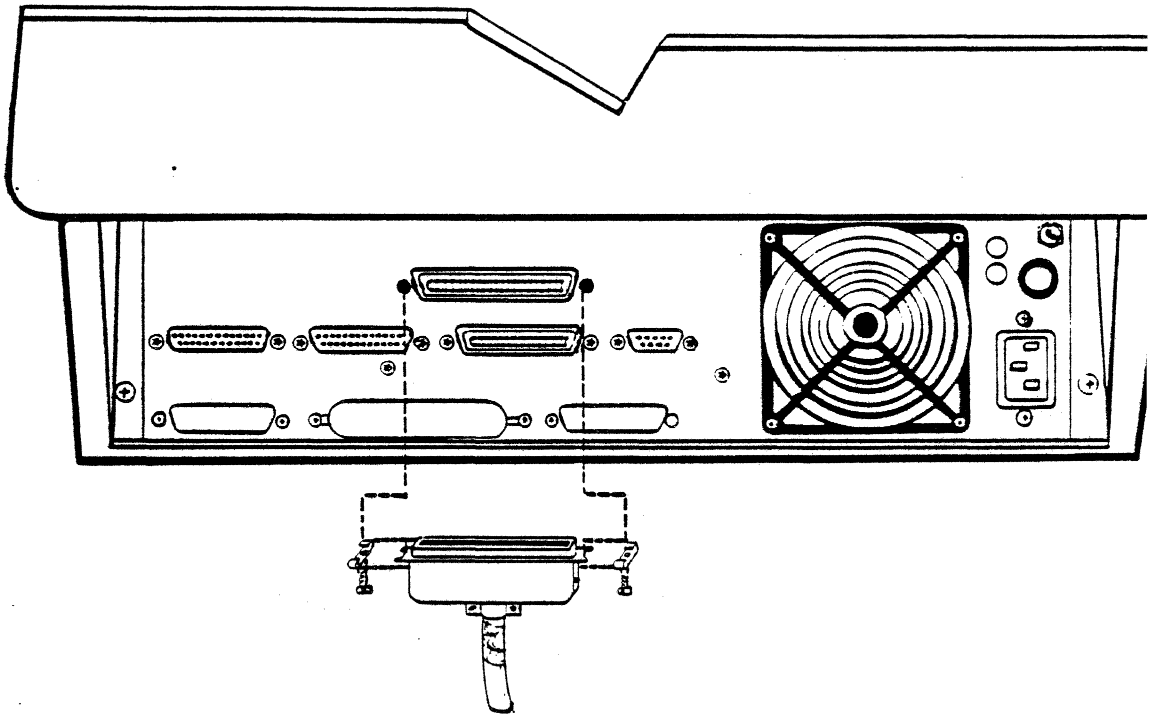
Installing The Printer

1. Place the printer on flat and sturdy surface, one to five feet from your Vector 4.

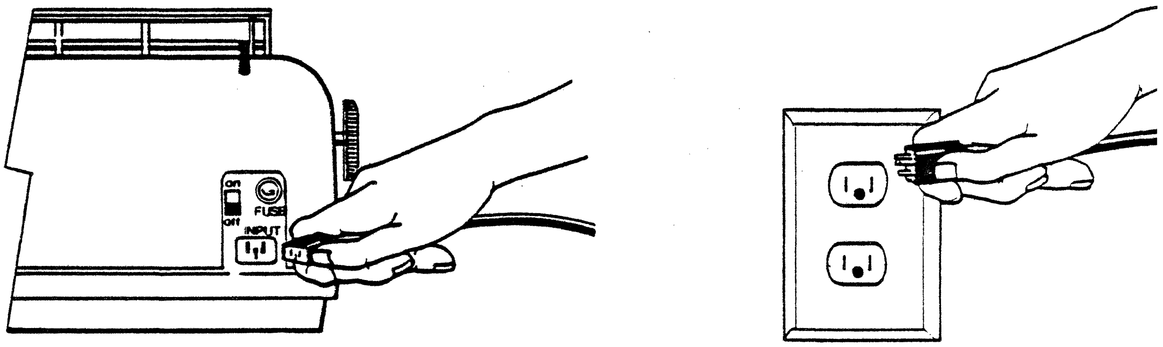


(Vector 7700 Shown)

2. Firmly insert the connector end of the printer ribbon cable to the larger of the two sockets on the rear of the console (see illustration).

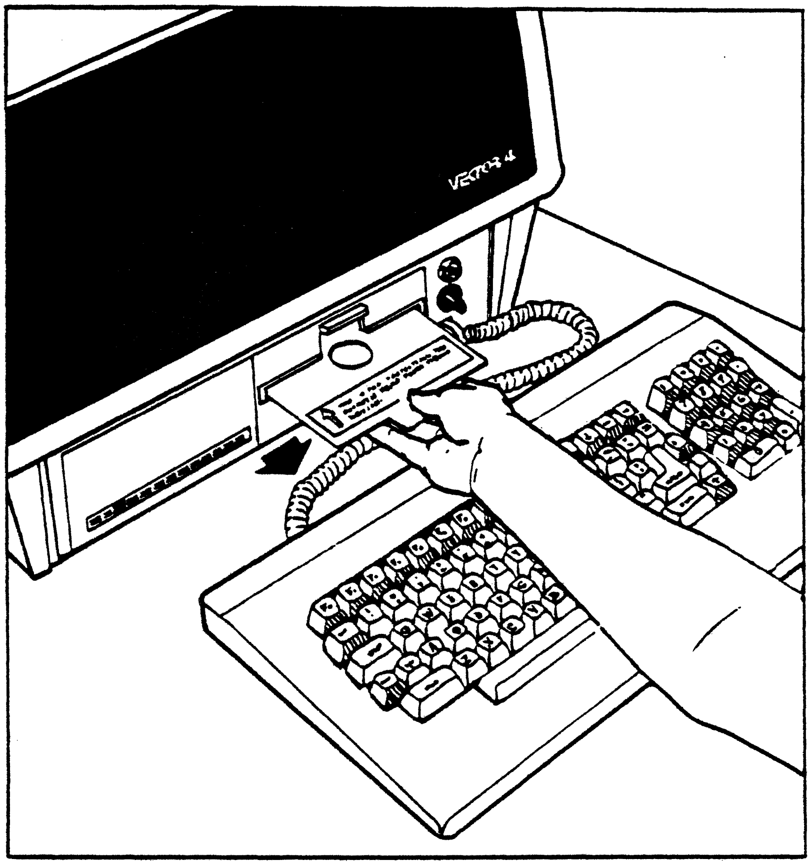


3. Connect the printer's power cord to the three-prong socket on the rear of the printer and to a grounded AC outlet.

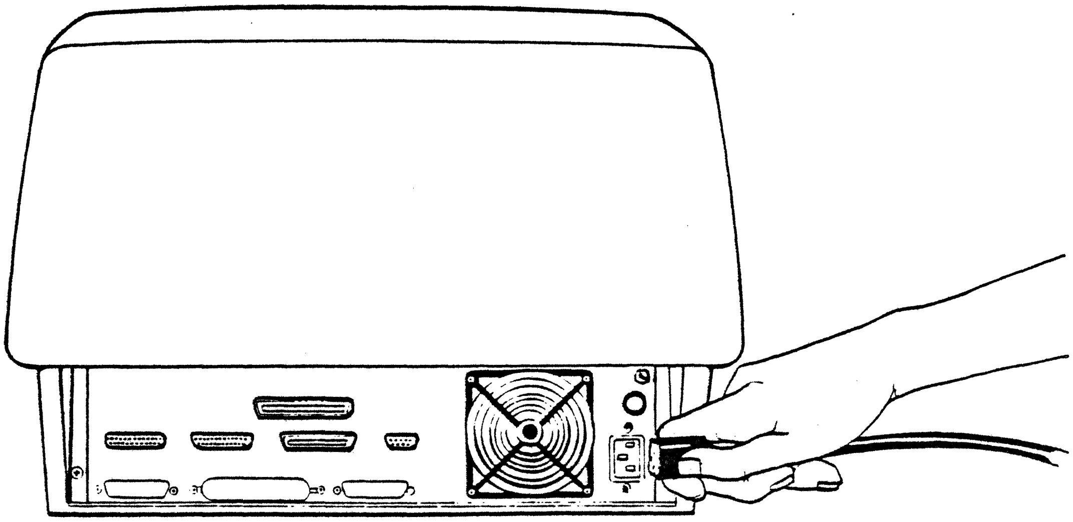


Connecting Your Vector 4 to Power

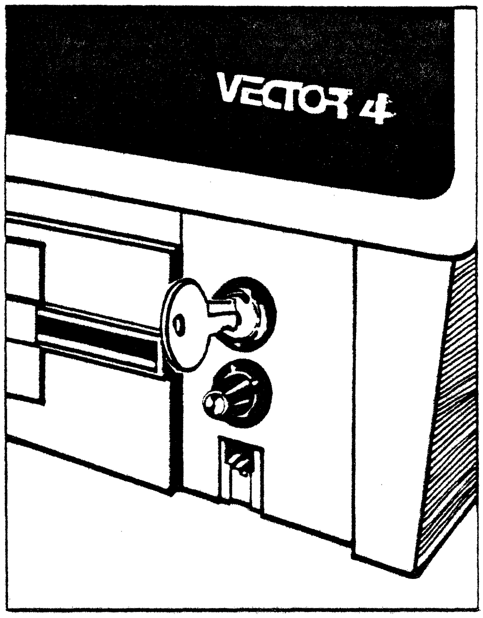
1. Remove the cardboard insert from the diskette drive or drives by pulling the tab marked **REMOVE THIS AT OPERATION** straight out from the drive.



2. Connect the Vector 4's power cable to the 3-prong connector at the rear of the computer.

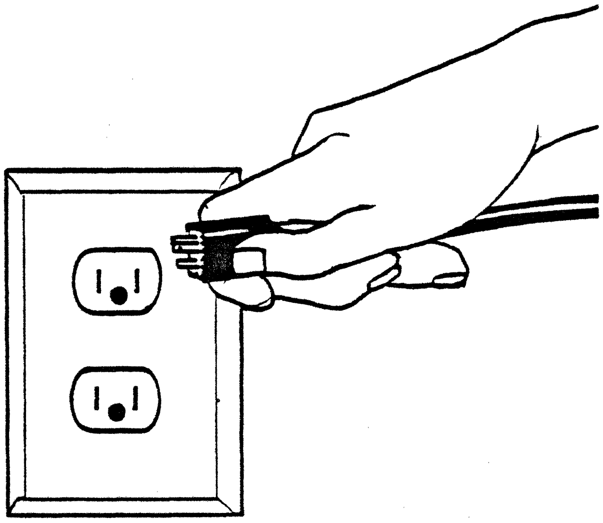


3. Make sure that the Vector 4's power key on the front panel is in a vertical position. If necessary, turn the key to the vertical position.



To assure proper operation, connect your Vector to a dedicated power line—that is, one which is not shared with high-load machines such as photocopiers or air conditioners. For more details, read the information on "Power Source" in Appendix A.

4. Connect the power cable to an appropriate wall outlet.

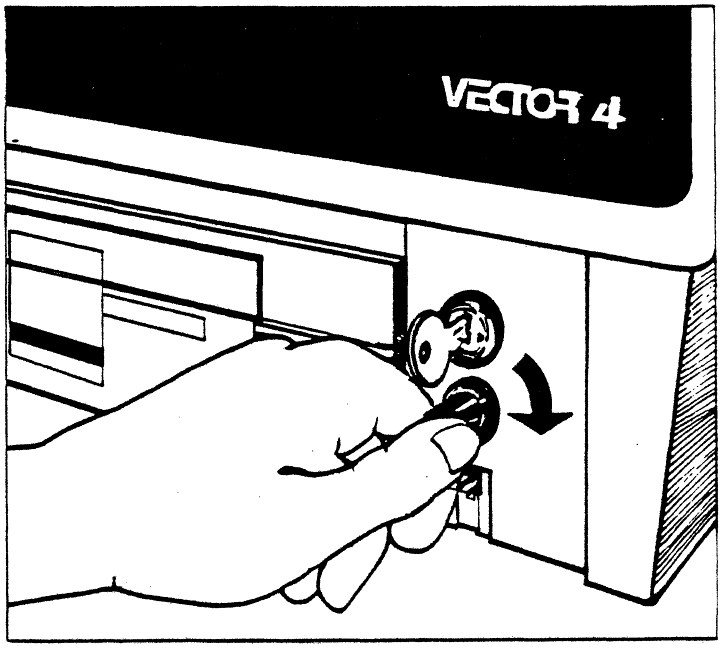


SECTION III

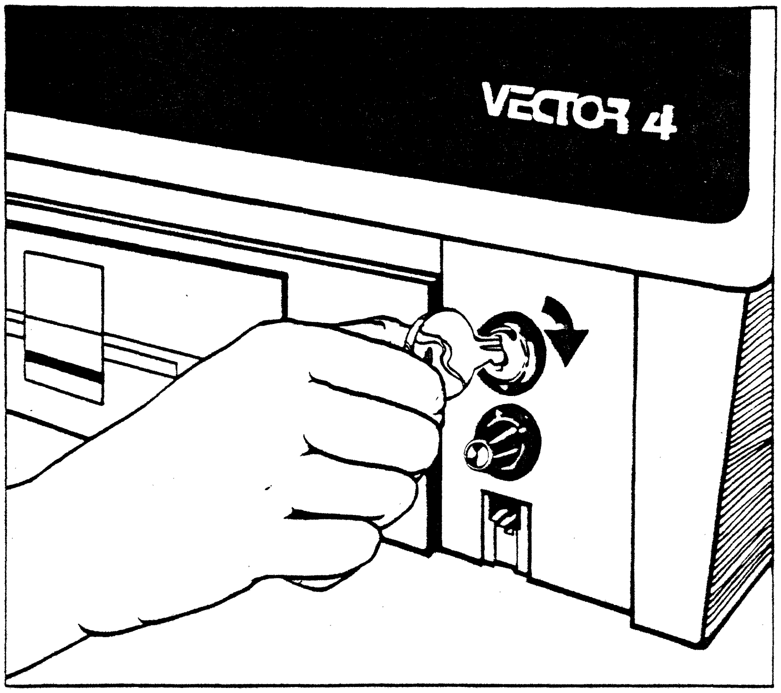
STARTING UP YOUR VECTOR 4

Turning On the Computer

1. Turn the contrast knob clockwise.



2. Insert the key into the lock on the front of the computer and turn clockwise. The console will "beep" to indicate power on.



The console will take approximately 30 seconds to warm up.

3. When the console warms up, this message appears on the screen:

```
VECTOR GRAPHIC V4/MM  [C]
--------------------------------
SBC EXECUTIVE - REVISION V.VV (XX)


...TESTING...
```

In your display, "MM" in the top line is replaced by the model number of your Vector 4 system--for example, "3Ø" (for V4/30) or "4Ø" (for V4/40). "C" is replaced by a letter representing the "configuration" of your system: the kind and number of its disk drives.

"V.VV" and "XX" in the second line also are replaced by numbers and letters representing the version of your system's software.

After a few moments, the line "...TESTING..." is replaced by

```
...PLEASE WAIT...
```

If the message does not appear after several minutes:

- o Turn the system off,
- o Wait 60 seconds,
- o Repeat the steps in this section, beginning with Step 1.

If the message does not appear after repeated tries, refer to Appendix C, "User's Guide to Troubleshooting The Vector 4," for information.

A more detailed explanation of the automatic self-test feature follows in this chapter.

4. If your system includes a hard disk and CP/M-86 has been installed, the message is replaced after a moment by the Systems Menu. It shows what operating systems are available on the disk, and it varies depending on what actually is installed on the disk. It may look like this:

```
[F] .. LOAD SYSTEM FROM FLOPPY

[1] .. LOAD CP/M-86 SYSTEM FROM LOCAL WINCHESTER
```

   Or it may include additional options:

```
[F] .. LOAD SYSTEM FROM FLOPPY

[N] .. LOAD SYSTEM FROM NETWORK

[1] .. LOAD CP/M-86 FROM LOCAL WINCHESTER

[2] .. LOAD MS-DOS 2.0 FROM LOCAL WINCHESTER
```

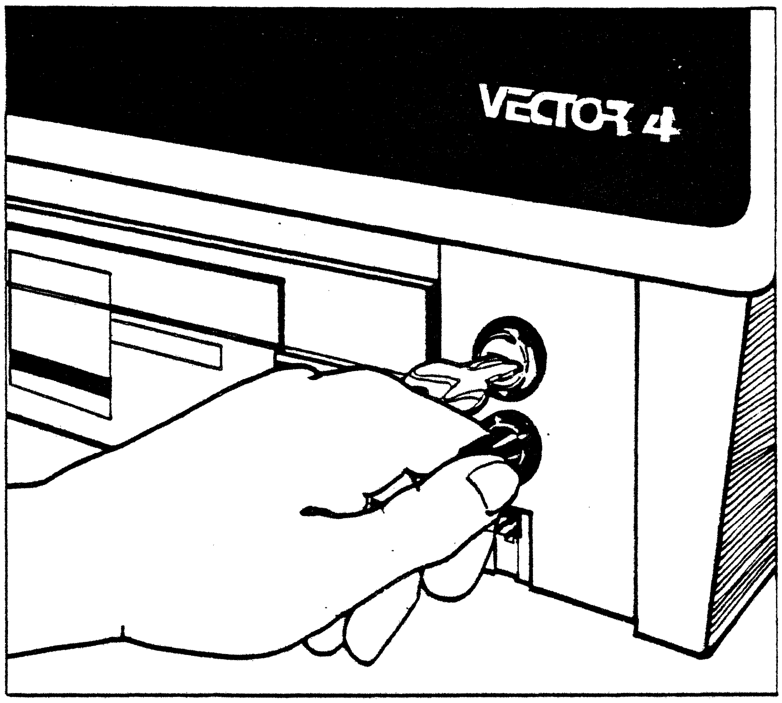
5. If no operating system has been installed on the hard disk, or if your system has diskette drives but no hard disk, the Basic Menu appears instead:

```
VECTOR GRAPHIC V4/MM  [C]
---------------------------------
SBC EXECUTIVE - REVISION V.VV (XX)


[W] .. LOAD SYSTEM FROM WINCHESTER
[F] .. LOAD SYSTEM FROM FLOPPY
[T] .. TEST ENVIRONMENT

 -
```

6. Adjust the screen display to suit lighting conditions and your personal taste. Turning the contrast control knob on the front of the computer clockwise makes the screen brighter; turning it counterclockwise makes the screen dimmer.



NOTE

For either system type, if the screen displays an error message, refer to Appendix C, "User's Guide to Troubleshooting the Vector 4."

Using Diskettes

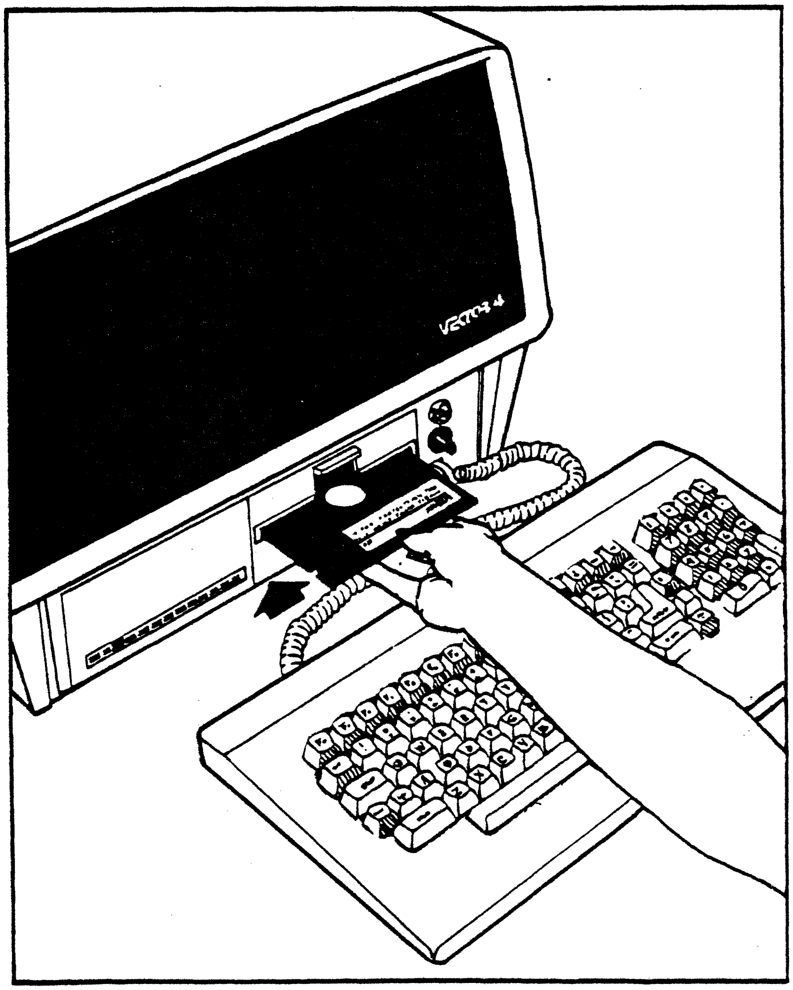
Diskettes are an important part of your system. All information handled by your Vector 4 will be stored either on diskettes or on the hard disk. Improper use of diskettes can cause loss of valuable information.

This section shows the proper methods for using diskettes. If you have never used diskettes before, read this section before continuing. Even if you plan to store most of your work on the hard-disk drive, knowing how to handle diskettes is important because you will periodically back up information from the hard disk onto diskettes.

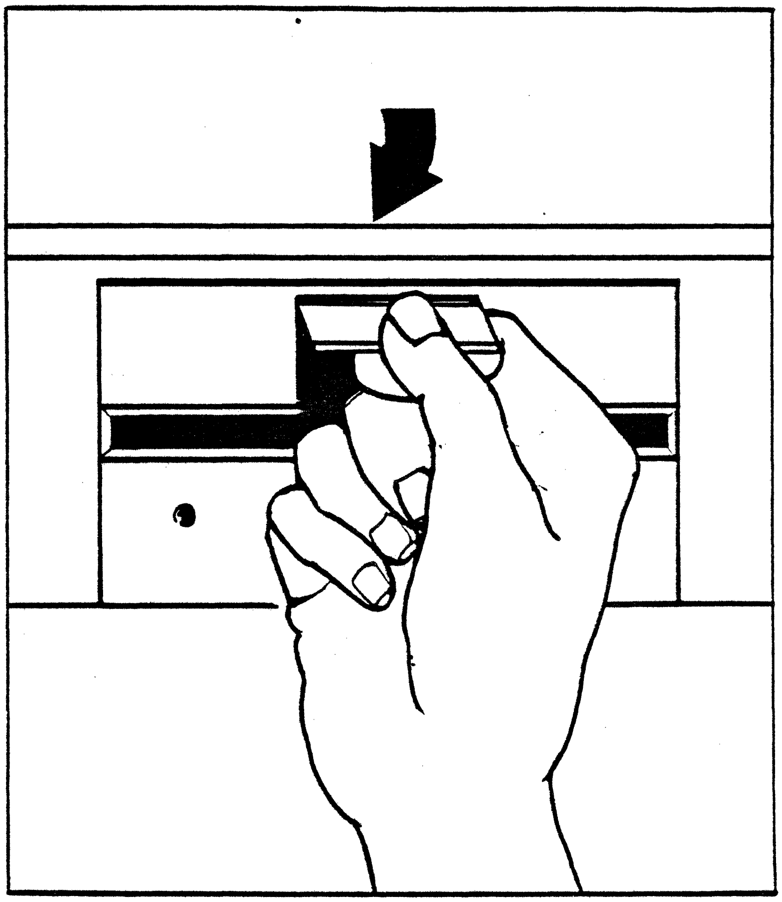
Loading Diskettes

There are two steps to loading a diskette: inserting and mounting.

1. Insert the diskette with the label side up, the square notch in the edge pointing to the left, and the oblong area of exposed disk media pointing inward. Press lightly against the diskette until it seats itself in the drive. Do not push so hard that the diskette bends.



2. The diskette is now inserted but not yet mounted. To mount the diskette, push down on the door of the drive until you feel increased resistance at the half-way point, then STOP. Continue pushing slowly until the door "clicks" and stays closed.



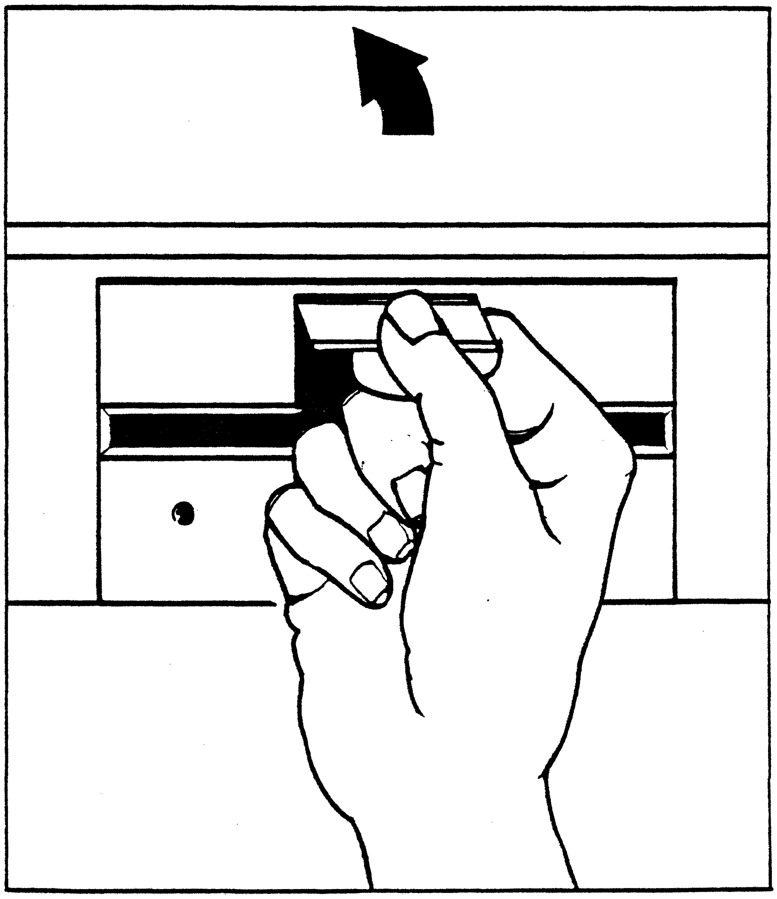
If you cannot fully close the door, the diskette is not inserted properly. Open the door, remove the diskette, and repeat Step 1.

Once mounted, the diskette is ready for use. When the diskette drive is in use, its red "activity" light glows. When the drive is no longer in use, the red light goes out.

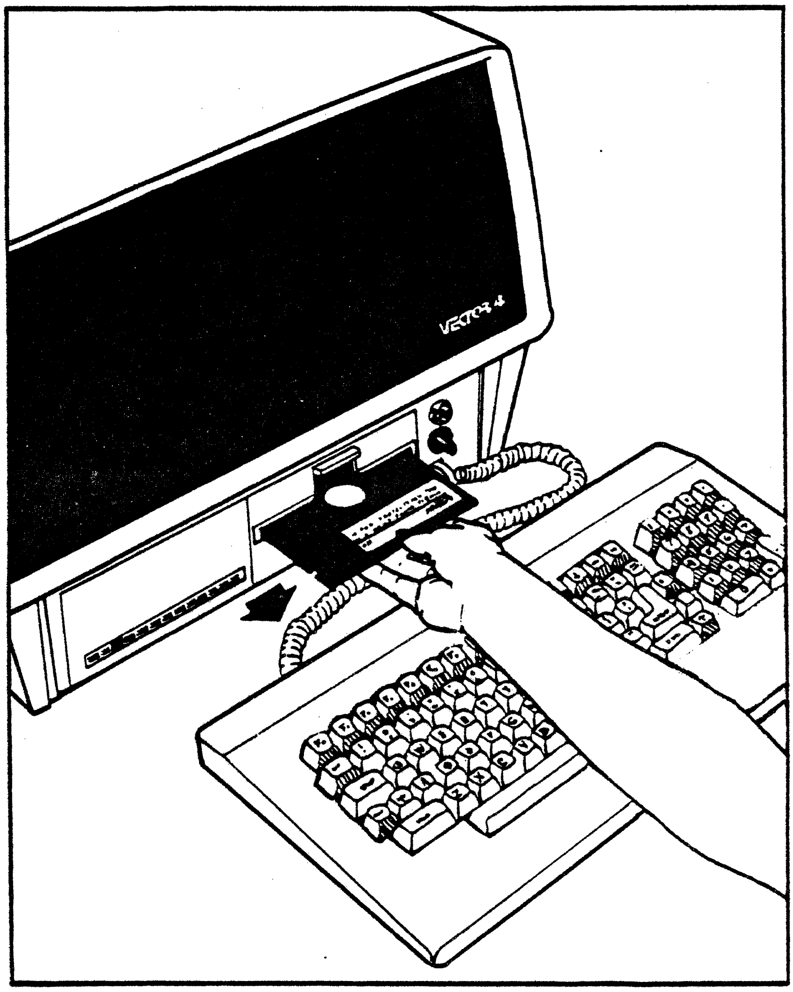
Unloading Diskettes

To unload a 5-1/4-inch diskette:

1. Make sure that the red light is off. Slowly open the door by swinging the panel upward.



2. Gently pull the diskette straight out from the drive. It should move out easily. If there is any resistance, be careful not to pull too hard. Excessive force can harm the surface.



It is best to remove the diskette when it is not in use. This reduces wear of the diskette against its protective jacket. You may leave it inserted but not mounted (that is, with the door to the drive open), or simply take the diskette out of the drive and replace it in its protective storage envelope. Refer to Appendix C, "Care and Handling Of Diskettes," for more information.

SECTION IV

BOOTING CP/M-86

To process information, your Vector 4 relies on a special program of instructions called the operating system. This is CP/M-86: the instructions that run the system. Now that you've powered up the computer, the next step is to bring in CP/M-86. This is done by "booting."

When to Boot

If your system is diskette-based (no hard disk), you need to boot whenever the Basic Menu is displayed on the screen:

```
VECTOR GRAPHIC V4/MM   [C]
-----------------------------------
SBC EXECUTIVE - REVISION V.VV (XX)


[W] .. LOAD SYSTEM FROM WINCHESTER
[F] .. LOAD SYSTEM FROM FLOPPY
[T] .. TEST ENVIRONMENT

    -
```

If your system has a hard disk, you need to boot whenever you see the Systems Menu on the screen:

```
[F] .. LOAD SYSTEM FROM FLOPPY

[1] .. LOAD CP/M-86 SYSTEM FROM LOCAL WINCHESTER
```

Or:

```
[F] .. LOAD SYSTEM FROM FLOPPY

[N] .. LOAD SYSTEM FROM NETWORK

[1] .. LOAD CP/M-86 FROM LOCAL WINCHESTER

[2] .. LOAD MS-DOS 2.0 FROM LOCAL WINCHESTER
```

(Remember, the Systems Menu varies depending on what operating systems are installed on your hard disk.)

Booting Hard-Disk Systems

If your system has a hard disk and CP/M-86 is already installed, select from the Options Menu.

* The first option, "Load System from Floppy," is offered in case you still want to boot from a diskette. (See the section below, "Booting Floppy-Disk Systems.")

* The second option, "Load System from Network," allows you to boot into Vector Graphic's LINC communications network if this is use.

* The fourth option allows booting from another operating system offered by Vector Graphic, MS-DOS 2.0.

* The third option, "Load CP/M-86 System From Local Winchester," is the one you will use now. (If one of the others is clearly what you intend, see the Vector Graphic user's manual for that system. To boot CP/M-86 from a floppy diskette, see below.) The "local Winchester" is the hard disk in your console.

1. To boot CP/M-86 from the hard disk, press **[1]**.

   The menu disappears and the screen displays:

   VECTOR GRAPHIC CP/M-86 REVISION V.VV (XX) Y(Y)

   A>

Booting Floppy-Disk Systems

To boot floppy-based systems (or if CP/M-86 has not yet been installed on your hard disk):

1. Insert the Vector Graphic System Diskette into Drive A (the right hand drive) and close the drive door.

2. Press **[F]**.

   The message disappears and the screen displays:

   VECTOR GRAPHIC CP/M-86 REVISION V.VV (XX) Y(Y)

   A>

CP/M-86 At Your Command

Whichever way you followed above, your system is now ready. Turn to the next section to learn how to use CP/M-86 commands.

If anything different from the above happens, see Appendix C, "User's Guide to Troubleshooting The Vector 4," for information.

The Automatic System Test

One of the important features of the Vector 4 is its Automatic System Test. You will notice that the screen displays the message—

...Testing...

—when you first turn on the power or press the RESET button. The Vector 4 performs three automatic test programs every time you start up.

These programs check—

- the ability of the memory to retain information accurately;
- the system executive program for any errors;
- the keyboard, to ensure that it is correctly attached and functioning.

This helps ensure that your Vector 4 is functioning properly.

If the automatic test uncovers any errors, it will display one of the three following messages:

```
MEM ERR, BLK xx, EVEN/ODD ADDR, MASK = xx
PROM CS ERROR
KBD ERROR
```

"xx" is replaced by a two-digit number.

If any of these messages appear when you turn your system on, see Appendix C, "User's Guide To Troubleshooting The Vector 4."

SECTION V

INTRODUCTION TO CP/M-86

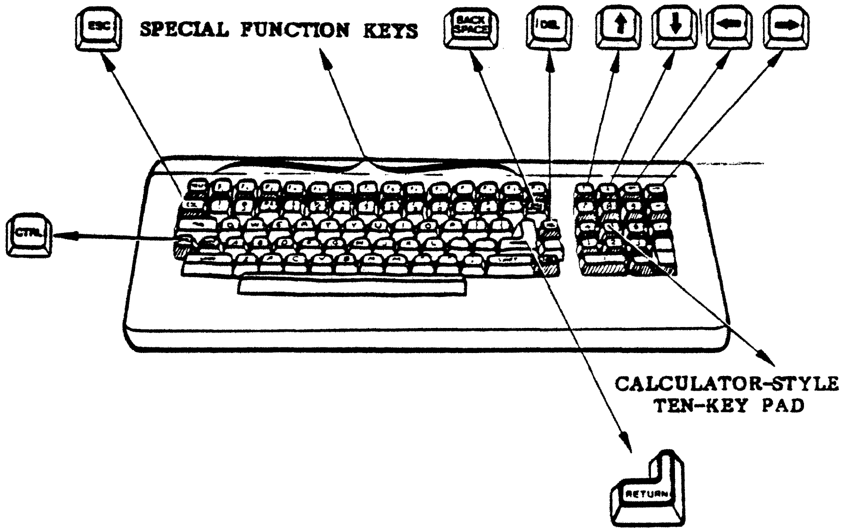
Vector 4 Operating Commands

CP/M-86 has two kinds of commands: screen-handling commands and program commands.

* Screen-handling commands are one- or two-letter entries for correcting what you have typed at the keyboard and see on the screen. They also allow you to start and stop some programs.

* Program commands call programs, both system and application.

All commands are entered at the keyboard. The keys on your Vector 4 keyboard are shown below:



Return key	**[RETURN]**	Escape key	**[ESC]**
Control key	**[CTRL-X]**. "X" stands for another letter key pressed while **[CTRL]** is held down.		
Tab key	**[TAB]**	Space Bar	**[SPACE]**
Backspace	**[BACKSPACE]**	Delete key	**[DEL]**
Line Feed	**[LF]**	Shift	**[SHIFT]**
Arrows	**[UP ARROW], [DOWN ARROW], [LEFT ARROW], [RIGHT ARROW].**		
Function keys	**[F n]** where n = the number of the function key to press.		

How Commands Are Shown

Your Vector 4 is programmed to expect certain commands, and it acts upon those entered in the right way. It can recognize entries that are correctly "spelled," but if it meets with a wrong form, it's stuck: all it can do is repeat your entry followed by a question mark.

To make clear the right spelling, commands are shown in this manual using the following symbols:

> The screen-handling commands (and any one- or two-key entry) are enclosed in square brackets—[ ].

(The brackets aren't part of the command, so don't type them with the key itself. They simply show that [RETURN] means the key with the word "RETURN" printed on it.

> Program commands are shown by format, like this:

```
A>ERA ufn <options>  [RETURN]
```

A format line shows the parts of a command without actually filling in the "blanks."

* It begins with the command name (here, ERA, for ERAse).

* Then comes one or more required parts. In the example, "ufn" is required: it shows what CP/M-86 needs to carry out the command. ("Ufn" stands for "unambiguous file name." You need to state the name of the file to be erased.)

* Third, optional entries may be included. Optional items are enclosed in angle brackets—< >.

* [RETURN] is the last step in every command. It sends the command to processing.

Often in using commands you will give a drive letter—such as A: or B:—to show CP/M-86 where it should look for a program or file. The drive letter is entered with a colon (:) and goes in front of the filename. For example:

```
ERA B:OLDFILE.MEM [RETURN]
```

In examples, command entries are shown in UPPER CASE. But in actual use you may enter commands either in upper or lower case.

All commands are described in full, with examples, in Sections VIII, IX, and X.

The Screen Handling Commands

These commands can be used to edit typing entries and control the screen display.

SCREEN HANDLING COMMANDS

To Do This...	Enter This...
Erase a mis-typed character	[BACKSPACE]
Erase an entire line	[CTRL X]
Clear screen before executing command line entered.	[CTRL D]
Start printer from keyboard: press once to start, again to stop	[CTRL P]
Pause scrolling of display: press once to stop, again to resume.	[CTRL S]

"Emergency Brakes"

Sometimes you may want to stop a program before it reaches its normal conclusion. These "emergency brake" commands force most programs on your Vector 4 to quit. You then return to the CP/M-86 command level (the A> prompt).

"EMERGENCY BRAKES"

To Do This...	Enter This
Stop most programs	**[CTRL C]**
Stop certain programs	**[ESC]**
Reset the entire system **NOTE:** RESET clears memory: All work since the last "write-to-disk" command is lost. In addition, after RESET you must REBOOT the operating system.	**Press RESET (rear of console)**

The Program Commands

Program commands allow you to use both system programs--the CP/M-86 commands and special Vector utilities--and application programs such as as Memorite or ExecuPlan.

Some program commands work any time you have booted CP/M-86, no matter what logical drive you are in (see more about logical drives below). Others must be on the drive or diskette whose letter appears on your screen prompt (for example, **A>**). The first type of commands are called "built-in," because they are in the part of CP/M-86 loaded into internal memory during booting. The second type are called "transient," because they remain on the disk until called for use. They only "visit" the internal memory.

The Electronic Filing Cabinet

Most of the time spent with your Vector 4 will be devoted to working with information stored in compartments on a disk called files. A computer file, like a file in an office drawer, is a collection of related information stored under a given name. All data stored on your disk or diskettes is organized in files.

There are two kinds of files: program files and data files. Of these two, you will work inside just the second, the data files. Data files store your information: lists of customers or business letters to vendors. In contrast, program files hold instructions for the computer (the transient commands just mentioned are examples of program files). Generally, they can be changed only by a skilled programmer. You will never change the contents of program files, but you will use them to move, copy, rename, delete, and check on the other type of file: your data files.

All CP/M-86 files have names formed following a few simple rules.

* Each file is identified by a two-part file name

* The first part is the name. It can be up to eight characters long, letters or numbers (it must begin with a letter).

* The second part is the extension. It can be one to three characters long.

* The two are separated by a period.

Even if your system is new, it already includes many files: the command files for your system programs. To see that this is true, type **DIR** after the **A>** prompt:

A>**DIR [RETURN]**

The DIR (Directory) command brings to the screen a list of the files stored on your current drive (Drive A). Notice the file names, eight letters or fewer, then a space, then the extension.

The difference between a file's name and its extension is this: The name identifies the file and is used by CP/M-86 to find the file in memory. The extension tells CP/M-86 what kind of file it is.

Wild Cards

Keep the directory on your screen. Notice that several files have an extension of CMD. These are the command files mentioned above, such as the transient commands and other program files.

Suppose that later, when you have data files, you want to look at a directory listing of only .CMD files. You have a quick way of doing this, using "wild card" symbols. In card games, wild cards are those that can stand for any other card. In CP/M-86, wild-card characters can stand for another letter or group of letters.

On your Vector 4, the wild cards are the asterisk (*) and the question mark (?).

The question mark (?) takes the place of one character. It means: "Any thing here is acceptable." The asterisk (*) can stand for several characters, from the position where it appears up to the end of the "field"—for example, up to eight characters for a file name, or up to three for an extension.

Why Use Wild Cards?

Wild cards give you a simple way to look at groups of files that share a similar characteristic, such as type (example: all files with a .MEM extension) or title (example: all files titled JAN83). The wild card lets you enter one command to cover many files.

EXAMPLE: To look at all files with the extension .CMD, enter this command:

```
A>DIR *.CMD [RETURN]
```

This displays only the .CMD (command) files on the Drive A.

EXAMPLE: To look at all Memorite files (these have a .MEM extension):

```
A>DIR *.MEM [RETURN]
```

Getting Around The Disks

If you have been following the instructions up to here, your screen now shows the A> prompt. This says that you are "logged onto" Drive A. You can move to another drive, say Drive B, by entering the letter of the drive followed by a colon. (If you have a floppy-only system, insert a diskette into the left-hand drive before continuing.)

Enter the command:

```
A>B: [RETURN]
```

The screen will display:

```
A>B:
B>
```

If you enter this command with no diskette in the drive, you will meet with your first "error message":

```
ERROR - B: DRIVE NOT READY
```

Just press **[RETURN]** and the **A>** prompt will return.

You also can select a "User Area" within the drive whose letter shows in the screen prompt. User areas are smaller segments of the drives themselves and are used as separate compartments for different kinds of files. At the start, the system puts you in User Area 0. But you have the option of selecting User Area 1, 2, 3, and so on up to User Area 15.

EXAMPLE: To change from User Area 0 of Drive A to User Area 4, type:

```
A>USER 4 [RETURN]
```

To change back to User Area 0, type:

```
A>USER 0 [RETURN]
```

You can also type the User command to find out what user area you currently are in:

```
A>USER [RETURN]
```

The screen displays the number of your user area.

Displaying A File's Name

As we saw, the resident command **DIR** can be used to display a directory of files on your current drive.

We also saw that the CP/M-86 wild cards ? and * can be used to mean "any character" or "any characters." A file name that includes a wild-card symbol is called an **ambiguous file name,** or AFN. In contrast, a file name that can mean only one file (it has no wild-card characters) is called an **unambiguous file name,** or UFN.

Although we used the **DIR** command without filenames, DIR is one of many CP/M-86 commands that accept ambiguous file names in the command line. You can use DIR to check for many files on the disk by entering a command with this format:

```
A>DIR <AFN> [RETURN]
```

EXAMPLE:

```
A>DIR *.CMD [RETURN]
```

RESULT:

The screen displays a listing of all files with an extension of CMD no matter what the file name.

EXAMPLE:

```
A>DIR VECTOR.* [RETURN]
```

RESULT:

The screen displays a directory of all files with the name VECTOR and any extension.

EXAMPLE:

```
A>DIR ST*.COM [RETURN]
```

RESULT:

The screen displays one file, STORE.COM. Its name begins with ST, the wild-card * replaces "ORE," and it has the extension COM.

EXAMPLE:

```
A>DIR ST*.CMD [RETURN]
```

RESULT:

The screen display shows the file STAT.CMD.

Copying Files: The PIP Command

PIP stands for "Peripheral Interchange Program." PIP is a command for moving files. You can use PIP to put programs onto the diskettes for use, and also to make backup copies of files on other diskettes. PIP can copy from one drive to another or from one disk or diskette to another, and also gives the option of changing the names of files as they are transferred.

You can use PIP with ambiguous file names (file names that include wild cards symbols) to copy groups of files at one time.

PIP's format varies with its use, but one thing always stays the same: the order of the "to" and "from" filenames. In every PIP command, you must give—

- FIRST: the name (and disk location) of the "target" file;
- SECOND: the name (and location) of the "source" file;
- THIRD: any options
- LAST, **[RETURN]**

In other words, in PIP commands, the "to" always comes first, the "from" second.

EXAMPLES:

To Copy to Drive B the file FILE.EXT from Drive A.	**PIP B:=A:FILE.EXT**
To copy all files from Drive A to Drive B.	**PIP B:=A:*.***

PIPping Several Files At Once

There are several ways to use PIP to copy groups of files. The simplest is shown just above: use the CP/M-86 wild card symbols (* and ?) to match your command entry against different files on the drive.

EXAMPLE: To copy all the command (.CMD or .COM) files from Drive A to Drive B:

```
A>PIP B:=A:*.CMD [RETURN]
```

And,

```
A>PIP B:=A:*.COM [RETURN]
```

The "repeat mode" of PIP is another way to copy multiple files if wild card symbols cannot easily be used. The repeat mode begins with this command line:

```
A>PIP[RETURN]
```

Type "PIP" and [RETURN] without hitting the [SPACE] bar. After a moment, an * prompt displays. You now can enter any valid PIP command line, without having to repeat the PIP command itself. This saves time. You can still enter ambiguous file names to name groups of files.

EXAMPLE:

```
A>PIP[RETURN]
*B:=A:*.CMD


COPYING:

STAT.CMD
PIP.CMD
(etc.)
```

To end the multiple PIP command line session, enter **[RETURN]** after the * prompt.

```
*[RETURN]
A>
```

Copying Files for Backup

For copying individual files, you have already learned the PIP command. There are two other ways of copying to a floppy diskette. One is for making a backup copy of a source diskette. The other is for backing up the hard disk. Diskettes can be copied using the DISKCOPY command. Backup of the hard disk is done through the STORE and RESTORE programs.

(The STORE/RESTORE commands are described below and in Section X. DISKCOPY is a function of the UTILITY program. See Section IX.)

The STORE Command

The STORE program is used to back-up files from the hard disk to diskettes. (If your system has only diskette drives, you can ignore this section as well as the section on RESTORE.)

It's good to be familiar with STORE before you plan your electronic office system, because the STORE and RESTORE programs have a special requirement: they work only for files with names of seven characters or less. This is because STORE allows copying to multiple diskettes. To keep track of how files continue from one diskette to another, STORE adds a number to the end of each filename. CP/M-86 allows eight characters per filename, but to make room for this "tracking number," one of those eight characters must be made available to STORE.

Remember this if you plan to do backups on diskettes (rather than to a SAFSTORE tape drive, for example).

The command line formats for STORE and RESTORE are virtually identical to that of the PIP program.

EXAMPLE: To copy all files from the Drive A (on the hard disk) to Drive E (one of the diskette drives). Enter this command while logged into Drive A:

```
A>STORE C:=*.* [RETURN]
```

STORE copies all files with names of seven letters or less.

As each diskette fills up, STORE prompts you to insert another. When a file must be continued to a second diskette, STORE adds the symbol $ to the file's name. If the file must be continued again, STORE changes the "$" to "0" and keeps the "$" for the last segment. For each segment of the file, another number is added.

The RESTORE Command

This program puts back on the hard disk the files STORE put on diskettes. Everything described above for STORE also holds true for RESTORE, except that the action is in reverse. RESTORE only works for STOREd files; you cannot use it to recopy files that were backed up the DISKCOPY or PIP.

Since STORE and RESTORE keep track of partial files by the number at the end of the filename, using RESTORE works this way (remember that computer counting begins with 0):

EXAMPLE: To RESTORE a file called SAMPLE.DOC to the hard disk, insert the first of your diskettes and enter the command:

```
A>RESTORE A:=C:SAMPLE0.DOC [RETURN]
```

Notice the "0" at the end of the file name. This begins restoring the entire file SAMPLE.DOC.

RESTORE prompts you to insert another diskette when it has filled up the first, and the process repeats until all your backup diskettes have been copied to the hard disk.

For complete information on STORE and RESTORE, see Section X.

Erasing Files

To remove unneeded files from mass memory, you can erase them using the ERA command. Be careful using ERA: once a file is erased, it cannot be recovered. Erase only files which are truly no longer useful or files which you know have been backed up. Although it sometimes seems convenient to gain new storage space by ERAsing old files, it's usually safer just to buy more diskettes: diskettes are far less expensive than lost work and information.

To erase our sample file from the current drive, enter:

```
A>ERA SAMPLE.DOC [RETURN]
```

When the A> prompt returns, the file has been erased.

If you want to erase a file from Drive B while logged onto Drive A, you have two choices. You can either move from A to B by the usual method (entering the command **B: [RETURN]**), then erase the file. Or you can enter this command:

```
A>ERA B:SAMPLE.DOC [RETURN]
```

The second method saves a little time.

Erasing Groups of Files

The ERA command also accepts ambiguous file names, so you can use the wild-card symbols to erase more than one file at a time. Again, be careful: make sure that the wild-card combination you create does not apply to files you wish to keep.

EXAMPLE: To erase all .MEM (MEMORITE) files from a diskette:

```
A>ERA *.MEM [RETURN]
```

EXAMPLE: To erase all files from a diskette:

```
A>ERA *.* [RETURN]
```

Since this could be disastrous, CP/M-86 gives you one "last chance" to reconsider, asking:

```
ALL FILES (Y-N)?
```

To proceed, enter **Y [RETURN]**. If you realize that the diskette holds valuable information, enter **N [RETURN]** to stop the program.

What If...

...You erase a good file, one you didn't want to erase?

It's lost. So learn to be careful with ERA, and erase only files you know are unimportant. Also, be sure to back up your information. A schedule for backup should be a regular part of any computerized officeplace.

For suggestions on backup schedule and techniques, refer to the section on Diskette Handling in the Appendices. The "Backup Rule Of Thumb" is: Always backup as soon as your mass memory holds more information than you can afford to lose.

Checking Remaining Storage

Once you start working with diskettes, you need to know about how much space you have available. To find out how close you are to the limit of a diskette's storage, you can use the STAT command.

EXAMPLE: To check the remaining area on the currently-logged disk:

```
A>STAT [RETURN]
```

The screen displays—

```
disk: <ATT>, Space: ###k
```

—for each drive that has been used since the last boot.

The symbols "###" are replaced by a number showing the remaining space, in Kbytes (about one-thousand bytes: 1K bytes=1024 characters of storage).

The <ATT> column lists the "attributes" of each drive--that is, whether any limits that have been placed on them. The attributes are:

R/O: Read-only: document cannot be changed or deleted until reset by STAT command.

R/W: Read-write: document can be changed or deleted.

SYS: Changes file to "system" type. It is not listed by the DIR command (it is listed by DIRS).

DIR: Removes "system" status from the file, so that it again appears in directory.

For another use of STAT, you can enter the command:

```
A>STAT d:[RETURN]
```

This brings this display—

```
Bytes remaining on d: ##k
```

—showing information only for the disk requested. In contrast, the **STAT [RETURN]** command gives the status of all drives accessed since the last cold boot.

Running STAT For Files

The uses of STAT shown above are for getting information about drives. You also can use STAT to find the size and attributes for one file or for a number of files. "Size" means how much of the disk surface is dedicated to the particular file. "Attributes" means any restrictions that apply to the file.

As with drives, you also can use STAT to change the attributes of a file: to "Read Only" if you wish to protect it against change or erasure, or "Read/Write" to allow change.

EXAMPLE: To use STAT to check the size and attributes of a file or range of files:

```
A>STAT <AFN> [RETURN]
```

EXAMPLE: To check the status of all .CMD (CP/M-86 program) files:

```
STAT *.CMD
```

The screen gives a display like the following for each file name that matches the AFN:

```
  Recs  Bytes  Ext   Acc
   ###   ###k    #  xxx d:FILENAME.EXT
Bytes remaining on d: ###k

A>
```

The "Recs" column gives the number of "records" (a CP/M-86 storage unit of 128 bytes) that makes up the file.

The "Bytes" column shows how much disk space is used by the file.

The "Ext" column gives the number of "extents" in which the file is stored (one extent = 128 records).

The "Acc" column stands for the access attributes of the file, as described above.

A typical status line for our file SAMPLE.DOC, including the STAT command, might look like this:

```
A>STAT SAMPLE.DOC [RETURN]

  Recs   Bytes  Ext Acc
      27      8k    1  R/W  A:SAMPLE.DOC
Bytes remaining on d: 422k

A>
```

(Most of the information in this status line is useful only in advanced programming.)

Using STAT to Change File Attributes

As we saw, you can use STAT to protect a file or diskette from change or erasure by setting it to "Read-Only." "Read-Only" means that you can use the diskette or file but you can make no changes.

EXAMPLE: To change any attribute for our file SAMPLE.DOC:

```
A>STAT d:SAMPLE.DOC <ATT> [RETURN]
```

Here, <ATT> is replaced by one of the four attributes (R/O, R/W, DIR, or SYS) explained above.

As with other commands, STAT can be used for files stored on a different drive. Enter the letter of the drive in place of the **d:** in the line above.

EXAMPLE: To protect the file SAMPLE.DOC from erasure, when it is stored on Drive B:

```
A>STAT B:SAMPLE.DOC R/O [RETURN]
```

The screen shows:

```
SAMPLE.DOC set to R/O
A>
```

EXAMPLE: To protect an entire disk or diskette the same way (say, a diskette in Drive C):

```
A>STAT C: R/O [RETURN]
```

Using STAT to Create System Files

STAT can be used to set CP/M-86 program files to "SYS" status, which means that they can be used from any user area of a drive so long as they are stored in User Area Ø. Ordinarily, CP/M-86 command files are only locally accessible: if you are working in User Area 5 and want to use the PIP command, there must be a copy of the PIP command file in User Area 5. By setting a "main copy" of PIP in User Area Ø to SYS status, you make it available from every other user area on the drive.

EXAMPLE: To set the CP/M-86 Editor, ED, to SYS status:

```
A>STAT ED.CMD SYS [RETURN]
```

EXAMPLE: To set all .CMD files on Drive B to SYS status:

```
A>STAT B:*.CMD [RETURN]
```

The opposite of SYS status is DIR: a file returned to DIR status is once again only locally accessible. Files may be either SYS or DIR status but not both.

EXAMPLE: To reset the ED command file to DIR status:

```
A>STAT ED.CMD DIR [RETURN]
```

Getting HELP From the System

The HELP command provides a quick, on-screen guide to the use of all CP/M-86 commands and of the Vector Graphic programs, UTILITY and CONFIG. HELP can be entered in two ways: either for a display of all available topics (commands and options), or for information about a specific command.

Entered alone, HELP shows a list of all topics. At the end of the list is a special prompt, **HELP>**, which allows you to request information on a topic by entering its name.

Entered with a command name, HELP gives information on how to use the command.

EXAMPLE: This command—

```
A>HELP [RETURN]
```

—displays the complete list of topics.

EXAMPLE: This command—

```
A>HELP STAT [RETURN]
```

—gives information about the use of the STAT command.

EXAMPLE: For further information about STAT:

```
A>HELP STAT OPTIONS [RETURN]
```

Use HELP in your work with CP/M-86 for quick information about commands and command options.

Setting Up Your System

Although your Vector 4 is ready to work now, there are several options you can and should set to make operation more efficient. These include:

- choosing between two possible printers you may have attached.
- selecting "modes" of use for the keyboard and console screen to suit them to an application program you intend to run
- setting the diskette drive for use of single- and double-sided diskettes

The general name for selecting these options is "configuring" your system. The commands to do it are part of the UTILITY and CONFIG programs. These are programs Vector Graphic provides to supplement the standard CP/M-86 commands.

Choosing Printers

The Vector 4 can support ("knows how to run") three different kinds of printers. To control these, the operating system uses special programs called "printer drivers." A printer driver is a program that sends instructions to advance the paper, type a character, and so on. Any program that uses the printer goes to the selected command file to get the printer services it needs.

For your Vector 4 system, the printer drivers are called List Device files, and CP/M-86 offers three, one for each of three kinds of printers. As the first step in configuring your system, you must inform CP/M-86 which List Device file should be used for the printer you have connected to your computer. This is done using the Printer commands of the CONFIG utility.

See Section IX, "CONFIG Printer Commands," for a description of how to configure your system for the attached printer.

Selecting Keyboard and Video Modes

Two other CONFIG commands allow you to select modes of use for the keyboard and console screen. These modes are ways of handling the "meaning" of each key and how this value is sent to and displayed on the console screen. Different modes are needed because different application programs are based on one mode or the other. For example, most programs in Vector Graphic's Professional Series Applications Packages require a video mode different from other software you may obtain for your system.

These commands are also part of the CONFIG Utility and are part of the Keyboard and Video Commands. You need to set the modes only if you have used one of the earlier Vector Graphic systems--such as the Vector 3--and software designed for it. If your Vector 4 is your first Vector Graphic system, the keyboard and video modes already are correctly set.

See Section IX for descriptions of the Keyboard and Video commands of the CONFIG Utility.

Logical and Physical Drives

Two terms you'll get to know as you work with your system are "logical" and "physical." They are used for the hard-disk and diskette drives, and they have to do with the difference between your system's hardware and software. A physical drive means one of your disk drive units—the circuitry and metal and so on—inside the Vector 4's console. A logical drive is the software that runs that hardware.

This double way of looking at things comes in handy because it allows you to have several software programs (logical drives) to run one hardware unit (physical drive). It gives flexibility. You can use the same diskette drive for two kinds of diskettes (see below), and you can divide your hard disk into different memory areas, called Drive A, Drive B, Drive C, and so on.

This is like having a room that, at one time, is used as nursery, then as a study, then later for storage. It's the same "physical" room, but it is put to different "logical" uses. If you could change the furnishings quickly enough, it would be the same as what you can do with your computer when you change one diskette drive from Drive E (single-sided use) to Drive F (double-sided use), or go from Drive A to Drive B on the same hard disk.

As you'll see in Section VII, the Disk Partition System can use logical divisions to place several completely different operating systems on the hard disk.

Single and Double Sided Diskettes

Your Vector 4 is able to use both single- and double-sided diskettes. If your system has two diskette drives (no hard disk), each drive is already set for use of one kind of diskette: both Drive A (on the right) and Drive B (on the left) are set for double-sided diskettes, and a third logical drive, Drive C (also the left-hand slot) is set for single-sided diskettes.

If your Vector 4 has the diskette drive and a hard-disk drive, you can assign two logical drive names to your diskette drive, one for double-sided diskettes, one for single-sided.

For example, you could set up one logical drive for double-sided diskettes and call it Drive E, then another for single-sided diskettes, called Drive F. For both of these, you would slip the actual diskette into the same physical drive slot. CP/M-86 handles the processing in the right way.

The hard disk usually is divided into logical drives A, B, C, and D, and it always begins with Drive A. On hard-disk systems, Drive A is reserved for the disk.

For hard-disk systems, setting logical diskette drives is part of the FORMAT function. It is done when CP/M-86 is installed on the hard disk. See Section IX, "UTILITY and CONFIG," and Section XI, "Installing CP/M-86 on Your Hard Disk," for information.

Do I Need Both Kinds Of Diskettes?

If your Vector 4 is your first Vector Graphic system, you may not need both kinds of diskettes and drives. Older Vector Graphic computers used single-sided diskettes, and to transfer information from that system to your Vector 4 you need a single-sided drive. Otherwise, double-sided diskettes are preferred: they hold twice the information of single-sided diskette.

Using Your System's Memory

Your diskettes or hard disk provide your permanent record of programs and data. They're not completely permanent, because you can always erase information if you want to. But until you do, the information stays, even when the computer is turned off.

What you see on your screen, on the other hand, is never permanent. It's held only temporarily in what might be called "scratch-pad memory," and it's there so that you can make changes and send information to processing. Once changes have been made, or a program has finished, you must save your work by "writing" the information from your screen back to mass storage, to the hard disk or to a diskette. If your work is new, a new file is created. If your file was old--there was a copy already there—this updated version takes its place and becomes your new permanent copy.

When you bring a file from the disk to your screen, what you see is a copy of what is on the disk. It's something you can work with knowing that an original is still safely stored. It's also something that's never "permanent" until you "write to disk," copying what you see on the screen back to mass storage.

Information in the "scratch-pad" memory stays there either until new information is brought in or until power is turned off. But because RAM can only be so large, new arrivals from mass storage must be "written over" information already there, erasing it.

System Housekeeping with UTILITY

Three of the most commonly-needed commands for maintaining your system are functions of Vector's UTILITY program. They are:

1- SYSGEN: "generates" (copies) an operating system for use on a disk;

2- DISKCOPY: copies the contents of one diskette to another;

3- FORMAT: "cleans up" and prepares the hard disk and floppy diskettes for use.

See Section IX for a full description of the UTILITY program and the three functions.

For More Details...

All of the commands described above and other CP/M-86 commands are discussed at greater length in Sections IX and X. See those sections for further examples of the use of CP/M-86 commands and for description of Vector's UTILITY and CONFIG programs.

SECTION VI

QUICK REFERENCE GUIDE TO VECTOR 4 COMMANDS

Setting Up Your System

To Do This...	Enter This...
Boot from floppy diskette	F
Boot from hard disk	1, for CP/M-86, or select from Systems Menu

NOTE: The Systems Menu varies depending on what operating systems have been installed on your hard disk. See Section IV.

Repeat system testing	T, ESC, or RESET*

NOTE: The above commands apply to a system that has been recently powered up or reset. *On ESC or RESET, screen returns to Basic Menu

To Do This...	Use This Program
Select single- or double-sided configuration for floppy drives (Permits multiple logical configurations for each physical floppy drive)	UTILITY: Format function

To Do This...	Use This Program...
Set up disk or diskette for operation: CAUTION: The Format Function erases all data from the disk or diskette. Use it only if the disk to be formatted holds no valuable information.	**UTILITY: Format Function**
Copy CP/M-86 system to new disk or diskette: Allows booting from the new copy. You must format the disk or diskette before using Sysgen.	**UTILITY: Sysgen Function**
Specify system options and setup:	CONFIG

Entering Commands

To Do This...	Enter This...
Send a command line to be processed.	[RETURN]
Erase a character entered by mistake.	[BACKSPACE]
Erase an entire line.	[CTRL X]

To Do This...	Enter This...
Clear the screen before executing the command line entered.	[CTRL D] [RETURN]
Cancel program operation; returns control to keyboard.	[CTRL C]
Set printer "ON" from keyboard: press once to send entry and screen response to printer, again to cancel.	[CTRL P]
Pause scrolling of display while reading or taking notes: press once to stop, again to resume.	[CTRL S]

"Emergency Brakes"

To Do This...	Enter This
Stop most programs.	[CTRL C]
Stop some programs.	[ESC]
Stop all activity (and lose information in memory) **NOTE:** This erases any new information added since the last "write-to-disk" operation. You also must re-boot the system. On hard-disk systems, if Basic Menu is present, press [W] to return to Systems Menu.	RESET (button at rear of console)

Wild Cards

To Do This...	Enter This...
Refer to several file names (1): . . . CP/M-86 matches any single character in the location, as it finds files on the current drive. Example: **FIL?.EXT** From a disk holding files named FILE.EXT, FILO.EXT, FILL.EXT, FILM.EXT, and FINES.EPL, CP/M-86 selects all but the last as matching FIL?.EXT.	?
Refer to several file names (2): . . . CP/M-86 matches any group of characters in the location. Can be used in combination with regular letters or as *.* to cover all files. Example: **FILE.*** From a disk holding files named FILE.EXT, FILO.MEM, FILL.COM, FILE.ASM, FILE.LD1, CP/M-86 selects all as matching FILE.* The "?" and "*" symbols are "wild cards." File names with wild cards are called ambiguous file names because they refer to more than one file. File names without wildcards are called unambiguous file names, because they refer to only one file (no ambiguity).	*

Copying and Backing Up Files

To Do This...	Enter This...
Make a copy: . . . of **old** file or disk, with name or location of **new.** Enter new name first, then old. Information in old file or disk still exists.	**PIP <new>=<old>**
Examples of commonly used PIP command lines:	
To make a copy on Drive B of the file FILE.EXT from Drive A.	**PIP B:=A:FILE.EXT**
Same as above, but PIP program copies file to User Area specified in # position	**PIP B:FILE.EXT[G#]= A:FILE.EXT**
Copy all files from Drive A to Drive B.	**PIP B:=A:*.***
Copy per command line entered, and verify transfer of data (slows operation somewhat but ensures accuracy).	**PIP <command>[V]**
Enter multiple PIP command lines: (Do not enter **[SPACE]** before **[RETURN]**). Enter any valid PIP command line (without having to repeat the letters PIP) when * prompt displays.	**PIP[RETURN]**
End multiple PIP command line session.	**[RETURN]**

To Do This...	Use This Program...
Copy files from hard disk to floppy diskette(s); continues one file across multiple diskettes.	**STORE**
Copy files placed by STORE program on floppy diskette back to hard disk.	**RESTORE**
Make "mirror image" of a diskette; does not require formatted diskette.	**UTILITY: DiskCopy Function**

File Management

To Do This...	Enter This...
View directory of all files on currently-logged drive.	**DIR**
View directory of all files on Drive d:.	**DIR d:**
"Move" to (read from, write to) another drive. Substitute desired drive letter for **d:**.	**d:**
"Move" to (read from, write to) another user area on current drive. Substitute 0 through 15 for #.	**USER #**

To Do This...	Enter This...
View directory for FILE.EXT on currently-logged disk; verifies that FILE.EXT exists on that disk.	DIR FILE.EXT
Rename a file; old file name no longer accesses file.	**REN new name.ext=old name.ext** Both new and old file names must be unambiguous (no wild cards).

Erase Files and Disks

To Do This...	Enter This...
Erase file FILE.EXT from disk d:.	ERA d:FILE.EXT

File Checking and Printing

To Do This...	Enter This...
Display file on screen (must be unambiguous file name).	TYPE FILE.EXT

Check And Change Status of Devices, Drives, and Files

To Do This...	Enter This...
Display status (R/W=read/write, R/O=read-only) and remaining storage space on currently-logged disk.	STAT [RETURN]
Display remaining storage on Drive d:.	STAT d:[RETURN]
List records, bytes, and extents used by file(s) named in command. Can be entered for another disk drive (STAT d:<afn>).	STAT <afn>[RETURN]
Change the attributes of file(s) to one of the following access <Acc> types: **R/O:** Read-only status; document cannot be changed or deleted until reset by STAT command. **R/W:** Read/write status; document can be changed or deleted. **SYS:** Changes document to system type; does not appear on screen with DIR command (does with DIRS command). **DIR:** Changes document back to non-system type (appears in directory).	STAT <afn><Acc>

SECTION VII

CP/M-86 AND THE DISK PARTITION SYSTEM

In Sections V and VI, you had an introduction to the CP/M-86 operating system. Here is a closer look at CP/M-86, an introduction to the Disk Partition System, and a brief description of how your Vector 4 system works.

A Closer Look at CP/M-86

CP/M-86 is divided into four functional parts:

o The Console Command Processor (or CCP)

o The Transient Program Area (or TPA)

o The Basic Input/Output System (or BIOS)

o The Basic Disk Operating System (or BDOS)

When you sit down to type at the console of your computer, you deal with the operating system at the command level. You have at your disposal all the CP/M-86 commands and can type them in to make the computer do what you want.

These commands include screen handling commands, to control input and output at the screen, and program commands, to call the various system and application programs. Most screen handling commands use the **[CTRL]** key to send them to processing. All program commands consist of letters and numbers, and **[RETURN]** sends them to processing.

The letters and numbers you enter at the keyboard are read and "translated" into commands by the Console Command Processor (**CCP**). Some commands are a built-in part of the Console Command Processor and so are termed "built-in" commands. Other longer commands run longer programs that must be stored on the disk or diskette when not in use. These are called "transient commands," because they only "visit" the operating system when needed. If the CCP is told to run one of these, it "loads" (copies) the program into the Transient Program Area from the disk or diskette where the program is stored.

The Transient Program Area is a section of the internal memory (in the RAM chips) used for holding programs while they run. It is the area of memory left over from when CP/M-86 itself is loaded into RAM. Except for the execution of the built-in commands (which are in the CP/M-86 area), all processing is done using instructions in the TPA.

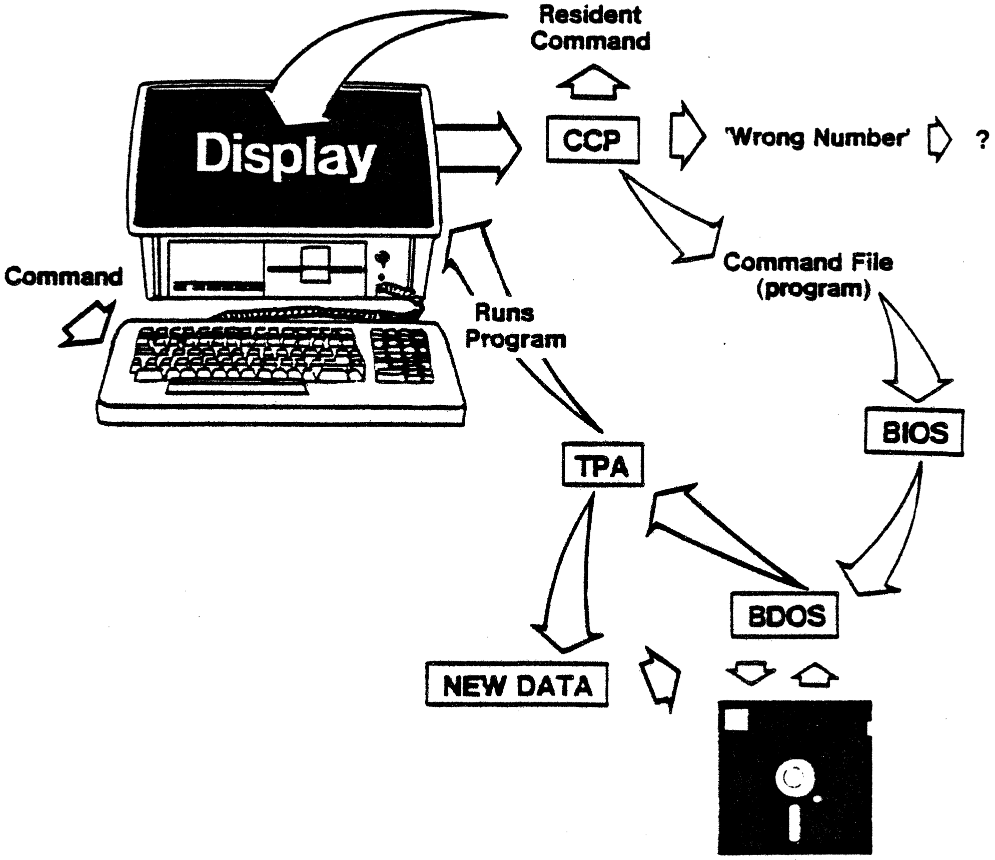


FIGURE 2 — Vector 4 CP/M-86'S LOGICAL STRUCTURE

When you enter a command, the Console Command Processor (CCP) checks your entry against the Built-in Commands. If your input matches one of these, the result of the command (for instance, a directory for the DIR command) is displayed on the screen. If you entered something else, the CCP looks at the directory on your drive. If it can't find what you entered, it prints your entry followed by a question mark. If it does find it, it turns over control to the Basic Disk Operating System (BDOS).

The program is located on the disk by the Basic Disk Operating System (BDOS) and sent to the Transient Program Area (TPA), through lines opened by the Basic Input/Output System (BIOS). The Transient Program Area stores the program while it runs. The results of the program are displayed on the screen, and any new data are recorded by the BDOS.

When processing of one program is complete, and you call another set of instructions, the old instructions are erased from the Transient Program Area. This is why it is called "transient": programs only "pass through" for processing.

But to load from disk into the TPA, CP/M-86 must first find out where to look on the disk, and then copy what it finds into the Transient Program Area. To do this, it uses the other two functions of the system. The Basic Input/Output System (BIOS) acts like the switchboard operator, opening lines between disk and TPA so that the program can be copied. Then the Basic Disk Operating System (BDOS) "reads" the directory of programs, finds the desired information on the disk, and sends a copy over the lines opened by the BIOS.

Review

The Console Command Processor interprets user commands and either works from its own instructions (in the case of built-in commands) or calls programs from mass memory (in the case of transient commands.)

The Transient Program Area stores and runs any programs called by user commands other than the built-in commands.

The Basic Disk Operating System keeps track of data as CP/M-86 sends it off to storage or pulls it from storage back into the RAM.

The Basic Input/Output System controls the circuits to be used for all transfer, turning the hardware on and off when required.

"Warm" and "Cold" Booting

As we saw earlier, booting is the way the system gets started: the main parts of CP/M-86 are copied from storage on the hard disk or System Diskette into the reserved section of RAM. There actually are two different kinds of boots: "warm boot" and "cold boot." A "cold boot" is performed after power has been turned off, when you press the RESET button, or during some programs which alter certain important settings of the system. In it, the entire operating system is again copied into the RAM.

Utility programs usually end either with a warm boot, to bring CP/M-86 up to date, or with a cold boot. These programs handle the booting automatically, as the last thing they do before returning control to the operating system. It is only for the first boot of the day, or when you have pressed "RESET" or turned off power, that you have to start the process yourself by pressing a selection key.

How CP/M-86 Manages Files

The command (program) files in your system are "ready-made": they're on the System Diskette you receive, and though you will often use them you never have to go "inside" them. In contrast, the data files you make up are your own creation: you give them names and make them as large or small as needed. What you put into a data file is up to you, as is how you manage it using CP/M-86 commands: copying, renaming, even erasing it. But there's also more to file management that CP/M-86 takes care of, invisibly, for you.

The most common job in your management of files is storage, and the first step is selection of a disk or diskette drive. After booting, the screen shows the **A>** prompt. This indicates that your Vector 4 is currently addressing Drive A, one drive you might use to store your files.

You could move to another drive to store files by entering the letter of that drive, a colon, and **[RETURN]**. To allow you to expand your storage, CP/M-86 is capable of reading from as many as 16 drives, given the logical names A through P. The number of your logical drives depends on your system. Vector 4s that are diskette-based have three logical drives, A, B, and C. Vector 4s with a hard disk usually are set to use four drives on the hard disk (A, B, C, and D) and two on the diskette drive (E and F). (See Section IX, "The Format Function.")

When you enter information to the logged-in drive, CP/M-86 keeps track of it using "records" of 128 bytes (one byte is roughly the same as one character.) When you want to copy this data to the disk, the computer sends each record off to the Basic Disk Operating System (BDOS), which "writes" it onto the logged-in disk or diskette. But a single record is too small as unit. It would be impractical for the BDOS to have to work with chunks of 128 bytes when it had to locate and transfer large quantities of data. So, to make file management faster, CP/M-86 uses several larger divisions.

CP/M-86 counts every 128 records and puts them in an larger gathering called an "extent." Every file takes up at least one extent, even if it actually contains fewer than 128 records. CP/M-86 also divides the surface of the disk into "allocation units." These are the actual storage locations on the recording surface. An allocation unit ranges in size from 2,000 to 8,000 bytes, depending on the disk. Since most files are much larger than this, their records usually are stored in many separate locations on the disk.

Make Room for Data

Organizing data to be stored is one part of file management. But if a program must change information already stored in file (say, the way an accounting program has to update accounts receivable), the Basic Disk Operating System has another job to do as well: after it finds the file to be updated, it must add, change, or remove information as the program directs. The BDOS can't "squeeze" new data into the existing files; it must store information where it has space on the disk surface, then remember where each segment of a file is "continued" from the previous part.

If a file needs more space for new records, a full new allocation unit is assigned, even if only one record is filled. Every extent can have up to eight allocation units assigned to it. All the information about the file's location (its "address": disk drive, file name, extent, and allocation unit) is stored in a File Control Block (FCB). CP/M-86 keeps a master list in memory of all allocation units currently assigned to each file, along with all unused units remaining on the disk.

As new records are entered or created, CP/M-86 updates its master list and assigns new allocation units where it has room on the disk. A copy of the File Control Block for the active file is stored in the Transient Program Area, and this also is updated to include the address of the new allocation units. CP/M-86 keeps track of all the variously stored parts of the file so that you only have to refer to it by name to get it for use.

If a file is erased or made smaller, the master list (and the file's FCB) is corrected to show the newly freed disk space, where CP/M-86 can write other information. This technique is called "dynamic file management."

How You See This

As the final step in its storage technique, CP/M-86 forms a directory for the disk or diskette. At the beginning of the entire data storage area, CP/M-86 writes an entry for each extent of each file. When you enter a "DIR" command, CP/M-86 displays the first entry for each file. (The other entries for the file "point" to the other segments, telling CP/M-86 where to find them.) The directory you see is made up from this general directory on the disk or diskette.

When you first log into the disk or a diskette, CP/M-86 copies up to 128 FCBs from the directory into memory. This way the operating system already "knows" where to look on the disk for any file you request. For a floppy diskette, this means that the entire directory is copied (a double-sided floppy diskette can hold 128 entries; a single-sided holds 64). For the hard disk, you may get only a portion of the files actually on record.

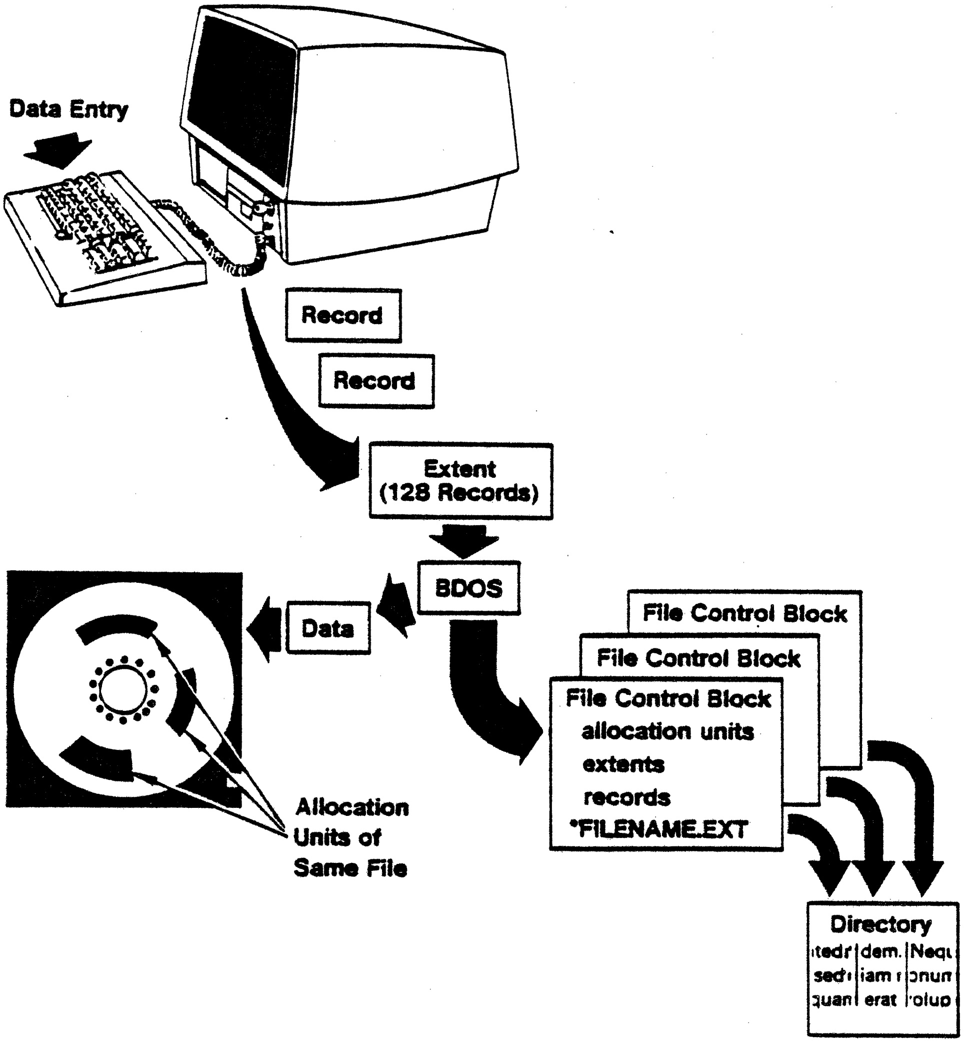


FIGURE 3 — Vector 4 CP/M-86'S FILE STORAGE AND ACCESS METHODS

As you enter data at the terminal, CP/M-86 counts each 128 bytes as a record. It counts 128 records entered to the same file as an extent. When you copy a file to disk, the Basic Disk Operating System sends the file to an allocation unit, which is a "geographic" section of the disk surface. This way, the BDOS doesn't have to move information already on the disk.

The BDOS keeps track of allocation units in a File Control Block, or FCB. When you request a file for use, the BDOS looks up the FCB for the file, then copies the file into the Transient Program Area.

When you type "DIR" for a directory, the Console Command Processor reads all the File Control Blocks for the disk, then displays the file names on screen. Up to 128 FCBs can be held in memory for each disk in use, making file access even faster.

Review

- A file is represented by an entry displayed on screen in the disk directory (by the command **DIR [RETURN]**).
- Each file is made up of at least one extent, which can contain up to 128 records of 128 bytes each.
- Each extent may take up no more than 8 allocation units of the disk surface.
- The locations of allocation unit(s), extent(s), and the file name for each file is stored in a File Control Block (FCB).
- All file management at this level is handled by the operating system. You only have to use the name of the file to have access to it.

How to Use Your Disk Drives

Your Vector 4 comes with one or two physical drives, and these may be arranged in one of several ways. You may have one diskette drive, or one hard drive and one diskette drive, or two diskette drives. As we saw earlier, these drives can be set up to run as different logical drives. This use of different logical drives has benefits both for you and for the computer.

For the operating system, use of multiple logical drives provides a further way of dividing storage space. If the hard disk is "split" into two or more drives, the BDOS can ignore all but one section of mass memory right away when searching for a requested file.

When you enter a command or ask for a file, the computer always assumes that you mean for it to look on the current drive, unless you specify a different drive name. This assumption is called a "default," and this is a term that will come up many times in your work with computers. Since the computer cannot really think for itself, its instructions must include automatic "decisions" for it to use when no specific instructions are given--for instance, when it has to decide what drive to use before you have specified one. Defaults keep the system running smoothly and relieve you of always having to give explicit instructions.

Using Your Drive Configurations

If your system has two diskette drives, you can use the double slots to run a program diskette in Drive A and a data diskette in Drive B. This allows you to run long programs that, if loaded in completely, would take more RAM space than the computer actually has. In this case, some of the program stays on the diskette while other parts are taken into the internal memory when needed. Your Memorite word-processing package, for example, works this way.

Both the Drive A and Drive B slots are set to use double-sided diskettes. A third logical drive, Drive C, also works through the left-hand slot (the "Drive B" slot) and is set for use of single-sided diskettes. This allows you to insert program diskettes of either format in that slot, then select the appropriate logical drive (Drive B or Drive C) to handle the job.

If your system has a hard disk, you can assign other logical names to the diskette drive. For example, after dividing the hard disk into Drives A, B, C, and D, you could set up a single-sided diskette drive named Drive E, and a double-sided drive named Drive F. Drive selections are made through the Format function of the UTILITY Program, as part of the overall process of formatting the hard disk. (See Section IX, "The UTILITY Program.")

Once logical diskette drives have been specified, you can use them to format all new blank diskettes, again using the Format function. Format detects which logical drive you have logged in, and the diskette is formatted accordingly. To format a single-sided diskette, then, you would use your logical single-sided drive. To format a double-sided diskette, you would use your double-sided drive.

Further Memory Divisions

An additional (but optional) division of memory is the user area. A user area is a section of a logical drive area. You can take advantage of up to sixteen different user areas. The default is User Area 0; the highest number is User Area 15. For example, files on Drive A could be placed in User Area 0, User Area 1, User Area 2, and so on.

You don't need to set up user areas as you do logical drives. They're already built in to the system "landscape." You have only to select one, through the USER command.

EXAMPLE: To select User Area 11 of Drive A:

```
A>USER 11  [RETURN]
```

When working inside a user area, think of it as if it were a separate disk. That is, entering **DIR** while working in User Area 5 of Drive B displays the names of files stored in User Area 5 but not of files in any other user area. User areas give you a further way to divide storage space and are especially helpful in assigning separate "work areas" among several people or tasks.

Neither drives nor user areas are actual areas on the disk or diskette. The operating system simply "thinks" about it differently in file management. There is no maximum size for a user area except the maximum size of the hard disk or floppy diskette itself.

Since organizing files by user area can take extra planning, if you have a floppy-only system it may be more work than it's worth. It's easier just to remove one floppy and insert another in its place, keeping different materials on different diskettes. But for a hard-disk system, the user area division can make your file system more efficient. Think of the user areas as different drawers in the same filing cabinet.

The Disk Partition System

As we saw above, logical drives and user areas offer ways of dividing mass memory—the hard disk and diskettes--into segments. CP/M-86 thinks about logical drives as if they were separate disks. It thinks about user areas as if they were separate sections of each disk. All of these segments are controlled by one operating system, CP/M-86.

Vector Graphic's Disk Partition System takes this approach one step farther. It makes it possible for you to divide your hard-disk into separate areas for completely different operating systems. In addition to CP/M-86, you can put up to nine other operating systems on the disk and have them available for use. Each operating system is listed in the "Booting Options" menu that appears when you start up the system, and any can be selected for use, either for the day or the until the next time you do a "cold boot."

Normally, one computer's hardware can't handle more than one operating system at once. This is partly because of different methods of I/O-- in how operating systems move information to and from the printer, console, and disk drives— and partly because each operating system has its own way of making up files. Recall CP/M-86's way of managing files, discussed above. Another operating system, MS-DOS 2.0, takes a very different approach. It stores information in units of different lengths and builds a different kind of "file control block." The hard disk and diskettes, when formatted, receive different "track and sector" markings, too.

But it's desirable to have several operating systems available: it gives you access to more application software. The Disk Partition System makes it possible by solving both of the basic problems. First, it provides a special program that handles input and output for the disk drives, printer, and console. Each operating system uses this program instead of its own instructions to control these devices. Second, it partitions (divides) the hard disk into separate areas for each operating system and its files. Each partition can be formatted in the right way and contains only files that belong to the resident operating system.

Each partition is an actual physical division. Within each operating system's area, you can again have separate logical drives and user areas.

In other words, the Disk Partition System takes the hard disk and turns it into as many "separate" hard disks as you have operating systems. Because the partitions are really physical, it's like having several disks combined into one.

Is the Disk Partition System Needed for CP/M-86?

If you plan to use only one operating system, CP/M-86, do you still need to use the Disk Partition System? Yes. The Disk Partition System provides several programs used by CP/M-86: System Generation (SYSGEN), Format, and Diskcopy. These are all functions of the UTILITY program (see Section IX). The Disk Partition System also supplies the CONFIG (Configuration) program (Section IX), which allows you to adjust your system to different kinds of hardware and different modes of use.

So using your Vector 4 even with only one operating system still means using the Disk Partition System.

What About Vector 4 Systems with No Hard Disk?

If your Vector 4 is a floppy-only system (no hard disk), you will not partition a hard disk, of course. But you still will use the UTILITY and CONFIG programs for many system jobs. Each of the commands in these programs includes options for floppy-only systems.

The CP/M Simulator

The Disk Partition System also includes a special translator program, the CP/M Simulator. The Simulator allows command and application programs written for Extended CP/M and Vector 4 CP/M to be run under the CP/M-86 operating system. Any software packages you have acquired for Vector 4 or Vector 3 systems can be run through CP/M-86 and the CP/M Simulator.

The CP/M Simulator is a command file named named RUN8.CMD. The system calls on the file automatically any time if finds a request for execution of a .COM program—these are any of the command files or programs written for the eight-bit CP/Ms, such as Extended CP/M and Vector 4 CP/M. The Simulator then takes over execution of the program, performing any special translations of operating system tasks that might be needed. When the program ends, the Simulator returns you to CP/M-86.

Since the operation of the CP/M Simulator is "transparent," you can run .COM files under CP/M-86 just as you would under Vector 4 CP/M. Enter the program name as usual.

EXAMPLE: To run the Memorite word-processing program with the Simulator:

```
A>MEMORITE [RETURN]
```

Is the Disk Partition System Ready For Use Now?

If your Vector 4 is new and has a hard-disk drive, the Disk Partition System is not installed on the hard disk when your system arrives.

To install CP/M-86 and the Disk Partition System, see "Installing CP/M-86 on Your Hard Disk," Section XI.

If your Vector 4 is a diskette-only system, you can use either Vector 4 CP/M or CP/M-86 by booting one or the other from its System Diskette. Then use the programs of the selected operating system for functions such as SYSGEN, DISKCOPY, and FORMAT.

Your Vector 4 System

Hardware and Software

Your Vector system is designed to process and store information. Two different parts of the system, hardware and software, work together to do this. "Hardware" means the mechanical and electrical parts of the system, such as the console, the disk drives, and, most important, the electronic circuits inside the cabinet. "Software" means programs of instructions for the system. These programs use the hardware to do their work and also control what the hardware does.

In your Vector system, hardware components do three things:

* memory,

* processing,

* input/output.

Memory

Your Vector system uses three different kinds of memory. They are:

* volatile memory,

* non-volatile memory,

* mass memory.

Each type has certain advantages that suit it for a specific use.

Volatile memory is called Random Access Memory (RAM). It holds information for jobs the computer is doing at the moment. This information can be program instructions or data for a program to use, and it can be "read from" or "written into" RAM very rapidly. This is the main advantage of RAM, that it is fast and helps the computer to process information quickly. For this reason, although all of your programs and data are stored in mass memory, they are moved into RAM before any processing is done.

The main disadvantage of RAM is that it must have electric current running through it to work. It only "remembers" while the computer is turned on, thus the name "volatile."

RAM is built into a number of silicon chips mounted on a printed circuit board inside the computer.

Non-volatile memory is called Read Only Memory (ROM). It too is located in chips (though of a different kind) inside the computer, and, as with RAM, its contents are available for use almost immediately. As a further advantage, ROM can hold data even when the computer is turned off, so it is used to hold information that is needed repeatedly in normal operation, such as the instructions that start up the system, or coded forms of the letters and numbers that appear on the screen. The contents of ROM memory are created by the manufacturer and can be changed only by removing and replacing the chips themselves.

Mass memory for your Vector system means either floppy diskettes or a hard disk. Floppy diskettes are 5-1/4" circles of Mylar and look something like a 45-RPM phonograph record sealed in a square, plastic, protective envelope (there are also larger or smaller diskettes). The hard disk is a platter made of a metallic alloy and is sealed inside the console cabinet in a protective box.

Both kinds of mass memory hold large amounts of information and hold it whether the power is on or off. Each has its own advantages, too. A hard disk makes storage and retrieval very fast and holds from ten to fifty times more data than can a floppy diskette. But it is permanently attached to the system, so the information it holds can be moved to a different system only with some difficulty. Floppy diskettes are slower and hold less data, but they are portable. You can insert a new diskette in about the time it takes to change a tape cassette. This means you can quickly place new information in your system or carry away information for use elsewhere.

Processing

Processing--the actual computing of the system--is done by the Central Processing Unit (CPU). In your Vector system it is a third kind of chip called a microprocessor. The Vector 4 actually has two microprocessors, named the Zilog Z80B and Intel 8088. Only one is in use at one time, as required by the kinds of programs being run. The microprocessor works closely with the RAM, because all of the "raw material" for processing comes from temporary storage in RAM, and the results of processing--the "finished product"—are returned there afterwards. The microprocessor chip and the RAM chips all are mounted on the same printed circuit board to aid this cooperation.

Input and Output

The "raw materials" and "finished goods" of processing come from and go to RAM. But so long as they remain there, they're of little use because we can't get to them. We need the system's third element, Input and Output (I/O), to move this information to and from the computer's interior. At first, inputconsists of the letters and numbers that you tap at the. keyboard. Sometimes this input tells the system to go elsewhere--perhaps to one of the disks--to get other information; this is also a kind of input. Output usually displays on your screen, but, depending on your instructions, it also can be sent to mass memory (the disks) or to the printer.

How Hard is This?

Controlling the many electronic steps that make memory, processing, and input/output possible is complicated and requires thousands of specific instructions. To simplify operation of your system, computer professionals plan and write these instructions and provide them to you as programs: planned sets of instructions. You can then request any program of instructions simply by entering one- or two-word commands at the keyboard. The complete set of programs created for your system is called software.

Two Kinds of Software

All the software you will use falls into two groups. The first, application software, includes word-processing programs, accounting and inventory packages, and any other "packages" you buy to handle some business task. The second, system software, runs the computer itself and provides a framework inside which the the application software can function. All the programs of this second kind--the system software--make up the operating system.

Application software is said to "run under" a particular operating system. This means that certain features of the two--particularly the way files of information are put together--are compatible, and that the operating system can correctly control the application program as it runs.

What the Operating System Does

To control what goes on inside the computer, an operating system does the jobs of translator, switchboard operator, and bookkeeper. It translates what is entered at the keyboard into a form the computer can use (electronic pulses). It selects the circuits to send these signals to the microprocessor, which acts on the instructions. And it stores the results of processing in mass memory or finds the needed files already there to update them with the results of processing.

The operating system "knows" how to do all of this because of the instructionsit contains, and it does most of it automatically. But there are also things it can do beyond its automatic functions. These services are at your disposal: they are the system commands. A system command names and "calls" (brings into operation) one of the operating system's user programs. The operating system then does work for you, carrying out the instructions in the program you have selected.

How Does the System Get Started?

While it is turned on, the computer keeps at least a portion of the operating system in its RAM memory chips, where it is close to the CPU and ready to do its various tasks. But RAM will hold information only when it is supplied with electric current. So when the computer's power is off, the operating system is stored on the hard disk.

This puts the operating system in a safe place but also creates a new problem: how to get the operating system back into RAM when the computer is turned on. The problem is solved by assigning this job to a special set of instructions stored in a ROM chip inside the computer. (Remember that ROM memory holds its information even without electric current.) When power is turned on, these special instructions go to work immediately and start transferring the first parts of the operating system into the computer.

Because this is a bit like "pulling yourself up by your bootstraps"--the computer is loading itself with its own instructions—it is called "booting," and the special starter instructions in ROM are called the "bootstrap loader" (they "load" the operating system into RAM).

In your Vector system, the ROM chip also contains information about your system's "configuration"—that is, about the kind and number of its disk drives. You may have two floppy-diskette units or a floppy drive and a hard-disk drive. The booting process is slightly different for each configuration. In addition, during booting this information about your system's drives is passed along to the operating system, so that it already knows what it has to work with when it is completely loaded. This saves you from having to supply this information to the operating system yourself.

SECTION VIII

BUILT-IN COMMANDS

What is a Built-in Command?

Built-in commands do jobs that are frequently used and fairly quick, such as changing a file name or erasing a file. Built-in commands are in the part of CP/M-86 that is brought into the Vector 4's internal memory during booting. Because they then are in RAM, rather than on the disk, built-in commands can be carried out immediately.

CCP Control Keys

Control key commands are the "little brothers" of built-in commands. The most important of the control-key commands are the screen-handling commands. Because they are used to edit your entry of all the other commands, these and the other control key commands are listed first below.

Control key commands are entered by pressing the **[CTRL-]** key (at the lower left corner of your keyboard) while entering a letter key (for example, **[CTRL-X]**). The list below explains each command.

[BACKSPACE] or **[DEL]** - deletes the last character entered at the keyboard. Both keys do the same thing.

[CTRL-X] - deletes an entire line entered at the keyboard.

[CTRL-U] - "cancels" the current line by not transmitting it to CP/M-86. **[CTRL-U]** does not delete your entry; it simply moves the cursor to the next line so that you can start over. The first line is ignored. After pressing **[CTRL-U]**, you will get the cursor but not the prompt (A> or B>). You don't need the prompt to enter your command again, but if you want it, press **[RETURN]**.

[CTRL-E] - marks "physical end-of-line." Splits a long command line for readability. The cursor moves to the beginning of the next line (as in a carriage return on a typewriter), but command line is not sent until **[RETURN]** is pressed.

[CTRL-C] - [CTRL-C] does two things: it restarts the system via a warm boot (effectively cancelling the program in progress); and it does a warm boot for a newly inserted diskette.

[CTRL-D] - clears the screen before any CCP command is executed. It must be entered before the command. You can also enter [CTRL-D] **[RETURN]** to clear the screen.

[CTRL-Z] - indicates the end of input from the terminal. Used only when entering a file from the keyboard (for example, when using the PIP program with the keyboard as the source of your input). This command is a carry-over from the days when the keyboard was a Teletype-style terminal and its input was punched directly onto tape.

[CTRL-P] - copies all subsequent console output to the currently assigned printer. To cancel **[CTRL- P]**, enter a second **[CTRL- P]**.

[CTRL-S] - temporarily stops a display moving by on the screen. Resume the display by pressing any key except **[CTRL-C]**. **[CTRL-S]** is used to stop output on high-speed terminals (such as the console of your Vector 4) to view a segment of the output of a program.

Brief List of Built-in Commands

Here is a summary of the built-in commands. Each command is explained in greater detail in the following pages. Commands are shown in UPPER CASE, but you can use either upper- or lower-case letters to enter the commands at the keyboard.

ERA	Erases a file or files
DIR	Lists the names of files in a directory
DIRS	Lists the names of .SYS (System) files in a directory
REN	Renames a file
USER n	Allows you to choose different working areas on a disk
TYPE	Types (displays) the contents of a file on the console screen

The ERA Command

The ERA (ERASE) command erases a file or several files from the disk. You can use either ambiguous or unambiguous file names. (An ambiguous file name uses wild card symbols, * or ?, to select several files; an unambiguous file name refers to only one file.) The following examples show some different uses of ERA.

ERAsing With Unambiguous File Names

1. To erase a file named SALES.MEM:

FORMAT:

```
A>ERA <UFN> [RETURN]
```

EXAMPLE:

```
A>ERA SALES.MEM [RETURN]
```

RESULT:

The file SALES.MEM is erased.

ERAsing With Ambiguous File Names

2. To erase several files with the same file name (e.g., SALES) but a different extension:

FORMAT:

```
A>ERA <filename>.* [RETURN]
```

EXAMPLE:

```
A>ERA SALES.* [RETURN]
```

RESULT:

All files with the name SALES (no matter what their extension) are erased.

3. To erase all files with different names but the same extension (e.g., MEM):

FORMAT:

```
A>ERA *.<EXT> [RETURN]
```

EXAMPLE:

```
A>ERA *.MEM [RETURN]
```

RESULT:

All files with the extension MEM are erased.

4. To ERAse several files with names or extensions that are slightly different but share common letters (for example, they all begin with the letter P).

   Suppose the files are named PAT, PUT, POT, PET. The extensions are MEM, MOM, MAM, MUM. Use wildcard symbols:

   FORMAT:

```
A>ERA X?Z.C?M [RETURN]
```

   EXAMPLE:

```
A>ERA P?T.M?M  [RETURN]
```

   RESULT:

   All the files listed in our example are erased, because their filenames fit the wildcard pattern.

5. To ERAse many different files on a disk, no matter what their names or extensions:

   FORMAT:

```
A>ERA *.* [RETURN]
```

   EXAMPLE:

```
A>ERA *.* [RETURN]
```

   RESULT:

Before erasing any files, the system asks you to confirm your command:

```
ALL FILES (Y/N)?
```

Since you do want all the files erased, enter **[Y]** and **[RETURN]**.

6. To erase all files on drive C with a MEM extension.

FORMAT:

```
A>ERA d:*.EXT [RETURN]          (Note: "d" stands for the drive
                          letter.)
```

EXAMPLE:

```
A>ERA C:*.MEM [RETURN]
```

RESULT:

All files on Drive C with the extension MEM are erased. No files are erased from other drives.

The DIR Command

The DIR (DIRectory) command displays the contents of the disk directory. The directory holds the names of all files stored on the disk. You can use DIR to see the name one file, several files, or all files.

1. To list the directory entry of one file—for example, named RECORD.MEM:

FORMAT:

```
A>DIR <UFN> [RETURN]
```

EXAMPLE:

```
A>DIR RECORD.MEM [RETURN]
```

RESULT:

The filename RECORD.MEM is displayed with the indicator A:to show that the file is on the current drive.

2. To list the directory entries of several files whose names are slightly different but share common letters—for example:

   Filenames: PUT, POT, PET, PAT.

   Extensions: MEM, MOM, MAM, MUM.

   FORMAT:

   ```
   A>DIR X?Z.C?M [RETURN]
   ```

   EXAMPLE:

   ```
   A>DIR P?T.M?M [RETURN]
   ```

   RESULT:

   All the sample files are listed because their filenames and extensions fit the wildcard pattern.

3. To list all files with the same extension—for example, MEM:

   FORMAT:

   ```
   A>DIR *.<EXT> [RETURN]
   ```

   EXAMPLE:

   ```
   A>DIR *.MEM [RETURN]
   ```

   RESULT:

   The names of all files with the extension MEM are displayed.

4. The DIR command frequently is used to see what files are stored on another drive. For example, to obtain a listing of the filenames in drive B while logged into drive A:

   FORMAT:

   ```
   A>DIR d: [RETURN]
   ```

EXAMPLE:

```
A>DIR B: [RETURN]
```

RESULT:

The directory for Drive B is displayed. You also could log onto the B drive, then enter **DIR [RETURN]**, but this is not as quickly done.

5. To list a particular file on Drive B—for example, named TECH.COM:

FORMAT:

```
A>DIR B:<UFN> [RETURN]
```

EXAMPLE:

```
A>DIR B:TECH.COM [RETURN]
```

RESULT:

The filename TECH.COM is displayed from the Drive B directory.

6. To list, from a different drive, the directory entries of several files whose names are slightly different but share common letters. Use wildcard symbols. For example, to list the file names used in previous examples:

FORMAT:

```
A>DIR B:*.A?M [RETURN]
```

EXAMPLE:

```
A>DIR B:P*T.C?M [RETURN]
```

RESULT:

The files PUT.COM, PET.COM, PAT.COM, and PAT.COM are shown, because their names fit the wildcard pattern.

The DIRS Command

The DIRS command works the same as DIR, but displays only the names of files that have the "System" attribute: files that form part of the operating system itself. In addition, DIRS shows only those System files that are in your current user area. You can always access System files (call them for use) from any user area, but DIRS lists them only when you are logged into the user area where the files actually reside.

FORMAT:

```
A>DIRS <filespec> [RETURN]
```

EXAMPLE: You are logged into user area 0:

```
A>DIRS [RETURN]
```

RESULT:

A list of all System files in user area 0 is brought to the screen.

If there are other, non-system files also in the user area, this message also is displayed:

```
NON-SYSTEM FILE(S) EXIST
```

Like DIR, DIRS can be used with wildcard symbols.

EXAMPLE:

```
A>DIRS *.CMD [RETURN]
```

RESULT:

For your user area on Drive A, displays all System files having a filetype **.CMD.**

The REN Command

The REN (REName) changes the names of files on your disk. The command format is:

```
A>REN <new>=<old>
```

where **<new>** is the new name for the file, and **<old>** is the old (current) name.

1. To change the name of a file called PRIMER.MEM to RECORD.TXT:

   FORMAT:

   ```
   A>REN NEWFILE.EXT=OLDFILE.EXT [RETURN]
   ```

   EXAMPLE:

   ```
   A>REN RECORD.TXT=PRIMER.MEM [RETURN]
   ```

   RESULT:

   The file is now called RECORD.TXT.

2. REN also can be used to change a file's extension without changing its name.

   FORMAT:

   ```
   A>REN Y.TXT=Y.MEM [RETURN]
   ```

   EXAMPLE:

   ```
   A>REN DAISY.TXT=DAISY.MEM [RETURN]
   ```

   RESULT:

   The old file DAISY.MEM now is called DAISY.TXT.

3. REN can be used to change a file's name but not the extension:

   FORMAT:

   ```
   A>REN X.ASM=Y.ASM [RETURN]
   ```

   EXAMPLE:

   ```
   A>REN RECORD.ASM=TECH.ASM [RETURN]
   ```

   RESULT:

   The file TECH.ASM is now called RECORD.ASM.

4. You also can rename a file on a drive other than the currently-logged disk:

   FORMAT:

   ```
   A>REN B:X.BAS=Z.BAS [RETURN]
   ```

   EXAMPLE:

   ```
   A>REN B: INITIAL.BAS=PROMPT.BAS [RETURN]
   ```

   RESULT:

   On Drive B, the filename PROMPT.BAS has been changed to INITIAL.BAS.

The USER n Command

USER allows you to divide the space on a disk into several user areas. This command is useful if you share your disk with others, or if you need to have separate compartments within your disk to store files. User areas are numbered 0 to 15 (these numbers take the place of "n" in the command). You can think of user areas as different drawers in the same filing cabinet. At the start, CP/M-86 assumes that you are in user area 0. You must give the USER command to move to another area.

FORMAT:

```
A>USER n [RETURN]
```

EXAMPLE:

```
A>USER 3 [RETURN]
```

RESULT:

CP/M-86 takes you to User Area 3.

USER can also be used to check which user area you are in at the moment.

```
A>USER [RETURN]
```

The system displays your user area number in response.

The TYPE Command

The TYPE command displays the contents of a file on the screen (it "types" it to the console).

FORMAT:

```
A>TYPE X.Y [RETURN]
```

1. EXAMPLE: To see the contents of a file named SAVE.MEM:

```
A>TYPE SAVE.MEM [RETURN]
```

RESULT:

The contents of the file SAVE.MEM are displayed on the screen.

2. You can TYPE files stored on other drives.

FORMAT:

```
A>TYPE d:filename.ext [RETURN]
```

EXAMPLE:

```
A>TYPE B:PROTOCOL.PRN [RETURN]
```

RESULT:

The file PROTOCOL.PRN on Drive B is be displayed on the screen.

SECTION IX

UTILITY AND CONFIG

UTILITY and CONFIG are programs Vector Graphic has added to the regular CP/M-86 commands to handle several important system jobs. UTILITY includes commands for copying the operating system, copying diskettes, and formatting a hard disk or diskettes (to prepare them to receive data). CONFIG includes a variety of commands useful in adapting your Vector 4 to run different application programs and peripheral devices.

The UTILITY Program

UTILITY does three system tasks:

1- It generates (copies) an operating system for use on a disk;

2- It copies the contents of one diskette to another;

3- It formats, or prepares for use, the hard disk and diskettes.

Any operating system needs these functions, but usually each has its own version. Because the Disk Partition System lets you install several operating systems, you need SYSGEN, DISKCOPY, and FORMAT commands that can work for any of them. The UTILITY does this for you.

Use UTILITY for these functions instead of versions of the commands provided by CP/M-86, by MS-DOS, or by any other operating system offered by Vector Graphic for your Vector 4.

To call the UTILITY program, type

```
A>UTILITY    [RETURN]
```

The screen displays:

```
UTILITY PROGRAM 1.XX (VV)

Available Functions:

          [0] Exit program
          [1] System generation
          [2] Floppy Diskcopy
          [3] Format

Make selection from above:
```

You select a function by typing its number, then [RETURN].

What the Functions Do

System Generation

SYSGEN allows the operating system to make a copy of itself. You may need this function to transfer the operating system to a backup diskette for safekeeping, or to move it from a diskette to the hard disk for regular use in booting and running the system. Both of these jobs are done by the System Generation function.

Floppy Diskcopy

The Floppy DISKCOPY function is used to make a quick, complete copy of one diskette to another, either to keep as a backup or for transferring files to another system.

Format

FORMAT puts a "framework" for the operating system onto a blank disk or diskette. CP/M-86 uses this framework to read (find) and write (store) data on the disk and to keep track of where data has been stored. For the hard disk, FORMAT also is the way the disk is partitioned (divided) into separate operating system areas. Neither the hard disk nor a diskette is usable until it has been formatted.

The FORMAT Function

FORMAT can be used in two ways: complete formatting, and re-formatting.

Complete formatting can be done both to a hard disk and to diskettes. It erases all existing data on the disk surface and writes the format markers. On the hard disk, this includes information about the operating system or systems to be installed.

Re-formatting is done only to the hard disk. It lets you format one logical drive area without affecting data in other areas on the disk.

A disk or diskette is not usable until it has been formatted. The operating system needs the framework to be in place before any data arrives so that it can orient itself in the disk and keep track of where new data has been stored.

You must format brand-new blank diskettes. Diskettes holding application software, on the other hand, should never be formatted, because FORMAT erases the entire disk surface before writing the orientation markers. Never format a disk or diskette containing valuable information. Always check a used diskette's directory before formatting it, and look at any file whose contents you can't readily identify. It's less expensive to buy a new diskette than it is to recreate lost work, and once FORMAT has done its work, the old files are lost.

Complete FORMAT of Hard Disk

This use of FORMAT "cleans and marks" the hard disk, names the operating systems to be used, and assigns logical drives for both the hard disk and diskette drives.

Complete formatting of the hard disk can be done only when the system is booted from a floppy diskette. (The hard disk can't format itself, because it would have to erase its own instructions.) Complete formatting can be done whether the disk has been formatted before or has never been formatted. It is different from the re-formatting option of FORMAT in that it affects the entire disk, whereas re-formatting works only on selected drive areas.

Formatting takes time, because the program must write the format markers to small, individual sectors on the disk one by one. These sectors number in the thousands. Average times for FORMAT are as follows:

10 Megabyte hard disk (Vector 4/40s)	=	40 minutes
5 Megabyte hard disk (Vector 4/30s)	=	25 minutes
Floppy diskette	=	7 minutes

The example below is given for a Vector 4/4Ø system, which has a 1Ø-Mbyte hard-disk drive and one floppy drive, and for installing one operating system, CP/M-86.

1. From the UTILITY menu, type [3] to select the FORMAT function.

If the hard disk has never been formatted, this menu is displayed:

```
FORMAT

Available functions:

        [0] Exit function
        [1] FORMAT floppy
        [2] Winchester format

NOTE: THE FLOPPY FORMATTING OPERATION WILL BEGIN
IMMEDIATELY UPON VALID SELECTION.  [ESC] CANCELS THE
FUNCTION AND EXITS.

Make selection from above:
```

If the hard disk has been formatted previously (you want to format the complete disk again), this menu appears:

```
FORMAT

Available functions:

        [0] Exit function
        [1] FORMAT floppy
        [2] Winchester format
        [3] Winchester reformat

NOTE: THE FLOPPY FORMATTING OPERATION WILL BEGIN
IMMEDIATELY UPON VALID SELECTION.  [ESC] CANCELS THE
FUNCTION AND EXITS.

Make selection from above:
```

2. Type [2] to select "Winchester format."

This menu appears:

```
FORMAT - WINCHESTER FORMAT

NOTE:  [ESC] CANCELS THE FUNCTION AND EXITS.

Enter system A type:
```

This asks you to define your first operating system. "System A" is the first of several operating systems that might be installed on the disk.

The type number of each operating system distributed by Vector Graphic is written on the label of the system diskette. Look at the label of the CP/M-86 System Diskette and find the system type number.

3. Enter the type number after the prompt (here "NN" is the number):

Enter system A type: **NN [RETURN]**

After a moment, this line appears again, asking you to define a second operating system, System B:

Enter system B type:

If you are installing a second operating system—for example, MS-DOS 2.0—look at the label on the system diskette and find the type number. Enter this figure for system B just as you did for system A.

The FORMAT function prompts you to define additional systems (up to system J, the tenth and last one possible) as long as you continue to enter type numbers in answer to each prompt. Only when you omit an entry and press [RETURN] does the program move on to the next menu line:

Remaining storage on Winchester: 9Mb
Enter file area allocation for system A:

Our example is for a ten-megabyte disk. CP/M-86 will take up some of this room, and FORMAT rounds off the remaining space to the nearest whole unit, so your remaining storage space is nine megabytes--about nine million characters. (Actually, you have almost all of the 1Ø Mbytes left; the rounding-off just doesn't show it. FORMAT will add the "extra" space to the last logical drive you create, as shown below.)

To select logical drives for System A, you will indicate how much of the remaining space should go to each drive. Suppose you decide to create two drives, A and B.

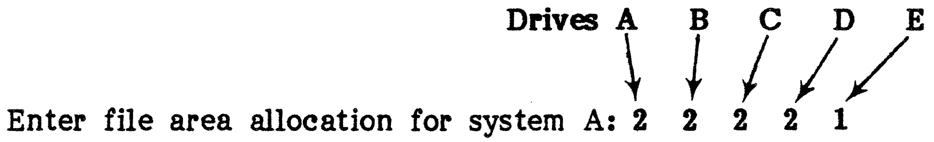
EXAMPLE: To divide the nine megabytes into two logical drives, A and B, one having five megabytes, the other four:

Drive A **Drive B**

Enter file area allocation for system A: **5 4**

(Logical drive B actually will contain almost five megabytes of space.)

EXAMPLE: To create five logical drives, A through E, of smaller units:

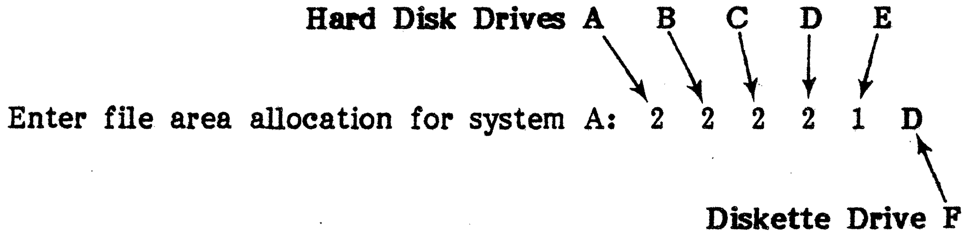


```
Enter file area allocation for system A: 2 2 2 2 1
```

Again, the last logical drive actually has almost two megabytes of storage.

Logical assignments for the diskette drive are made now, too. These are defined by typing [D] for Double-sided or [S] for Single-sided in the allocation sequence.

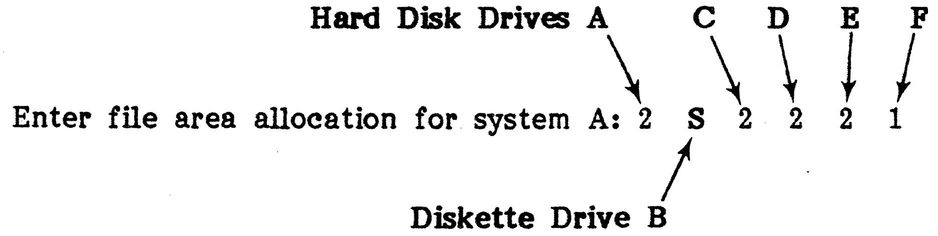
EXAMPLE: To define a double-sided drive as Drive F:



```
Enter file area allocation for system A: 2 2 2 2 1 D
```

You can put a diskette drive anywhere in the sequence except as Drive A. Drive A is reserved as the System Drive and must be assigned to the hard disk.

EXAMPLE: To define Drive B as a single-sided diskette drive:



```
Enter file area allocation for system A: 2 S 2 2 2 1
```

If you skip one of the two assignments for diskette drives ([D] for double-sided or [S] for single-sided), FORMAT automatically adds it to the end of your sequence. In the example above:

```
Enter file area allocation for system A:  2  S  2  2  2  1 D
```

FORMAT adds diskette
Logical Drive G
(double-sided)

If neither assignment is made, FORMAT makes both of them, assigning the double-sided drive first:

```
Enter file area allocation for system A: 2  2  2  2  1  D  S
```

FORMAT adds logical Drives F and G,
double-sided drive first

Usually it's convenient to put your diskette drives last, because it's easier to remember drive letters.

4. When your allocations are complete, tap [RETURN]. FORMAT then confirms your entries. For our example line above:

```
System A has 5 logical Winchester units totalling 9 Mbytes.
System A has floppy accessed single-sided as Drive B.
System A has floppy accessed double-sided as Drive G.
```

When you have allocated all storage, this message appears:

```
Enter [RETURN] to accept allocation, [ESC] to abort:
```

Press [RETURN] and the FORMAT operation begins at once. Press [ESC] to stop the function and return the system to CP/M-86.

After a formatting of the hard disk, the system tells you it will perform a RESET when you tap [RETURN]. This reset is needed to put into effect new values created during formatting. When the message appears, type [RETURN].

Error Messages in Formatting the Hard Disk

Two error messages may appear during your use of FORMAT for the hard disk. Both are indications there may be a defect in the physical surface of the disk. The first is:

```
ERROR IN DIRECTORY AREA -
    SYSTEM <system letter> PARTITION <drive #>
    REPARTITION DISK
```

The second is:

```
PHYSICAL ERROR <error code>
<ON WRITE / VERIFY>, SECTOR <sector #>
```

As FORMAT writes its markers to the disk, it tests the recording surface by writing and then reading a pattern of test data. The two error messages above signal that a possible flaw was detected.

The first message is the more important, because the directory area for each logical drive is CP/M-86's only guide to finding data on the disk, and errors in these areas cannot easily be repaired. The message tells you to repartition the disk. This will move the directory to another location on the disk surface, away from the source of the error.

EXAMPLE: You have partitioned the System A area of the hard disk as in the example above—

Logical drive A	=	2	Mbytes
Logical drive B	=	2	Mbytes
Logical drive C	=	2	Mbytes
Logical drive D	=	2	Mbytes
Logical drive E	=	1	Mbytes

—and this error message appears:

```
ERROR IN DIRECTORY AREA -
    SYSTEM A PARTITION 3
    REPARTITION DISK
```

You must change the size of your logical drives, or their order, to re-position the directory of Drive C (the third partition).

EXAMPLE: You decide to create fewer drives of larger size:

Logical drive A = 3 Mbytes
Logical drive B = 3 Mbytes
Logical drive C = 3 Mbytes

EXAMPLE: You decide to keep the same number and size of partitions but change their order:

Logical drive A = 2 Mbytes
Logical drive B = 1 Mbytes
Logical drive C = 2 Mbytes
Logical drive D = 2 Mbytes
Logical drive E = 2 Mbytes

In each case, by changing the size or order of the drives, you can cause the directory area to be relocated on the disk (the directory is the first data stored for each drive area).

The second error message also is caused by a possible physical flaw on the disk surface, but in this case the error is in a data area rather than in the directory area of a logical drive. The message tells you that the error appeared either "On Write," when FORMAT was writing the test codes to the disk, or "On Verify", when it was re-reading them. The "Sector #" gives the exact location where the error was found.

The "Error Code" shows what kind of error has occurred. These codes are explained in Appendix C, "User's Guide to Troubleshooting the Vector 4." See that appendix for interpretation.

NOTE

If your system produces either of the above error messages, contact your dealer and inform him of the problem.

Reformatting of the Hard Disk

Reformatting works on only a selected logical drive of the operating system in use. For instance, you might want to re-format Drive B without changing Drives C and D. Reformatting does not affect CP/M-86 itself or any storage areas except the one selected from the function menu.

Reformatting may be done whether the system was booted from a diskette or from the hard disk. The menus are similar, as shown below. When booting was from a diskette, complete formatting of the hard disk also is allowed.

1. From the UTILITY menu, type [3] to select the FORMAT function.

This menu appears:

```
FORMAT

Available functions:

    [0] Exit function
    [1] FORMAT floppy
    [2] Winchester format*
    [3] Winchester reformat

NOTE: THE FLOPPY FORMATTING OPERATION WILL BEGIN
IMMEDIATELY UPON VALID SELECTION.  [ESC] CANCELS THE
FUNCTION AND EXITS.

Make selection from above:
```

*This option appears only when the operating system was booted from a diskette.

2. Type [3] to select "Winchester reformat."

The FORMAT function first checks which operating system is being used and how its file storage area is divided. Then it makes up a menu showing your choices for reformatting. Menus for a system like the one in the examples above would look like this:

Hard-disk-booted systems:

```
FORMAT - WINCHESTER REFORMAT

Available functions:

    [0] Exit function
    [1] LOGICAL DRIVE A (2 Mbytes)
    [2] LOGICAL DRIVE B (2 Mbytes)
    [3] LOGICAL DRIVE C (2 Mbytes)
    [4] LOGICAL DRIVE D (2 Mbytes)
    [5] LOGICAL DRIVE E (1 Mbytes)

NOTE: THE FORMATTING OPERATION WILL BEGIN IMMEDIATELY
UPON VALID SELECTION. [ESC] CANCELS THE FUNCTION AND
EXITS.

Make selection from above:
```

Floppy-booted systems:

```
FORMAT - WINCHESTER REFORMAT

Available functions:

     [0] Exit function
     [1] CP/M-86 Partition 1 (2 Mbytes)
     [2] CP/M-86 Partition 2 (2 Mbytes)
     [3] CP/M-86 Partition 3 (2 Mbytes)
     [4] CP/M-86 Partition 4 (2 Mbytes)
     [5] CP/M-86 Partition 5 (1 Mbytes)

NOTE: THE FORMATTING OPERATION WILL BEGIN IMMEDIATELY
UPON VALID SELECTION. [ESC] CANCELS THE FUNCTION AND
EXITS.

Make selection from above:
```

For systems booted from a floppy, the "partitions" are the logical drives of the hard disk. Partition 1 is always Drive A; Partition 2 is the next hard-disk drive (Drive B, unless that was assigned to a diskette drive); Partition 3 is the next, and so on.

3. Enter the number of the area you wish to reformat.

EXAMPLE: To reformat Drive B:

```
Make selection from above: 2 [RETURN]
```

When this function is done, the menu reappears, to allow reformatting of other areas.

4. To continue, enter the function number of the next area.

5. To end the function and exit to the operating system command mode, type [Ø].

FORMATting of Floppy Diskettes

FORMATting of diskettes is simpler than for the Winchester disk, because diskettes are never partitioned. This means also that there is no reformatting of diskettes, since reformatting applies only to selected partitions of the hard disk.

Diskettes are formatted mainly to prepare them for storing files. Since different operating systems use different file types, diskettes must always be formatted under the operating system whose files they will hold.

CP/M-86 will not format single-sided floppy diskettes, although it will read any you may have from earlier systems.

The menus for the floppy format option differ depending on the configuration of your system's drives, as shown below.

1. From the UTILITY menu, type [3] to select the FORMAT function.

This menu is displayed:

```
FORMAT

Available functions:

     [0] Exit function
     [1] FORMAT floppy
     [2] Winchester format
     [3] Winchester reformat

NOTE: THE FLOPPY FORMATTING OPERATION WILL BEGIN
IMMEDIATELY UPON VALID SELECTION.  [ESC] CANCELS THE
FUNCTION AND EXITS.

Make selection from above:
```

2. Type [1] to select "FORMAT floppy."

Hard-Disk Systems and Single-Floppy Systems

On both systems with a hard disk and systems with one floppy drive, this message appears:

```
FORMAT

NOTE: [ESC] CANCELS FUNCTION AND EXITS.

Insert diskette to be formatted in drive and press [RETURN] when
ready:
```

3. Insert the diskette to be formatted in the diskette drive, then press [RETURN].

This message is displayed:

```
- FLOPPY FORMAT IN PROGRESS -
```

The drive activity light comes on while the FORMAT function takes place. When the function is done, the selection menu reappears, allowing you to format additional diskettes if you wish. Pressing [RETURN] ends the FORMAT session.

Systems with Dual Diskette Drives

If your system has two floppy drives, this menu appears:

```
FORMAT

Available functions:

     [0] Exit function
     [1] FORMAT floppy in Drive A
     [2] FORMAT floppy in Drive B

NOTE: THE FLOPPY FORMATTING OPERATION WILL BEGIN
IMMEDIATELY UPON VALID SELECTION. [ESC] CANCELS
FUNCTION AND EXITS.

Make selection from above:
```

NOTE: Do not press a key for selection until the diskette is in the drive.

3. Insert the diskette to be formatted in the selected drive. Drive A is on the right, Drive B on the left. Then press [RETURN].

When the function is done, the selection menu reappears. Pressing [RETURN] ends the FORMAT session.

Exiting the FORMAT Function

Often a new diskette is formatted so that it can be used immediately to store files. In that case, you needn't remove the diskette from the drive when formatting is done but can go directly to the application.

4. To end the Floppy Format function and return to the CP/M-86 command level, press [RETURN].

THE SYSGEN (System Generation) FUNCTION

System generation is the process of making a copy of the operating system onto the hard disk or a diskette. The disk or diskette may then be used for booting and to run the computer system.

If a disk or diskette has never been formatted, this must be done, using the FORMAT function of UTILITY, before SYSGEN can be used. In addition, the operating system can be copied only to double-sided diskettes.

At the outset, the System Generation function checks your system for two things: the kind and number of disk drives, and how the operating system already running was booted. It then offers one or more ways the function can be used for your system:

* If your system has no hard disk, the operating system must have been booted from a floppy diskette. The SYSGEN function can be used to make copies of the system diskette.

* If your system has a hard disk but was booted from a floppy, you can use SYSGEN to install an the operating system on the hard disk or to make a backup diskette. These choices are offered.

* If your system has a hard disk that also was used in booting, you don't need to install the operating system there, so the function assumes you want to make a backup diskette.

These three uses of the System Generation function are described below for each system type: single-floppy, dual-floppy, or both floppy- and hard-disk drive. See the section that covers your system's configuration.

Although SYSGEN copies the "kernel" instructions of an operating system (these have the .SYS (System) extension), you still must copy all those program files that make up the complete operating system (these have a .CMD or .COM extension in Vector 4 CP/M-86 and are the transient commands). See "Copying Other Operating System Files" below for details.

Single-Floppy Systems

On systems with one floppy-diskette drive, SYSGEN is used to make a backup copy of CP/M-86 on another diskette. Since you can't get along without your system diskette—you need it for daily booting—it's important to have at least one copy of the diskette for use in case the original becomes faulty.

1. Make sure you have a double-sided diskette, formatted through the FORMAT function, ready to use as the destination diskette.

2. Call the UTILITY program as described above and type [1] to select the System Generation function.

This message appears on the screen:

> **SYSTEM GENERATION**
>
> **NOTE: [ESC] CANCELS THE FUNCTION AND EXITS**
>
> **Insert diskette containing system in drive and press [RETURN] when ready:**

3. Place your System Diskette (in the diskette drive, close the drive door, and press [RETURN].

You will see the drive activity light come on while a temporary copy of the operating system is moved into the computer's internal memory. After a few moments, a second message appears:

> **Insert diskette to be Sysgened in drive and press [RETURN] when ready:**

4. Remove your System Diskette from the drive, then insert the diskette that will receive the copy.

5. Close the drive door and press [RETURN].

This message appears:

> **- FLOPPY SYSTEM GENERATION IN PROGRESS -**

The activity light comes on again while the temporary copy is transferred from memory to the new diskette. After a few moments, the UTILITY menu reappears on the screen.

Dual-Floppy Systems

On systems with two floppy-diskette drives, the System Generation function also is used to make a backup copy of the operating system. Because there are two drives, however, the steps are slightly different. Rather than having to switch diskettes in and out of a single drive, you can put each of them in a separate drive.

1. Make sure you have a formatted, double-sided diskette ready to use as the destination diskette.

2. Call the UTILITY program, then type [1] to select the System Generation function.

This message appears on the screen:

```
SYSTEM GENERATION

NOTE:  [ESC] CANCELS THE FUNCTION AND EXITS

Insert diskette to be Sysgened in Drive B and press [RETURN] when
ready:
```

3. Insert a double-sided diskette, formatted through the FORMAT function, in the diskette drive on the left, Drive B.

4. Close the drive door and press [RETURN].

This message appears:

```
- FLOPPY SYSTEM GENERATION IN PROGRESS -
```

The drive activity lights flick on and off while a copy of the operating system is transferred from the source diskette to the new diskette. When the function is done, the UTILITY menu reappears.

Hard-Disk Systems

On systems with both a floppy drive and a hard-disk drive, SYSGEN can be used to install CP/M-86 on the hard disk or to make a backup copy on a diskette, as with floppy-only systems (above). It's convenient to copy the operating system to the hard disk, because it then boots automatically (you no longer need to insert a system diskette). If your system includes a hard disk, this is the set-up you should use.

System Generation to the hard disk may already have been done by your sales representative, or by Vector Graphic, or by an experienced member of your staff. Still, it sometimes is necessary to run SYSGEN again for the hard disk—for example, if you have had to re-format the entire disk (formatting erases everything, including the CP/M-86 file itself).

Using SYSGEN to Install an Operating System on the Disk

If your Vector 4 system is completely new--CP/M-86 was not installed previously and nothing is on the hard disk--you must format the disk before CP/M-86 can be installed and before any files can be put there. See "The FORMAT Function" above for steps.

If your system is not new—if you have been using it with Vector 4 CP/M--you also need to format the disk before installing CP/M-86, but first you must save whatever is already on the disk. See Section XI, "Installing CP/M-86 on Your Hard Disk," for steps. DO NOT format the hard disk until you have completely copied its contents to backup diskettes or tape (if your system includes a SAFSTOR tape drive). FORMAT erases all data, and if you have not saved your files they will be lost.

1. Make sure the disk is formatted. (See "The FORMAT Function.")

2. Call the UTILITY program, then type [1] to select the System Generation function.

This message appears:

```
SYSTEM GENERATION

Available functions:

                 [0] Exit function
                 [1] Floppy system generation
                 [2] Winchester system generation

NOTE:   [ESC] CANCELS THE FUNCTION AND EXITS

Make selection from above:
```

3. Type [2] to select "Winchester system generation."

This message appears:

```
 - WINCHESTER SYSTEM GENERATION IN PROGRESS -
```

You will see the activity lights of the two drives come on as the function copies the operating system from the floppy diskette to the hard disk. Then, when the function is done, the UTILITY menu reappears.

4. Remove your system diskette from the diskette drive and put it away for safekeeping.

The operating system is now installed on the hard disk and can be used for booting and daily operation. At this point, unless you're through with the system for the day, press the RESET button at the rear of the console and then re-boot from the hard disk.

Using SYSGEN to Copy An Operating System to a Backup Diskette

If you have a hard-disk system and booted from the hard disk, the System Generation function knows it doesn't have to install CP/M-86, so it assumes that you want to make a copy to a backup diskette.

1. Call the UTILITY menu, then type [1] to select the System Generation function. This message appears:

```
SYSTEM GENERATION

NOTE: [ESC] CANCELS THE FUNCTION AND EXITS.

Insert diskette in drive and press [RETURN] when ready:
```

2. Insert a double-sided diskette, formatted through the FORMAT function, in the diskette drive.

3. Close the drive door, then press [RETURN].

The screen displays:

```
- FLOPPY SYSTEM GENERATION IN PROGRESS -
```

The drive activity lights come on as the operating system is copied from the hard disk to the diskette. When the function is done, the UTILITY menu reappears.

Copying Other Operating System Files

After SYSGENing an operating system to the hard disk or a diskette, you can boot from the new copy. But SYSGEN has not copied the complete operating system, because that also includes the many command files not in the kernel of system instructions. For example, in CP/M-86 there are all the Transient Commands: programs such as PIP, STAT, and HELP. And on Vector 4s, any operating system also takes in the special programs CONFIG and UTILITY. To copy an operating system completely, you must also copy these command files.

In CP/M-86, you do this using the PIP command. (For other operating systems, use the corresponding "copy" command. See the appropriate Vector Graphic manual for the operating system.) The transient command functions all are stored as command files that have either the extension .CMD or .COM, so it's easy to use the asterisk (*) wildcard to copy all the files at once.

If you have copied CP/M-86 from a diskette to the hard disk or to another diskette:

1. Re-insert the System diskette in the diskette drive. The PIP function is on the System diskette, and you need access to it.

2. Enter the PIP command for repeat mode:

```
d>PIP [RETURN] *
```

*NOTE: Here and below, "d" stands for whatever drive letter has been assigned the floppy drive.

When the asterisk cursor appears, enter these commands (enter one command each time the asterisk re-appears):

```
A:=d:*.CMD [RETURN]
A:=d:*.COM [RETURN]
A:=d:*.DEV
```

The * wildcard in the filespecs names all files on the diskette drive with the extensions given, and these are copied to Drive A (the hard disk). The "DEV" files are the three printer driver files.

If you booted from the hard disk (CP/M-86 already is installed there):

1. Re-insert the newly SYSGENed diskette (the diskette that received the copy) in the diskette drive.

2. Enter the same PIP commands as above but copying from Drive A to the diskette drive:

```
A>PIP [RETURN]

d:=A:*.CMD [RETURN]
d:=A:*.COM [RETURN]
d:=A:*.DEV [RETURN]
```

These steps complete the copying of CP/M-86.

THE DISKCOPY FUNCTION

The DISKCOPY function allows you to make a quick and complete copy of one floppy diskette to another.

DISKCOPY does not require that the destination diskette (where the copy will be made) be formatted first. You can copy to an already-used diskette or to a new diskette without having to run the FORMAT function first.

DISKCOPY may be used only for double-sided diskettes.

If your system was booted from a hard disk and if there is enough space available, the disk is used for temporary storage during copying. Data on the source diskette is copied to the hard disk, then DISKCOPY prompts you to remove the source diskette from the drive and insert your destination diskette. DISKCOPY then copies the data from the hard disk to the destination diskette.

If the Winchester doesn't have enough space, DISKCOPY copies as much as it can from the source diskette into internal memory (RAM), then prompts you to insert the destination diskette. It copies the part it has stored onto it. Then you must re-insert the source diskette, and the function takes another portion. This process repeats until the source diskette has been completely copied.

If your system has two floppy diskette drives, the DISKCOPY is made from one drive to another.

DISKCOPY for Systems with One Diskette Drive

1. From the UTILITY menu, type [2] to select "Floppy Diskcopy."

   The screen shows:

   FLOPPY DISKCOPY - SINGLE DRIVE CONFIGURATION

   NOTE: [ESC] CANCELS THE FUNCTION AND EXITS.

   - WINCHESTER IS BEING USED FOR TEMPORARY STORAGE -*

   Enter the number of copies desired:

*This message appears only if the system was booted from a hard disk and there is enough space available.

2. Enter the number of copies to be made of the source diskette.

   Insert source diskette and press [RETURN]:

3. Put the diskette to be copied in the diskette drive, close the drive door, and press [RETURN].

DISKCOPY reads the contents of the diskette, then the screen displays:

Insert destination diskette #n* and press [RETURN]:

*Here, "n#" is replaced by the number of diskette in the sequence.

4. Remove the source diskette from the drive and insert your destination diskette (the diskette that is to receive the copy).

5. Close the drive door and press [RETURN].

If the hard disk had a diskette's worth of space available, DISKCOPY can complete the copy in one try. When copying is complete, the main menu reappears on the screen. If RAM was used and only a portion of the diskette could be copied, you are prompted to re-insert your source diskette. DISKCOPY takes more data from the diskette and stores it. Then this message again appears—

```
Insert destination diskette and press [RETURN]:
```

—and copying continues. When all data from the diskette has been transferred, the UTILITY menu reappears.

6. Remove the destination diskette from the drive.

DISKCOPY for Systems with Two Diskette Drives

1. From the UTILITY menu, type [2] to select "Floppy Diskcopy."

   The screen displays:

```
FLOPPY DISKCOPY - TWO DRIVE CONFIGURATION

Available functions:

        [0] Exit function
        [1] Copy from A to B
        [2] Copy from B to A

NOTE:   THE FLOPPY COPYING OPERATION WILL BEGIN
        IMMEDIATELY UPON VALID SELECTION. [ESC] CANCELS
        THE FUNCTION AND EXITS.

        Select one of the above:
```

DISKCOPY copies from Drive A to Drive B or vice versa in the same way. It only matters that you make sure your source and destination diskettes are in the correct drives when you press a key to select the function.

EXAMPLE: To copy from Drive A to Drive B (Function 1):

1. Place the source diskette (the diskette to be copied) in Drive A (on the right) and close the drive door.

2. Place the destination diskette (the diskette that will receive the copy) in Drive B (on the left). Close the drive door.

3. Press [1] to select "Copy from A to B."

When the function is done, the menu reappears to allow you to make additional copies of the same diskette or to copy other diskettes.

EXAMPLE:

To make additional copies of the diskette in Drive A:

4. Remove the diskette from Drive B and insert another diskette.

5. Press [1] to again select "Copy from A to B."

To end a session with DISKCOPY, press [RETURN]. The main menu reappears.

In any session with DISKCOPY, if an error message appears, write down the message, then see Appendix C, "User's Guide to Troubleshooting the Vector 4" for information.

The CONFIG (Configuration) Utility

CONFIG customizes your Vector 4 for use of different of peripheral devices, such as printers and modems, and allows you to select different operating modes for the keyboard, screen, and disk drives. CONFIG itself is a kind of "master program" that includes many individual commands. You can use these to—

- o select and control the type of printer in use
- o set the layout of the page
- o adjust the baud rate of a serial printer or modem
- o redefine keys on the keyboard to produce new characters or complete commands
- o set the keyboard to operate in one of several different modes
- o set the video driver software for different modes and character sets
- o set error-checking features for the disk-drive control programs
- o introduce an auto-command to the booting procedure

How CONFIG Commands Are Entered

The system runs CONFIG commands in several different ways.

1 - In the highest-level approach (the most automatic), a complete set of CONFIG commands, stored in a special file, is executed automatically during booting. By the time the system prompt appears on the screen—the operating system is loaded and you can begin work--all the needed configuring has been done.

2 - In a second way, like the first but less automatic, CONFIG commands again are stored in a file, but you start execution of the commands yourself. In both the first two ways, you can create and change the CONFIG files as needed for your system.

3 - Finally, you can type CONFIG commands one at a time in the operating system command mode.

The first way lets your system do its housework automatically as soon as it is started up. The second lets you select and use one of several CONFIG files, each geared to a different need. (For example, you can quickly configure the system to use different printers or application programs.) The third lets you change your system's configuration in mid-stream by entering CONFIG commands at the system command level.

The CONFIG Auto-Command File

The system runs the automatic execution of CONFIG commands during booting this way:

For each operating system installed on your system, a special filename is reserved for a CONFIG auto-command file. This filename has the extension .BEG (short for "Begin"). The rest of the file name is the name of operating system. For CP/M-86, the complete file name is "CPM-86.BEG" (note that the slash is gone from the name). For MS-DOS 2.0, it is "MS-DOS2.BEG". During booting, as the bootstrap program loads and initializes the selected operating system, it searches for the .BEG file in disk memory. If it finds it, as its last booting job it executes the CONFIG commands the file contains. Then when the system prompt appears on the screen, all configuring of the system is already done.

If your system includes a hard disk and you have several operating systems installed, there is a different .BEG file for each of them. During booting of any one of the operating systems, the loader program looks for the appropriate .BEG file.

Vector Graphic supplies a sample .BEG files for each operating system it offers. This file does some basic configuring for the system: it sets up two printer-driver programs, selects one for use (the serial printer), and programs the HELP key to output the command "HELP" (to call the HELP program).

The sample .BEG file also produces the messages that appear on the screen just after booting (above the A> prompt):

```
SERIAL PRINTER DRIVER LOADED
PARALLEL PRINTER DRIVER LOADED
SERIAL PRINTER SELECTED AS LIST DEVICE
```

These messages are the result of one of CONFIG's commands, REMARK, which can be used in the .BEG file to display information about how the file's other commands are configuring the system.

The Vector Graphic-supplied .BEG file can be edited to include other CONFIG commands you may need. Your dealer may already have added some commands to better suit the file to the applications intended for your system. As shipped from Vector Graphic, the file contains these commands:

```
KCODE "HELP  H  E  L  P  RETURN"
PLOAD 1 SERIAL.DEV
REMARK "SERIAL PRINTER DRIVER LOADED"
PLOAD 2 PARALLEL.DEV
REMARK "PARALLEL PRINTER DRIVER LOADED"
PSEL 1
```

```
REMARK "SERIAL PRINTER SELECTED AS LIST DEVICE"
```

You can see how the REMARK command is used to produce the screen display. Notice also that, for all the commands, the utility name "CONFIG" is not part of the command line (as it is when the commands are entered at the CP/M-86 command level). This is because the CONFIG utility name is instead "entered" by the bootstrap program: it calls the utility, passing to it the .BEG filename as the source of its instructions.

The .BEG file can be edited using SCOPE or ED (the Vector Graphic and CP/M-86 text editors). See "Creating and Editing CONFIG Batch Files" below. What the rest of the commands mean also is discussed later this section.

Other CONFIG "Batch" Files

As noted above, you can edit and enlarge the .BEG CONFIG file. You also can create completely separate files of CONFIG commands to be executed when needed by command from the operating system command mode. This file is like a SUBMIT file in Vector 4 CP/M or CP/M-86, or like a .BAT (Batch) file in MS-DOS 2.0. A SUBMIT or Batch file actually could contain <u>any</u> commands that frequently must be entered in sequence. In this case, it is a "batch" of CONFIG commands.

You begin execution of commands in the file by entering a command of this format:

```
A>CONFIG /filename [RETURN]
```

For example, say you have created a file of CONFIG commands named "COM-CNFG.TXT" to configure your system for use of a communications device and teleprocessing. To begin execution of this file's commands, you would enter—

```
A>CONFIG /COM-CNFG.TXT [RETURN]
```

The slash (/) is an important part of the command line. It tells CONFIG to expect not a command but the name of a batch file containing multiple commands. In response, CONFIG gets the contents of this file and begins executing the the commands one by one just as it would if you entered them at the keyboard.

Your batch files for CONFIG are created using MEMORITE, SCOPE, or the CP/M-86 text editor, ED. See "Creating and Editing CONFIG Batch Files" below.

Entering CONFIG Commands at the CP/M-86 Command Level

CONFIG commands entered in the CP/M-86 command mode work the same as the commands in a .BEG or batch file but have a slightly longer format, because the command line must also include the CONFIG program command ("CONFIG"). And, as usual for CP/M-86 commands, the entire command line is sent to processing by [RETURN]. Command lines in the batch files, as seen above, do not include "CONFIG" or [RETURN].

EXAMPLE: To load and select the serial-printer driver (as in the sample .BEG file shown above):

```
A>CONFIG PLOAD 1 SERIAL.DEV [RETURN]
A>CONFIG PSEL 1 [RETURN]
```

Entering Multiple CONFIG Commands

When typing CONFIG commands at the operating system command level, you may enter multiple commands on a command line--that is, following one "CONFIG" utility name. For example, if entered from the CP/M-86 command mode, the PLOAD and PSEL commands shown above might also be typed this way:

```
A>CONFIG PLOAD 1 SERIAL.DEV  PLOAD 2 PARALLEL.DEV  PSEL 1  [RETURN]
```

This is also true for commands put into a CONFIG batch file: the commands and their parameters may be entered one after another on each line, if desired.

In all of the examples in the sections below, both the utility name "CONFIG" and [RETURN] are omitted, since the commands could appear either on a CP/M-86 command line or in a batch file.

CONFIG Commands and Command Parameters

CONFIG commands fall into five groups:

1 Printer Commands

2 Communications Commands

3 Disk System Commands

4 Keyboard/Video Driver Commands

5 Miscellaneous Commands

The basic format for all CONFIG commands is the same:

command parameter(s)

Within the CONFIG utility are all the separate commands making up the five command groups. Most of these commands are entered with parameters, "modifying" information that directs the execution of the command one way or the other. For instance, in the PLOAD commands of the sample .BEG file above, the parameters are the printer-driver numbers--"1" and "2"--and names—"SERIAL.DEV" and "PARALLEL.DEV". These tell CONFIG what list-device files are to be loaded.

As noted above, CONFIG command lines entered from the operating system command mode (when the system prompt is displayed) must include the utility name, "CONFIG", and end with [RETURN]. CONFIG commands put into a batch file (such as the .BEG file) are typed in without either "CONFIG" or [RETURN] as part of the line. In addition, in either way of entering the commands, multiple commands and parameters may be placed on one line.

Delimiters in CONFIG Commands

Some CONFIG commands require delimiters, characters that mark off the beginning and end of a parameter. The commands are those that have parameters consisting of a string of characters that might include a space.

Any characters can be used as delimiters. If delimiters are needed in a command, CONFIG assumes that the first character it finds after the actual command word is the delimiter. Then it looks for the next occurrence of the character to mark the end of the output string (two delimiters are always required to frame the string).

It's simplest to pick as delimiters characters that are regularly used in English for the same purpose: the single- and double-quotation marks or the slash mark. If your string includes one of these already, however (say, double quotation marks), you must choose a different character for the delimiters.

For example, if you wanted to print the REMARK message—

THIS "MESSAGE" CONTAINS QUOTES

—you would enclose the parameter (the string of message characters) using a delimiter other than the double quotes—using, for instance, the slash mark.

The command would then be typed like this:

```
REMARK /THIS "MESSAGE" CONTAINS QUOTES/
```

The sections below show how to use each kind of CONFIG command.

THE PRINTER COMMANDS

The Vector 4 can support (have attached and run) three different kinds of printers. To control these, the operating system uses special sets of instructions called printer drivers. CP/M-86 for your Vector 4 has three printer drivers, one for each type of printer. They are stored on the disk in files with these names:

1 NEC.DEV

2 SERIAL.DEV

3 PARALLEL.DEV

These files are called list-device files, and if you are using either a Vector 7700 or 3500 letter-quality printer, one of the list-device files, NEC.DEV, already has been programmed to operate those printers. The other two drivers, SERIAL.DEV and PARALLEL.DEV, are designed to handle any printer that fits into the general types "Serial" or "Parallel" (electronic terms for how data is sent between the computer and printer) unless the printer has been modified for some special application.

If you are using a printer other than the Vector-standard 7700 or 3500, you need to know its type: whether it is serial or parallel. You also need to make sure that the list-device files on your disk or diskette (SERIAL.DEV and PARALLEL.DEV) are right for controlling it. Because it is a non-standard printer, you may need a special driver to run it, or you may need to make changes in the printer drivers Vector Graphic provides. See the operating instructions for your printer to determine which list-device file to use. Then refer to the **Vector 4 CP/M Programmer's Guide** for instructions on how to make needed changes in the printer drivers.

The first of the printer commands, PLOAD, moves a driver program from where it is stored on disk into internal memory, where it actually can be used. Because all the other printer commands modify settings in the driver, none of them can have any effect until PLOAD has been entered to make the driver available. PLOAD must be entered before any other printer command is used.

To print while working in CP/M-86 (rather than in an application program), you must use CONFIG commands to select the printer. Some application programs, however, can handle printer selection for you. For example, if you print exclusively from Vector Graphic's MEMORITE Word Processing program, you should not have to use CONFIG to select the printer. MEMORITE contains commands to manage this for you. To know whether this is also true for other of your application programs, check the manual for the program, looking for information on selecting and attaching printers.

The PLOAD Command

PLOAD moves a printer-driver from the disk into its work location in internal memory. It also initializes the driver: it tells the driver how to "talk" to the printer--how data is to be sent to it. As we saw earlier in the section on batch CONFIG files, it's possible to use two PLOAD commands to load two different drivers. Then one of them can be selected for use by the PSEL command (see below). This offers flexibility: if you have two printers connected on your system, you can move quickly from one to the other just by entering PSEL for the desired printer.

FORMAT:

PLOAD <driver #> <d:filename.ext>

The <driver number> is either "1" or "2."

For <d:filename.ext>, give the name of the driver program: NEC.DEV, SERIAL.DEV, or PARALLEL.DEV, or the name of a special driver if you are using one. (The drive letter can be omitted if CONFIG is on your local drive.)

PLOAD gets the named file from the disk and connects the name of the file to the driver number. Then, using PSEL, you can select the program by number.

EXAMPLE: To load the NEC.DEV program as Driver #2:

PLOAD 2 A:NEC.DEV

EXAMPLE: To load two driver programs, SERIAL.DEV and NEC.DEV, as Drivers 1 and 2:

PLOAD 1 A:SERIAL.DEV PLOAD 2 A:NEC.DEV

Remember to turn on your printer before using PLOAD, to make sure that the command can do the needed initializing of the driver. But if you forget, you can use the PINIT command (see below) to initialize the driver.

The PSEL (Printer Select) Command

The PSEL command is the logical follow-up to PLOAD. Since you could load two printer drivers at one time, you need a command to select one program (and printer) for use. This is PSEL: it says that output from a program should go through the named printer driver to the associated printer. PSEL also gives two other options: you can direct output to the screen or "discard" it ("Ignore output").

FORMAT:

PSEL <option>

The four options are:

Option #	Output sent—
Ø	Ignore output
1	To Printer 1
2	To Printer 2
3	To the screen

EXAMPLE: To select Printer 2 (the driver that has been loaded as "Driver 2" and the printer it controls):

PSEL 2

In the examples above (for PLOAD), Driver 2 was NEC.DEV. So this PSEL command selects the Vector 7700 or 3500 printer to print your output.

EXAMPLE: To direct the output of your program to the screen:

PSEL 3

This option sends the "printout" to the console screen. You probably will use this only if you are training in the commands for printing while no printer is attached to the system.

EXAMPLE: To cause program output to be ignored:

PSEL Ø

Choose this option if you do not have a printer connected to the system. In addition, some application programs (such as MEMORITE) handle selecting and attaching the printer directly. (See the program manual, when using other application programs, for information on printers.)

The PSIZE Command

The PSIZE command specifies the size of the form in use. The printer driver uses this value to calculate when to advance the paper to a new sheet. "Size" means the actual length of the sheet, not the size of the print area (see the PLINES command below).

FORMAT:

PSIZE <driver #> <page size in lines>

EXAMPLE: To set the page size for Printer 2 to 76 lines:

```
PSIZE 2 76
```

The PLINES Command

The PLINES command gives the size of the print area on each page. The print area includes the top margin.

FORMAT:

PLINES <driver #> <number of printed lines>

EXAMPLE: Printer 2 has been set for a top margin of 10 lines (by the PMARGIN command—see below). To get 56 lines of printing per page, enter—

```
PLINES 2 66
```

In other words, the number in the PLINES command is the last line of printing for each page (in this example, line 66).

The PMARGIN (Print Margin) Command

The PMARGIN command sets the size of the top margin.

FORMAT:

PMARGIN <driver #> <margin size in lines>

EXAMPLE: To set Printer 2 to allow a top margin of 10 lines:

```
PMARGIN 2 10
```

Checking Format Settings of PSIZE, PLINES, and PMARGIN

Obviously, the values you enter using PSIZE, PLINES, and PMARGIN should be "possible." That is, the number of lines-per-page set by PLINES shouldn't be greater than (or even equal to, normally) the overall size of the sheet. The program might not check for "impossible" or even unsightly format settings, so plan your formats. It's helpful to draw the actual layout on a sheet of paper to get an idea of the effect of your commands. You can use a ruler to figure the size of margins and print area if you know your printer's setting for lines-per-inch (for example, six lines of printing per inch is common).

Also note that none of the format-control commands works unless the next command, PAUTO, is enabled.

The PAUTO (Printer Auto-Page) Command

The PAUTO command enables and disables the auto-page feature of the printing program. When enabled, auto-paging keeps track of the format counts for page size, lines-per-page, and so on, then advances the form and sets the top margin as specified.

FORMAT:

```
PAUTO  <driver #>  <mode>
```

When <mode> is 1, auto-paging is enabled. When it is Ø, auto-paging is disabled.

EXAMPLE: To enable the auto-paging feature for use by Printer 1:

```
PAUTO  1  1
```

As noted above, PAUTO must be enabled for PSIZE, PLINES, and PMARGIN to have any effect.

The PBAUD (Printer Baud Rate) Command

The PBAUD command adjusts the baud rate for a serial printer. Serial printers receive information from the computer one bit at a time, rather than in groups as do parallel printers. The baud rate is the rate at which printer/computer communications take place.

For your system, seven baud rates are possible:

110 **300**
600 **1200**
2400 **4800**
9600

FORMAT:

```
PBAUD  <driver #>  <rate>
```

EXAMPLE: To set the baud rate for Printer Drive 1 to 300:

```
PBAUD  1  300
```

How do you know what baud rate to select? The user's manual for your serial printer should tell you. Refer to the specifications section of that manual, or check its table of contents for information on selecting the baud rate. Both the Vector 7700 and 3500 are parallel printers and no PBAUD command is needed.

The PINIT (Printer Initialization) Command

The PINIT (Printer Initialize) command tells the printer driver what protocol (way of communicating) it will use to send data to the printer. The PLOAD command also does this initializing of the driver, so you need to use PINIT only if you gave the PLOAD command when the printer was turned off or if the printer doesn't seem to be receiving data properly and you want to give it a fresh start.

(All of the other printer commands except PSEL also initialize the driver automatically whenever they are used.)

FORMAT:

```
PINIT <driver #>
```

As before, <driver #> is "1" or "2."

EXAMPLE: To initialize printer driver 2:

```
PINIT 2
```

THE COMMUNICATIONS COMMANDS

CONFIG's communications commands control and modify the driver for a telecommunications device—usually, a modem, which allows your system to send and receive information over the telephone lines. As with the printer commands, one communication command (CLOAD) transfers the driver from disk to its work location in RAM. Other commands modify the program to adapt it to the modem in use and to its protocol for sending and receiving data.

There is only one communications channel for the system, so neither a "Select" command (to choose between different drivers) nor driver numbers are needed.

See the user's manual for your communications device for information about the correct value to be used with the commands described below.

The CLOAD Command

Your system was shipped with one communications driver program, named ACOM.DEV, on the System Diskette. The CLOAD command moves this program from disk into its RAM work location and initializes it.

FORMAT:

```
CLOAD <d:filename.ext>
```

EXAMPLE:

CLOAD ACOM.DEV

No driver number is needed for CLOAD, because there is only one resident communications driver at a time.

The CINIT (Communications Initialization) Command

CINIT readies the modem for operation, putting the computer-to-modem lines in their "start" state. (CLOAD also initializes the communications channel, as do all of the other communications commands whenever they are used.)

FORMAT:

CINIT

EXAMPLE: To re-initialize the communications channel:

CINIT

The CBAUD Command

The CBAUD command adjusts the baud rate for the communications device. Like serial printers discussed above, modems pass information to and from the computer one bit at a time. The baud rate is the rate at which communications take place.

The seven baud rates that apply for the serial printer driver also are allowable here. CBAUD must specify one of these:

110	**300**
600	**1200**
2400	**4800**
9600	

FORMAT:

CBAUD <rate>

Note again that you need not specify a driver number.

EXAMPLE: To set the baud rate to 1200:

CBAUD 1200

The CPARITY Command

The CPARITY command indicates whether parity-checking will be used.

FORMAT:

CPARITY <mode>

<Mode> may be	0 = No parity checking
	1 = Even parity
	2 = Odd parity

"Even parity" means that each "word" of information sent has an even number of "one" bits. "Odd parity" means that each word should have an odd number of one bits.

EXAMPLE: To set the modem for an "even parity" protocol:

CPARITY 1

In parity checking, the sender and receiver agree that the "words" of information they exchange will be on a par in terms of the number of "one" bits they contain. Data words will always have odd parity (an odd number of one bits) or even parity (an even number). Special hardware adds a parity bit to each data word when necessary to make it fit into the scheme.

Parity checking causes bad data--information that has changed during transmission—to stand out because of its wrong bit count.

The CSTOP Command

The CSTOP command indicates the number of "stop bits" to be used. Stop bits are added to transmitted data to aid in timing and to provide needed delays.

FORMAT:

CSTOP <number>

<Number> may be	1 = One stop bit
	2 = Two stop bits

EXAMPLE: To set the device to a communications protocol of two stop bits:

CSTOP 2

The CWORD Command

The CWORD command gives the length of the data word sent along the communications line, in number of bits.

FORMAT:

CWORD <length>

"Length" may be

7 = Seven bits to a data word, or
8 = Eight bits to a data word

EXAMPLE: To set for use of data words of seven bits:

CWORD 7

THE DISK SYSTEM COMMANDS

CONFIG's disk system commands turn on (enable) or turn off (disable) several error-checking features of the control program for the disk drives. Each of these error checks is already at a "default" setting when you receive your system, and normally you need not alter these settings. The disk system commands are useful mainly in advanced programming, when speed in the use of the disk memory is more important than the usual margin of safety.

The DSOFT Command

DSOFT enables/disables soft-error reporting.

> NOTE: DSOFT is for diagnostic testing only and should not be used in normal system applications. The system initially is loaded with DSOFT disabled.

FORMAT:

DSOFT <mode>

Mode may be Ø = Disable
1 = Enable

EXAMPLE: To enable soft-error reporting:

DSOFT 1

The DRETRY Command

DRETRY enables/disables the error-retry feature.

> CAUTION: When DRETRY is disabled, the disk system does not try to recover after an error is encountered. DRETRY should be used only during diagnostic procedures. The system initially is loaded with DRETRY enabled.

FORMAT:

DRETRY <mode>

Mode may be	Ø = Disable
	1 = Enable

EXAMPLE: To disable the error-retry:

DRETRY Ø

The DCHECK Command

DCHECK enables/disables the pre-write ID check.

> CAUTION: DCHECK allows the user to disable the disk position check, which normally is required to maintain reliability in the system. The system initially is loaded with DCHECK enabled.

FORMAT:

DCHECK <mode>

Mode may be	Ø = Disable
	1 = Enable

EXAMPLE: To enable the pre-write ID check:

DCHECK 1

The DVERIFY Command

DVERIFY enables/disables the error-checking feature that reads data newly-written to the disk and compares the "source" and "disk" versions to check for differences.

> WARNING: When DVERIFY is disabled, the disk I/O speed will increase with a sacrifice of system reliability. The system initially is loaded with DVERIFY enabled.

FORMAT:

DVERIFY <mode>

Mode may be ∅ = Disable
             1 = Enable

EXAMPLE: To disable the verify check:

DVERIFY ∅

The DPROMPT Command

DPROMPT enables/disables the error prompting feature.

> NOTE: DPROMPT, when enabled, allows the user to decide on recovery procedures by prompting him for input. The system initially is loaded with DPROMPT disabled.

FORMAT:

DPROMPT <mode>

Mode may be ∅ = Disable
             1 = Enable

EXAMPLE: To disable error prompting:

DPROMPT ∅

THE KEYBOARD COMMANDS

The keyboard commands in CONFIG allow you to redefine the values of keys on the Vector 4's keyboard. Each key can be programmed to output a character or a string of characters. The string can be a key-combination, such as a CTRL sequence, or a program command. CONFIG places the conversion values in a translation table, and this table becomes the source of each key's "meaning" in later work.

Why Re-Program the Keyboard?

Assigning new values can be helpful in application programs, which often use CTRL- and ESC-key combinations. It also can simplify work with high-level languages such as FORTRAN and BASIC, which call for repeated use of certain commands--for example, PRINT or LIST. In this case, one of the Vector 4's function keys could be defined to output the command word itself: **PRINT "** (in BASIC), or **WRITE** (in FORTRAN).

How Many Keys Can Be Redefined?

There is no fixed number of conversions that can be made. Any number of keys may be redefined so long as there is enough space left in the translation table. The translation table holds up to 511 characters. Any translation entry may be up to 82 characters long. (Spaces in the string count as characters, too.) So the number of conversions possible depends on the length of the output keycodes you select.

A "keycode" is simply the computer's internal form for a key. When you press a key, you send a unique code representing the symbol on that key to processing. If that code is in the translation table, CONFIG intercepts it and sends along different keycodes—the output keycodes—instead.

Actually, two of the characters in every translation entry are not part of the output keycodes; they are used by CONFIG instead. The first character (or, byte) is the input keycode, the standard value of the key. The second is the length of the output string. The third byte begins the output character or string itself. For example, re-programming the [F1] key to produce the string **Print "** (as in our example above) will take up nine bytes in the table: one for the input code of F1 key (its "standard" code), one showing the length of the output string, and seven for the string itself (five letters, one space, and the double-quotes mark = seven).

But you don't have to keep track too closely of how much space you have left. CONFIG lets you know when the table is full.

Creating Translation Files

You can simplify your work with various application programs and programming languages by creating separate conversion tables for each and saving them in files, to be recalled later for use when needed. To create a translation file, use Vector Graphic's SCOPE or MEMORITE, or the CP/M-86 text editor, ED. (See the **SCOPE User's Guide** and **MEMORITE IV Primer** for information on how to use SCOPE and MEMORITE. See Chapter 5 of the **User's Guide to CP/M-86** for details on use of ED.)

See "The KLOAD Command" below for a description of how multiple re-definitions could be entered in the file.

The KCODE Command

The KCODE command is used to redefine the value of a key. KCODE has the format:

KCODE <delimiter> <input keycode> <output keycode> <delimiter>

In effect, a KCODE command says to CONFIG for each key that is redefined: "When this key is pressed [and this keycode generated], send this translation value to the executing program."

KCODE is one of the CONFIG commands that requires delimiters (see "Delimiters in CONFIG Commands" above). The input keycode and output keycode following the command word must be marked off by characters such as double- or single-quotes or slashes. (Quotation marks are simple and are recommended.)

Each key on the Vector 4's keyboard produces a unique keycode. The keycodes are as follows:

HELP ESC TAB DEL BACKSPACE LINEFEED
RETURN ENTER SPACE

F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14

UP DOWN LEFT RIGHT

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

1 2 3 4 5 6 7 8 9 0 (as entered from the keys at the top of the keyboard)

#1 #2 #3 #4 #5 #6 #7 #8 #9 #0 #- #, #.

(The cross-hatch sign—#—indicates that these characters are enterec from the numeric keypad at the right of the keyboard. These keys have internal codes different from those of the other keys for the same characters.)

! @ # $ % ^ & * ( ) _ - + = { } [ ] ' " ; : ,
. < > ? / | ~ `

In other words, all of the characters appearing on the Vector 4's keyboard are available for use as input codes. The exceptions are the CTRL and SHIFT keys, which modify the value of other keys but have no character value themselves. Values for CTRL and SHIFT are shown as follows:

Parentheses around a value indicate CTRL of the code.

Brackets around a value indicate SHIFT of the code.

EXAMPLES:

CTRL-A	=	**(A)**
SHIFT F1	=	**[F1]**
CONTROL SHIFT F3	=	**([F3])**

(NOTE: Older Vector 4 keyboards may have an "F0" key in place of the "HELP" key and keys "C1" through "C4" in place of keys "F11" through "F14". In both cases, the keys on the older keyboards may be used as keycodes in KCODE commands in place of the keys on the newer keyboards.)

The lower-case letters of the alphabet are included as keycodes in table above even though CP/M-86 doesn't "see" lower-case letters when they are input at the command level. When you type commands in response to the CP/M-86 system prompt, your lower-case input is changed to upper-case. To redefine a key to output lower-case letters, you can include the re-definition in a translation file, created through MEMORITE or SCOPE, and load this file using the KLOAD command (see "The KLOAD Command" below for more on this).

Conversions already set up in the table are deleted if you enter the input code but omit any output code.

EXAMPLE: The F1 key was previously redefined to output the CP/M-86 command "DIR". The command to erase the re-definition from the translation table is:

KCODE "F1"

Examples of Reprogramming

Here are some possible applications of the reprogramming function.

1. To recode the F1 function key to output the string "LIST" followed by a carriage return (a common command in BASIC):

```
               <input keycode>     <output keycode>

KCODE     "F1              L   I   S   T    RETURN"
```

2. To recode the HELP key to output the command "HELP" followed by RETURN:

```
            <input keycode>        <output keycode>

KCODE      "HELP            H    E    L    P     RETURN"
```

Notice that, for the input keycode above, the letters of the word "HELP" are closed up. This is to inform CONFIG that the word represents one keycode, the output of the HELP key. In contrast, for the output keycode, the letters "H E L P" are spaced. This is because each letter represents an individual code, one for each key.

This can also be seen in Example 1, where the output keycode "L I S T" is spaced to show that each character represents a separate keycode.

Watch for this difference when using other "word" keys--such as SPACE, BACKSPACE, or TAB—in your commands. These stand for single keys that output one code, so their "names" should be typed closed up. The letters of an output string, however, must be spaced, because each letter corresponds to the output of a different key.

3. To recode the F6 function key to set the current drive to C, User Area 1:

```
            <input keycode>              <output keycode>

KCODE          "F6            C  :  RETURN  U  S  E  R  SPACE  1  RETURN"
```

In this command line, "RETURN", "SPACE" and "RETURN" represent the [RETURN] and [SPACE] keys, and since these each produce one keycode, these parts of the command line are spelled closed up. "U S E R", in contrast, represents four different keycodes, one for each letter, so these parts of the command line are separated by (at least) one space.

4. To recode the F10 function key to do a Memorite "Copy Section" command from within the Editor mode, save the source section, remove the section delimiters, and return to the Edit mode:

```
            <input keycode>            <output keycodes>

KCODE          "F10     (A)  C  S  RETURN  N  RETURN  Y  RETURN  ESC"
```

To use this function in a MEMORITE session, mark off the section to be copied using [DEL "] delimiters, position the cursor to the copy point, then press the [F10] key. As programmed, that key would produce five commands and responses to Memorite queries:

(A) = Enter Auxiliary Command Mode
(the parentheses around the "A" = "CTRL of key")

C S RETURN = Copy Section and Return

N RETURN = "Remove source section?" - NO and RETURN

Y RETURN = "Remove section delimiters?" - YES and RETURN

ESC = Return to Editing Mode

5. If an output keycode is a number with more than one digit, it is treated as a direct HEX value. The HEX value must begin with a decimal digit (that is, F3 HEX should be written ØF3). For example, to recode the shift up-arrow key to produce a CF HEX code:

   <input keycode><output keycode>

KCODE "[UP] ØCF"

The KLOAD Command

The KLOAD command loads a keyboard translation file into the conversion table area in memory. The KLOAD command does not erase previous contents of the table; the translation codes are added to any existing translations in the table. (The table can be cleared using the KCLR command, described below.) When a key is assigned a new translation keycode, however, the previous value is replaced.

The table area is limited to 511 bytes, so the file itself must not include more than this number of keycodes. See the preceding section on the KCODE command for a description of how each translation entry uses table space.

The format of the KLOAD command is:

KLOAD <d:filename.ext>

EXAMPLE: To load a translation list TRANSLST.TXT. It is stored on Drive A:

```
KLOAD A:TRANSLST.TXT
```

The translation file should contain KCODE commands "stripped" of the "KCODE" command word and of delimiters. For example, this KCODE command entered in the CP/M-86 command mode—

```
KCODE "HELP  H  E  L  P"
```

—should appear in our sample translation file, TRANSLST.TXT, this way:

```
HELP  H  E  L  P
```

When creating the file (using MEMORITE or SCOPE), type each redefinition on a separate line.

EXAMPLE: Suppose our TRANSLST.TXT file contains the five redefinitions shown in the examples above. The contents of the file look like this:

```
F1       L  I  S  T   RETURN

HELP     H  E  L  P   RETURN

F6       C  :  RETURN  U  S  E  R  SPACE  1  RETURN

F10      (A)  C  S  RETURN  N  RETURN  Y  RETURN  ESC

[UP]     ØCF
```

See the **SCOPE User's Guide** and **MEMORITE IV Primer** for information on how to use SCOPE and MEMORITE. See Chapter 5 of the **User's Guide to CP/M-86** for details on use of ED.

The KCLR Command

The KCLR command clears the keyboard translation table. When the table has been cleared, no conversions are defined and none are made.

The KMODE Command

The KMODE command sets the keyboard to one of several different operating modes. The command format is:

```
KMODE <mode>
```

The modes are:

0 = Physical mode
1 = Physical mode, foreign translation
2 = Vector 3-compatible mode
3 = Vector 3-compatible mode, foreign translation

The KCLICK Command

KCLICK turns on and off the "click" sound produced when each key is struck. A mode of "1" enables the clicker, "0" disables it.

EXAMPLE: To enable clicking:

```
KCLICK 1
```

THE VIDEO DRIVER COMMANDS

CONFIG's video driver commands modify the program that controls the display of characters on the screen. The video-driver program, together with the keyboard-related instructions described below, acts as a kind of "middle-man" between the keyboard and the screen itself.

The VMODE (Video Mode) Command

The video driver for the Vector 4 has two modes, one for application programs designed for earlier Vector systems (such as the Vector 3), and one for more recent software packages. The VMODE command allows you to select one of these modes (the two modes are simply different ways of handling the video functions).

FORMAT:

```
VMODE <mode>
```

For <mode>, you may enter

Ø = Vector-3 compatible video commands
1 = ANSI standard subset

By default, the operating system is loaded with the video driver in the ANSI mode, Mode 1. Mode Ø is what is needed for most programs in Vector Graphic's Professional Series. If you wish to use your system with any of these software packages, use VMODE to set the video driver to Mode Ø.

EXAMPLE: To set the video driver to the Vector 3-compatible mode:

```
VMODE Ø
```

EXAMPLE: To use an application program written for use on the Vector 4:

```
VMODE 1
```

"ANSI standard subset," incidentally, means that this mode uses commands chosen by a committee of professionals from the data processing industry (ANSI stands for American National Standards Institute). The Vector-compatible commands, in contrast, are Vector's custom approach. This is why you must specify the mode, or set of commands to be used, when running software packages based on one or the other approach.

The VCHAR (Video Character Set) Command

The VCHAR command is a follow-up to the VMODE command. VCHAR indicates which character set is in use. The available options are:

Mode 0 = Standard character set (the standard alphabetic and numeric characters and puncuation symbols, plus "block" graphics characters)

Mode 1 = Alternate character set (the same alphabetic and numeric characters and symbols as above, but includes "thin-line" graphics characters instead of block graphics characters)

Mode 2 = Alternate character set, foreign translation (the same alphabetic and numeric characters and symbols as in Mode 1, but instead of graphics characters includes foreign letters and symbols)

(NOTE: Foreign character sets for Mode 2 are available only in foreign-configured systems.)

Example: To set the video-driver program to use the standard character set:

```
VCHAR Ø
```

Example: To set the program to use an alternate, foreign-translation character set:

```
VCHAR 2
```

The VINIT (Video Initialization) Command

The VINIT command resets the video program to its "start" state. All internal tables and modes are set back to default, including the keyboard translation table. There are no parts of the command other than the command word itself.

Format:

VINIT

Example: To initialize the video program:

VINIT

The VTIME Command

VTIME displays the time on the system status line. The time is continuously updated on the screen and may be shown either in normal video (white-on-black) or reverse video (black-on-white).

To set the time shown by VTIME, use the CP/M-86 TOD command (see Section X, "The Transient Commands").

FORMAT:

VTIME <mode>

When <mode> is "Ø", the time display is in regular video. When it is "1", the time display is in reverse video.

EXAMPLE: To display the time in normal video:

VTIME Ø

MISCELLANEOUS COMMANDS

Five CONFIG commands aid in use of the CONFIG program itself. These are:

Command	Function
LIST	Displays all current configuration settings
STATUS	Displays configuration settings only as set by recent CONFIG commands
DMAP	Displays the logical mapping, or assignments, for the disk drives
REMARK	Displays a comment line ("for information only") on the screen during execution of program instructions
CHAIN	Transfers control, at the end of CONFIG's duties, to an application program

The LIST Command

The LIST command shows the current configuration for the printer, keyboard, video, disk, and communications drivers at the end of execution of CONFIG. It shows in detail the set-up you have specified for the system.

FORMAT:

LIST

EXAMPLE:

LIST

A typical display might look like this:

```
PR 1: SERIAL        TYPE=0233 PSIZE=060 LINES=040 MARGIN=10 AUTO=E BAUD=1200
PR 2: PARALLEL      TYPE=0122 PSIZE=060 LINES=050 MARGIN=05 AUTO=E
COMM: BASIC         BAUD=4800 PARITY=ODD WORDSIZE=8 STOPBITS=1
DISK: DSOFT=0 DRETRY=1 DCHECK=1 DVERIFY=1 DPROMPT=0
KEYBOARD=COMPATIBLE  VIDEO=STD CHAR, ANSI  LIST=PRINTER1
KEYBOARD REDEFINITIONS DONE=09  TABLE SIZE=511  REMAINING SPACE=460
```

Notice that the entries on each line correspond to the CONFIG commands for each category: printer, communications, disk, keyboard, and video.

The LIST command gives a complete display of your system's operating set-up. You also can bring up a partial display that reflects only the effect of CONFIG commands entered since the system was booted, using the STATUS command, as explained below.

The STATUS Command

STATUS does the job of LIST but in a more selective way. It shows your system's set-up—its status—only in terms of the most recently-executed CONFIG commands. LIST, in contrast, shows the system's configuration for every possible device and setting. Since some of these might not be important in your use of the system, STATUS lets you look at only the settings you have made or modified yourself.

STATUS displays the line or lines from the LIST display that describe the device affected by your CONFIG commands.

EXAMPLE: You enter this CONFIG command—

```
PLOAD 2 A:PARALLEL.DEV
```

—then enter this command:

```
STATUS
```

The result is a display of only the line from the LIST display for "PR 2":

```
PR 2: PARALLEL    TYPE=0122 PSIZE=060 LINES=050 MARGIN=05 AUTO=E
```

EXAMPLE: You enter these CONFIG commands—

```
PLOAD A A:SERIAL.DEV
PSIZE 1 90
PLINES 1 70
PMARGIN 1 10
PAUTO 1 1
KCODE "HELP  H  E  L  P"
KCODE "F3  C  O  N  F  I  G"
```

—then enter this:

```
STATUS
```

STATUS gives you the LIST lines for each affected device:

```
PR 1: SERIAL TYPE=0233 PSIZE=090 LINES=070 MARGIN=10 AUTO=E
KEYBOARD REDEFINITIONS DONE=02 TABLE SIZE=511 REMAINING
SPACE=497
```

The DMAP Command

DMAP displays the "logical mapping" of the physical disk drives. For example:

```
A: HARD DRIVE  B: HARD DRIVE  C: HARD DRIVE  D: FLOPPY SS
E: HARD DRIVE  F: FLOPPY DS  G: HARD DRIVE
```

This "map" is typical for a system with a Winchester hard disk and a floppy drive. As you can see, though there are only two physical drives, there are seven logical drives. This is possible partly because the Winchester disk can "divided" into two or more drives, at least in terms of how the operating system regards it (this is what is meant by "logical"). It's also possible because the floppy drive can be used in different ways: when its logical name is "C", it is operated as a drive for double-sided diskettes ("DS"). When its logical name is "D", it takes single-sided diskettes ("SS").

For the Winchester drive, different logical names tell the operating system to use separate areas of the disk. For the diskette drive, different logical names tell the operating system to use different instructions to run the drive.

Format:

```
DMAP
```

Example: To display the logical mapping of your system's drives:

```
DMAP
```

The REMARK Command

The REMARK gives the user a way to print comments during the execution of CONFIG commands. REMARK comments are placed in your CONFIG file and appear on the screen as CONFIG commands are processed. For example, this comment—

```
THE DRAFT PRINTER HAS BEEN SELECTED FOR USE
```

—could be used to show which printer driver program had been selected by the PSEL command. It would be produced by putting this command in the CONFIG file after the PSEL command itself:

```
REMARK "THE DRAFT PRINTER HAS BEEN SELECTED FOR USE"
```

The format for REMARK is:

```
REMARK "message"
```

The message must be enclosed in delimiters, as usual for this type of CONFIG command.

The CHAIN Command

The CHAIN command instructs the operating system to move directly to a program or application program when it has finished with CONFIG commands.

FORMAT:

```
CHAIN "<d:program name> <parameters>"
```

EXAMPLE: To CHAIN to the Memorite word processing program (running it through the CP/M Simulator, RUN8):

```
CHAIN "RUN8 MEMORITE".
```

In this example, "RUN8" is the program name, "MEMORITE" is a parameter.

EXAMPLE: To CHAIN to the CP+ program, when it is stored on Drive B:

```
CHAIN "B:CP+"
```

Creating and Editing CONFIG Batch Files

Although CONFIG commands for your system could be entered one by one each day, a better way is to let them execute automatically from a command file. As already noted, the booting program is programmed to do this: it searches for a CONFIG file with a .BEG extension and performs the configuration instructions the file contains. You also can create other CONFIG files and send a selected batch file to processing from the operating system command mode.

You can edit the .BEG file and create other CONFIG batch files using SCOPE, MEMORITE[1],or the CP/M-86 text editor, ED. Although the system's CONFIG file must always have the .BEG extension, your own CONFIG file or files can have any names you wish and usually have an extension of .TXT (if created in SCOPE or ED) or .MEM (if created in MEMORITE).

Some examples of typical system configurations and CONFIG files are shown below.

Sample CONFIG Batch Files

The most common need for CONFIG commands is in setting up and selecting a printer for use.

1. To set up and select a serial printer as Device 1, use these commands:

```
PLOAD 1 SERIAL.DEV
PSEL 1
```

2. To set up both a serial and a parallel printer and select the parallel printer for use, use these commands:

```
PLOAD 1 SERIAL.DEV
PLOAD 2 PARALLEL.DEV
PSEL 2
```

3. For a more user-oriented status output, you could omit the LIST function and include your own remarks in the command set, as shown below. In this batch, the printer driver for the Vector 7700 is loaded instead of the standard parallel driver. The last command CHAINs to a display of the drive directory (through the DIR command).

```
PLOAD 1 SERIAL.DEV
PLOAD 2 NEC.DEV
PSEL 2
REMARK "EPSON PRINTER AVAILABLE FOR USE"
REMARK "NEC PRINTER AVAILABLE FOR USE"
REMARK "NEC PRINTER CURRENTLY SELECTED"
CHAIN "DIR"
```

[1] NOTE: Normally, Memorite puts a special header on a file which should not be present in the batch CONFIG file. To write your created file to disk with the header omitted, use the "WD R" ("Write to Disk Raw") command.

SECTION X

CP/M-86 TRANSIENT COMMANDS

What Is a Transient Command?

Transient Commands do commonly-needed jobs of system housekeeping, such as moving files from place to place, collecting and displaying information about the status of your disks, or copying the contents of your hard disk to backup diskettes.

The instructions that make up each transient command are stored in a command file. Because the transient programs are fairly long, these files are kept on disk even after the system is booted. They are brought into the Vector 4's internal memory (where the rest of CP/M-86 was placed during booting) only when called for use. You can identify command files in the directory listing by their .CMD extensions.

All transient commands are called in the same way. Type the command name (omit the .CMD extension), any command lines or options used by the program, and **[RETURN]**. CP/M-86 looks for the program on your current drive and user area. If you've typed your input correctly, the screen first displays the command name and version number, then the program goes to work.

If your input isn't right in some way, the screen displays your entry followed by a question mark. When this happens, first check your input for right spelling and syntax. Then make sure the command file you've called is in your current drive and user area. These are shown in your screen prompt—for example, **AØ>** or **B12>**.

Use the **DIR** command to check your directory: Type:

DIR <command name>

If the system responds—

FILE NOT FOUND

—log onto User Area Ø and repeat the **DIR** command.

When you try to run a transient command, CP/M-86 looks for the command file first in the user area shown in your system prompt. If it doesn't find it there, it goes to User Area Ø and looks for a "SYS" (System) version of the file. Originally, all transient commands are stored in User Area Ø of Drive A. If you're working in User Area 12 and want to use a transient command, a copy of the program must be in that user area, or you must have set the program in User Area Ø to SYS status, using the STAT command.

You could copy a frequently-used command file to any other drive and user area (using PIP), and CP/M-86 could then run the program from there. But current user area and in User Area Ø, where copies of all transient commands can always be found (the command file still must be on your current drive).

SYS status also keeps transient commands from showing up in (and cluttering up) the directory listings for your user areas. Normally, you will want to see only your data files. It is recommended that any often-used command file be set to SYS status. See "The STAT Command" below.

Kinds of Transient Commands

There are three kinds of transient commands: utilities, tests, and programming commands. Utilities help with disk and file management, and you will use them frequently. Tests and programming commands are used for program development, most often in advanced programming functions. They will not be discussed in this manual. (See the optional **CP/M-86 Programmer's Guide.**)

Usually, transient commands are used with options that let you specify more exactly what you want the program to do. These options are mentioned in the "Overview" below. In the following section, "Procedures for Using CP/M-86 Transient Commands," they are described in full. To get the most out of each command, make sure you know what options can be used.

One of the commands described below, RUN8, has a special purpose. RUN8 brings into use the CP/M Simulator, a program that translates software written for Vector 4 CP/M and allows it to be run under CP/M-86.

Overview Of CP/M-86 Transient Commands

HELP: Provides brief information on the use of CP/M-86 commands and options. Entered with no command tail, lists the available topics. Entered with a command tail (the name of a command), gives details about the use of the specified command.

STAT: Lists the number of bytes of storage space remaining on the specified disk or used by the specified file, and can display or alter disk, file, or device status.

PIP: Copies programs and text files between disks and user areas. Can also be used to transfer files to or from logical devices, such as printers or communication lines.

STORE: Used for backing up a hard disk to a floppy if there is more information on the hard disk than can be held by a single floppy. Automatically keeps track when files are continued across two or more diskettes.

RESTORE: Returns to the hard disk files put on diskette by the STORE program. (Typically, you would use STORE to create backup copies of hard disk information, then use RESTORE to copy from the backups to the main disk.)

TOD (Time of Day): Lets you examine and set the system time-of-day clock.

RUN8 (CP/M Simulator): Allows you to run Vector 4 CP/M programs and packages under CP/M-86.

How to Use CP/M-86 Transient Commands

Like the resident commands, all transient commands are called by entering the name of the command, a command tail (any options or information that the commands require), and **[RETURN]**. Always be sure to read the explanation of each command (either here or in the Appendix, "Diskette Directories") before entering a command. Some programs may cause unexpected results if used without understanding of all options.

The HELP Command

The HELP command provides a quick, on-screen guide to the use of CP/M-86 built-in and transient commands. Because it can be entered in two ways—either for a display of all available topics (commands and options) or for information about a specific command—it is useful both for those just becoming familiar with CP/M-86 and for the most experienced user who needs only a "refresher" on a seldom-used command.

FORMAT:

A>**HELP [command name] [command options]**

Entered by itself, HELP displays a list available topics. At the end of the list is the special prompt, **HELP>**. This allows you to ask for additional information on a topic by entering its name.

Entered with a command name, HELP gives information on how to use the command.

EXAMPLE:

A>HELP [RETURN]

This command displays the complete list of topics.

EXAMPLE:

```
A>HELP STAT [RETURN]
```

The program displays information about the use of the STAT command.

EXAMPLE:

```
A>HELP STAT OPTIONS [RETURN]
```

Added information shows how to use the options available with the STAT command.

The STAT Command

The STAT program can be used in two ways:

- o to list statistical information about files or disks ("passive" use);
- o to change the status of disks, devices, or files ("active" use).

STAT can be entered as a one-word command or together with command options. Most of the uses with options fall into the "active" category, because they allow you to change some feature of the system's operation.

Using "Passive" STAT Commands

Checking Size And Storage

You can use the STAT program to find the space remaining on a disk or diskette. Enter the command:

```
A>STAT [RETURN]
```

The screen displays:

```
disk: <ATT>, Free Space: ###k
```

for each disk that has been used since the last boot. The <ATT> column lists the attributes of the entire disk.

You can also enter the command:

```
A>STAT d:[RETURN]
```

replacing **d:** with a drive letter (for example, A:).

This brings the display:

```
Bytes remaining on d: ###k
```

This form of the command displays information only for the disk requested,whereas the **STAT [RETURN]** command (without a drive letter) shows information for all drives which have been accessed since the last cold boot.

This form of the STAT command is used most frequently to check how much space remains on a floppy diskette, to determine whether or not you can fit more files onto it.

You can use STAT to ask the current user area. Enter:

```
STAT USR:
```

The screen displays:

```
Active User : #
Active Files: # # # # #
```

where the # signs represent user areas that contain files.

Checking Files

The typical command format for using STAT to check files is:

```
A>STAT <Ambiguous Filename> [RETURN]
```

Example:

To use STAT to check the size and attributes of a file named SAMPLE.DOC. Enter:

```
A>STAT d:SAMPLE.DOC
```

This will bring the following display:

```
 Recs  Bytes  Ext Acc
      ###    ###k    # xxx d:SAMPLE.DOC
Bytes remaining on d: ###k
A>
```

To check the size and attributes of several files with the .DOC extension, enter:

```
STAT d:*.DOC
```

This produces a display like the one above but for each file having the .DOC extension.

In these displays, the "Recs" column lists the number of records (128 bytes each) making up the file. "Bytes" shows the number of bytes. You would expect this second number to be equal to the number of records times 128, but it usually is greater because it reflects the total number of bytes which could be stored in the space CP/M-86 has allocated to the file.

The last column in the STAT display, marked "Acc", stands for the access attributes of the file. CP/M-86 defines four attributes, which can be assigned to a file individually or in limited combination. These attributes are:

R/O: Read-Only status; file cannot be changed or deleted until this attribute is reset by the STAT command.

R/W: Read-Write status; file can be changed or deleted.

SYS: Changes file to System type; file does not appear on screen when DIR command is given (does appear in the DIRS command display). The SYS attribute protects files from unauthorized use. SYS files cannot be copied through PIP.

DIR: Removes SYS attribute so that file again appears in Directory.

Files cannot be both Read-Only and Read-Write at the same time, nor can they be both SYStem and DIRectory. But any other combination is permitted—for example, Read-Only and Directory, or Read-Write and System.

A typical status line for the file SAMPLE.DOC might look like this:

```
   Recs  Bytes  Ext Acc
    27     8k    1 R/W A:SAMPLE.DOC
Bytes remaining on d: 422k
A>
```

This would mean that the file SAMPLE.DOC, stored on Drive A, consists of 27 records of 128 bytes each, is allocated 8 kilobytes (about 8000 bytes) of the disk surface. It uses one extent, and is available for read and write activity. In addition, the total unused space on the disk in Drive A is 422 kilobytes.

Running An "Active" STAT

An "active" STAT command is one which changes in some way the status of a file, disk, or device. To see a list of the active STAT commands, enter **STAT VAL: [RETURN]**. The screen displays:

```
Temp R/O Disk: d:=R/O
Set Indicator: d:filename.ext R/O R/W SYS DIR
Disk Status  : DSK: d:DSK:
User Status  : USR:
IObyte Assign: <possible assignments>
```

This probably doesn't help much right now, so here is explanation of what the more useful functions mean and how to use them.

Changing File Attributes

To change the attributes of a file, enter the command:

```
A>STAT d:<Afn> <attribute> [RETURN]
```

where <attribute> can be any of those mentioned above (R/O, R/W, DIR, or SYS).
You can use the STAT program for files stored on another disk by entering the drive letter for **d:** in the line above. Otherwise, the program acts on files on the current disk or diskette.

The STAT command can be used to protect a file from being changed or erased accidentally. For example, to protect the file SAMPLE.DOC (stored on the B drive), enter the command:

```
A>STAT B:SAMPLE.DOC R/O [RETURN]
```

The screen displays the message:

```
SAMPLE.DOC set to R/O
A>
```

Changing Disk Attributes

To protect an entire disk from changes, enter:

```
A>STAT d:=R/O [RETURN]
```

substituting the drive letter in the location of **d:** in the example. This sets the whole disk to Read-Only, just as a write-protect tab keeps a floppy diskette safe from changes. This form of STAT can be used for either floppy or hard disk.

The default status of any disk (its status if you set no attribute yourself) is is R/W, since this is the common use of disk memory. The R/O status is available for protecting a disk or diskette from unwanted changes. While it is in effect, it makes it impossible to copy information onto the disk, to erase files or parts of files from the disk, or to add information to individual files on the disk.

But its effect is not permanent. You can change the attribute later if you wish. To remove the R/O status, enter **[CTRL C]**. The STAT d:=R/O command protects a disk only until the next warm or cold boot.

Setting a File to SYS Status

When you call any transient command for use, CP/M-86 looks for the command file only on your current drive and user area. But the most commonly-needed transient commands are stored in User Area Ø of Drive A. If you're working in a different user area—as you frequently will—you don't want to have to go to User Area Ø every time you need a transient command. The SYS function of STAT lets you change command files to "System" status, and this causes CP/M-86 to check for a requested program both in the current user area and in User Area Ø, where it always can be found.

EXAMPLE: To set the PIP transient command to SYS status:

```
A>STAT PIP.CMD SYS [RETURN]
```

The screen displays:

```
PIP.CMD set to SYS
```

The PIP program is now accessible from any user area of Drive A. As always, if you are working on a different drive, you can still use the program by prefixing "PIP" with the letter of the drive where it is stored.

EXAMPLE:

```
B>STAT A:PIP.CMD SYS [RETURN]
```

It is recommended that you use STAT SYS for each .COM and .CMD file in your system. You must set one of your .CMD files, the RUN8 program, to SYS for it to work (see "The RUN8 Command" below).

Other Functions of the STAT Program

As shown by the STAT VAL: display, the STAT program has several other uses. Most other STAT commands, however, change input and output devices, such as tape readers, batch processors, teletypes, paper-tape punch devices, or non-standard list devices. Briefly, they are:

Enter **STAT VAL: [RETURN]** for a list of the valid device names (listed under **IObyte Assign**) supported on your specific system.

Enter **STAT DSK: [RETURN]** to find out the specific status and capacity of the disk, in records, bytes, extents, et cetera.

Changing the status of other devices will not be covered in this manual, since it requires more experience with the system. For more information, see the **Vector 4 CP/M Programmer's Guide.**

Review

The STAT program can be used as a single command (**STAT [RETURN]**) to show remaining disk space. Or it can be used with the following command options:

TO FIND/CHANGE	COMMAND LINE AND COMMENTS
Size of a file	**STAT d:SAMPLE.DOC [RETURN]** (Lists number of actual records per extent under "Recs" column)
	STAT d:SAMPLE.DOC S (Lists actual file size in records under "Size" column)
Change file/disk attributes	**STAT d:<SAMPLE.DOC> R/W** (Makes file or disk available for editing or other changes)

Change file/disk attributes

STAT d:<SAMPLE.DOC> R/O
(Protects the file or disk from changes)

STAT d:SAMPLE.DOC SYS
(Changes the file to a System file)

STAT d:SAMPLE.DOC DIR
(Changes the file back to non-System status)

The PIP Command

PIP stands for "Peripheral Interchange Program". The PIP command copies programs and text files between disks. It also can be used to transfer files to or from logical devices, such as printers or communication lines (the devices mentioned above in the description of the STAT command), and to combine or rename files (either in copying them to another disk or to the same disk).

PIP is an extremely flexibly command and has various formats. The basic command line format is:

```
A>PIP <target>=<source> [RETURN]
```

where **<target>** and **<source>** can be disks, files (ambiguous or unambiguous), or logical devices. As with the STAT command, this manual will cover use of the command only for disks and files. If you plan to use other devices, refer to the manual for that device for instructions on how to move information to and from it.

If you try to PIP to a read-only (write-protected) disk, the screen flashes the message, **ERROR - DISK IS READ-ONLY.** If this happens, make sure there is no write-protect tab on the diskette. If the write-protect notch is clear, remember that you must enter a **[CTRL C]** after inserting a new diskette in the drive. (Newly-inserted diskettes are Read Only by default until you press **CTRL C]** to do a warm boot.) If the disk is still considered read-only, see the section on STAT to find out how to open the diskette for writing.

The PIP.CMD file is on User Area Ø of your disk, and you must use the program while logged into that user area. As we will see, the [G#] option allows you to copy files between all other user areas.

The One-Line PIP Command

You probably will use the PIP command more than any other CP/M utility. Because PIP allows you the option of using ambiguous file names for your source, you can copy a large amount of information with a single command.

As an example, let's say you want to make a copy of everything in the current user area of the A drive. To copy it to the B drive, you need only enter:

```
A>PIP B:=A:*.* [RETURN]
```

The screen displays:

```
COPYING -
```

then shows each file's name as it is copied to the target drive. The PIP command keeps the file names from the source as it moves information to the target. In this example, it changes only the drive letter. All the transient commands on the A drive, for example, will have the same command names on the B drive.

To move one file from A: to B: (in the same user area), enter:

```
A>PIP B:=A:SAMPLE.DOC [RETURN]
```

The next time you enter DIR on the B drive, you will see SAMPLE.DOC in the directory.

You can use PIP to make an extra copy of a file, renaming it as it is copied. For example, to make a spare copy of the file **SAMPLE.DOC**, changing the extension to **.OLD**, enter:

```
A>PIP SAMPLE.OLD=SAMPLE.DOC [RETURN]
```

On which drive would the .OLD file be recorded? The same one, drive A:, because you didn't specify another. The program assumes you are copying within the same surface.

All of these possibilities can be used in combination. The full PIP command line (with all available options) is as follows:

```
A>PIP <disk:><target.ext>[G#]=<disk:><source.ext><[G#]><[V]>
```

The [V] option (for "Verify") tells the program to check that it has made an accurate copy. This check slows the program somewhat, but it can be important for those "better-safe-than-sorry" operations (such as copying a backup diskette).

The [G#] option works as a "locator" and can mean either "Go to" or "Get from." When placed after the target filename, it tells the BDOS to place the copied file in a specified user area (# is replaced by the number of the user area). When placed after the source filename, [G] tells the BDOS to find the file in the named user area. In copying between user areas, then, PIP can either "Go to" an area with the copy or "Get it from" another, or both.

EXAMPLE:

You are working in User 5 of the A drive, and you want to copy to a floppy diskette in the C drive for backup. First insert a formatted diskette in Drive C. Next, to access the PIP program, log onto User Area Ø. You have the choice of copying to the same User Area (5) or to some other area.

a. To copy to User Area 5 on the C drive, enter:

```
A>PIP C:SAMPLE.DOC[G5]=A:SAMPLE.DOC[G5] [RETURN]
```

The PIP program copies the file to the user area specified by the first [G#] locator and gets it from the user area named by the second [G#] locator. When the A> prompt reappears, the file has been copied to User Area 5 on Drive C.

b. To copy to a different user area—say, User Area Ø—enter:

```
A>PIP C:SAMPLE.DOC[GØ]=A:SAMPLE.DOC[G5] [RETURN]
```

If there is a copy of PIP.CMD on Drive C, you could get the PIP program from that drive, even though you're logged onto Drive A:

```
A>C:PIP C:SAMPLE.DOC[GØ]=A:SAMPLE.DOC[G5] [RETURN]
```

With CP/M-86, you can run programs stored on drives different from the one shown in your screen prompt (as we did above) if you prefix the letter of the other drive to the program name.

The Repeat PIP Command

If you plan to move many documents to different places during one session, you can do this using the "Repeat" version of PIP. Enter **PIP [RETURN]**. (Note that you cannot enter a space after PIP: the screen will display the error message **INVALID FORMAT:**). This puts PIP in the "repeat" mode. After a few moments, an asterisk prompt (*) appears on the screen to tell you that the PIP program is ready for a command. Enter any valid PIP command line at this time and type [RETURN]. When copying for this file is done, the asterisk prompt reappears. You now can type another PIP command line.

Valid PIP Command Lines

PIP accepts the following command lines, entered either after the PIP command word itself or separately in the repeat mode (following the * prompt). Wherever an ambiguous file name (afn) is indicated, you can substitute an unambiguous filename (ufn) if you wish. Unless you specify another user area with the [G#] option, the system defaults to the current user area for the source and target of the copied file.

B:=A:<afn>

Copies **to** Drive B **from** Drive A all files matching the ambiguous file name.

B:=<afn>

Copies **to** Drive B **from** the drive you are currently addressing all files matching the ambiguous file name.

B:=A:<ufn>

Copies **to** Drive B **from** Drive A the specific file named.

B:=<ufn>

Copies **to** Drive B the specific file entered.

B:=A:<afn/ufn>[G5]

Copies **to** the current user area on Drive B from Drive A, User Area 5, a file named by an ambiguous or unambiguous file name. The file name of the copy is unchanged.

B:<afn/ufn>[G2]=A:<afn/ufn>

Copies to user area 2 on Drive B the file named by an ambiguous or unambiguous file name.

<ufn2>=<ufn1>

Copies within the same drive and area the specified file <ufn1> and renames it at the same time <ufn2>. The source file <ufn1> still exists.

Use of ambiguous file names and wild cards can save a lot of time, as the examples above show. Remember these rules:

- o When copying from one drive or user area to another, you don'tneed to specify a target file name unless you plan to rename the file.

- o When copying in the same drive and area, the target name must be different from the source name. CP/M-86 does not allow two files with the same filenames and extensions to be stored in the same area of a drive. You can copy two files with identical file names (it could be copies of the same file) to two different user areas on the same disk.

- o You don't need to specify the source disk name unless it's different from the disk you're logged into.

- o When copying between user areas, use the [G#] option to specify either the target area or source area or both.

- o When moving files from one disk to another, either the target file name or target disk must be given. For example, if you're logged on drive A, you cannot enter **PIP =B:*.CMD** to move all .CMD files from B to A. You must type **PIP A:=B:*.CMD.** The target must be explicit. You could, however, log onto the **source** drive (in this example, B) and then type **PIP A:=*.CMD.**

- o The **target** file or drive name comes first, the source name second.

- o Look at your source files for patterns that allow you to use wildcards. To copy five files all beginning with TR (for example, TRANSCOM.MEM, TROUBLE.CMD, TRIP.DOC, TRAINING.ASM, and TRANSLAT.BAS), you could enter one command line:

  PIP <target disk>=<source disk>TR*.* [RETURN].

A final note on the PIP program: You cannot PIP any files set to **SYS** status by the STAT program, or files which have an extension of DBS (for database). You have two alternatives: you either can use STAT to set the SYS files to DIR, or use the DISKCOPY or STORE programs to move the entire contents of the disk. (See below for information STORE/RESTORE. See Section IX, "The UTILITY Program," for details about use of DISKCOPY.)

The STORE Command

The STORE command is used to back up files from the hard disk to diskettes. (If your system only has floppy drives, you can ignore this section, as well as the section on RESTORE.)

The STORE and RESTORE programs can be used only with files having names of seven characters or less. CP/M-86 allows eight characters per filename, but STORE takes one of these spaces for its own use. It adds a number for use in managing long files that are continued across two or more diskettes. This number shows which portion of the file belongs on diskette Ø, 1, 2, and so on.

If you plan to do regular backups of your hard disk onto diskettes (and a regular backup schedule is a good idea), you will use the STORE/RESTORE programs for the job and, looking ahead, should assign only seven-letter names to files. (You can use the RENAME command to change eight-letter file names if necessary.)

The command line for STORE and RESTORE is like that of the PIP program, including all references to disks, user areas, and ufn/afn specification.

EXAMPLE:

To copy all files from the hard disk A to floppy drive C, log onto the A drive, then enter:

```
A>STORE C:=*.*
```

This copies all files with names of seven letters or less. Files with eight-letter filenames are skipped.

Here's what happens when you enter the command:

1. STORE copies the files, adding the character **$** to the end of each filename.

   EXAMPLE:

   PROGRAM.CMD is copied to the diskette as PROGRAM$.CMD.

2. If the target diskette fills up in a middle of copying the file, the **$** is changed to **Ø**. Then the program prompts you to enter a new diskette.

   EXAMPLE:

   The first part of PROGRAM$.CMD is stored on the first diskette as PROGRAMØ.CMD.

3. The file that was half-copied is continued on the next diskette. When the end of the file is reached, the program adds **1** to the filename to indicate that it comes after the **Ø** file.

EXAMPLE:

The rest of PROGRAM.CMD is copied to the backup diskette as PROGRAM1.CMD.

4. When the last section of a file has been copied, STORE makes the last character of the filename **$.**

   EXAMPLE:

   The diskette storing the last section of PROGRAM.CMD stores the file as PROGRAM$.CMD. This tells the RESTORE program that this is the end of the file.

5. This process could continue until all 5 megabytes have been copied. If a single file (a data base or a huge program) takes up more than 2 floppies, for example, STORE would continue to add numbers (from Ø, through 1, 2, 3, and so on). Nine floppy diskettes can handle the largest file possible under the 5-megabyte system.

   EXAMPLE:

   Should PROGRAM.CMD take up over 1900 kilobytes, it would be stored as:

DISKETTE	**FILE NAME**
1	PROGRAMØ.CMD
2	PROGRAM1.CMD
3	PROGRAM2.CMD
4	PROGRAM$.CMD (final section)

Be careful to number your diskettes. When you RESTORE their contents later, you must insert the diskettes in the correct order for the file to be re-copied correctly.

The RESTORE Command

The RESTORE program returns to the hard disk files put on diskettes by the STORE program. It is strictly a companion command to STORE: you cannot use RESTORE to move files if they were put on diskette by the DISKCOPY or PIP programs. Everything described above for STORE holds true for RESTORE except that the action is in reverse.

Be sure to enter your backup diskettes in the correct order. Once RESTORE has begun copying files, it asks you for each diskette as it needs it.

EXAMPLE:

To RESTORE all files copied from Drive A of the hard disk to diskettes in Drive C (the diskette drive):

```
A>RESTORE A:=C:*.* [RETURN]
```

EXAMPLE:

To RESTORE the file PROGRAM.CMD to your hard disk, insert the first of your diskettes (the files on this diskette have had "Ø" added) and enter:

```
A>RESTORE A:=C:PROGRAMØ.CMD [RETURN]
```

This begins the restoration of the file that was stored, in our earlier example, across four diskettes. Note that the filename in your command now must include the "Ø" that STORE added when copying the file.

When the program has copied the last file on the directory of the first diskette, the screen displays:

INSERT NEXT DISK TO CONTINUE OR [CTRL C] TO EXIT

This is the only point in the process at which you can stop the program without resetting the entire system.

If the hard disk is full or out of directory space, the program stops automatically. One option available with the RESTORE program (and nowhere else) is the **K** (for "Kill") option. Use this cautiously: it kills (deletes) the file before it is copied to the hard disk.

Normally, the program goes through this procedure:

- The file in process is copied with an extension of **$$$**, meaning that it is temporary.
- Once copying of the file is done, the original output file is "killed."
- The temporary **$$$** file is renamed to the proper extension.

Use the **Kill** option only if:

- The file you want to copy already exists on the hard disk.
- There is not enough room on the hard disk for the file.
- You cannot make enough room on the hard disk for the file.

The TOD (Time of Day) Command

Your CP/M-86 system has an internal clock that runs any time the system is turned on. The TOD command allows you to set this clock and to check its current setting. When you first receive your system, the clock is set to the time in which one part of CP/M-86, the BDOS, was created. You can use TOD to change this setting to the correct time and date.

EXAMPLE:

To see the current time and date setting:

```
A>TOD [RETURN]
```

TOD gives a display like this:

```
06/01/83      09:15:50
```

The clock follows twenty-four-hour time, so this display shows the current setting to be 9:15 and 50 seconds AM on June 1, 1983.

EXAMPLE:

To set the clock:

```
A>TOD 07/01/83 08:31:00 [RETURN]
```

This sets the clock to 8:31 AM, July 1, 1983.

The clock is reset to ØØ:ØØ any time the system is turned off or reset.

You can have the system time displayed continuously on the screen by use of the VTIME command. See "The VTIME Command" in the previous section on the CONFIG Utility.

Use of TOD is optional. The system runs properly whether or not the clock's setting is correct.

The RUN8 Command

The RUN8 command is unlike other commands described in this section. It is a .CMD file that the system uses automatically, so you will never enter a "RUN8" command line. The RUN8 file allows programs written for Extended CP/M, Vector 4 CP/M, and other versions of CP/M (the eight-bit operating system) to be run under CP/M-86. RUN8 calls into action the CP/M Simulator, which stands in for CP/M, doing any operating system jobs different from those done by CP/M-86.

CP/M programs that run under the Simulator can be recognized by their .COM extensions. These might be command files designed for earlier Vector Graphic systems that used Extended CP/M or Vector 4 CP/M, or they might be any CP/M program that works in the conventional way. One set of Vector Graphic software, the Professional Series application packages, consists entirely of .COM files that rely on the RUN8 Simulator for executing in the CP/M-86 environment.

Although you will never actively call on RUN8, there are several things you should do to let it work most effectively:

1- As shipped, RUN8.CMD is stored on Drive A, User Area Ø, and it must stay there to work. If you wish, you can PIP copies of the file to other drives or user areas, but never erase the file from its original location.

2- To be used from other user areas on the drive, the RUN8 file should be set to SYS status through the STAT command:

   A>**STAT RUN8.CMD SYS [RETURN]**

3- Any .COM files—that is, eight-bit programs written for CP/M—also should be set to SYS status if you want to run them from user areas other than User Area Ø. (See "The STAT Command" in this section for more information of the SYS attribute. It is recommended that all .CMD and .COM files be set to SYS to make them easily accessible from anywhere in a drive.)

When the Simulator runs a program, you will have no sense that an intermediary is at work translating any of the program's functions. The program runs just as it would under Vector 4 CP/M. Operation of the Simulator is transparent. When you end work with a program, the Simulator returns you to the CP/M-86 command level.

Certain Vector 4 CP/M .COM files are not supported under the CP/M Simulator, either because they are replaced by newer programs or because they were uniquely suited to the earlier operating system. These are listed in Appendix C, "User's Guide to Troubleshooting the Vector 4."

SECTION XI

INSTALLING CP/M-86 ON YOUR HARD DISK

Two Operating Systems: Vector 4 CP/M and CP/M-86

Your Vector 4 system comes with two operating systems, Vector 4 CP/M and Vector 4 CP/M-86. Both are provided on System Diskettes. In addition, if your system includes a hard disk, the Vector 4 CP/M operating system already is installed on the hard disk and ready for use.

You can run your Vector 4 with either of the two operating systems. Vector 4 CP/M gives you access to all application programs developed for Vector systems. Vector 4 CP/M-86, when in use, provides the greater power of a sixteen-bit operating system and also takes over the functions of Vector 4 CP/M through its CP/M Simulator. Application programs designed to run under Vector 4 CP/M also will run under Vector 4 CP/M-86 using the CP/M Simulator.

On floppy-disk systems, you can use either operating system on your Vector 4. On Winchester-disk systems, once CP/M-86 is installed on the hard disk, you will use that operating system. Because the Simulator takes over for Vector 4 CP/M, you no longer need Vector 4 CP/M to run application programs designed for it. Vector 4 CP/M is, in effect, made a part of Vector 4 CP/M-86.

What About Use of Vector 4 CP/M?

The Vector 4 CP/M operating system is covered in the **Vector 4 USER'S MANUAL**, which you received with your system. On hard disk systems, it can be used up until the time that you convert the disk for use of CP/M-86. Then, the CP/M-86 Simulator comes into action and allows you to continue use of all Vector 4 CP/M programs.

How are Vector 4 CP/M and CP/M-86 Different?

If you are familiar with Vector 4 CP/M, you will find that CP/M-86 is very similar. Commands are similar and files are named and organized the same way. The most important differences between Vector 4 CP/M and CP/M-86 are invisible to you as a user.

How Do I Change Operating Systems?

How to change from Vector 4 CP/M to CP/M-86 differs depending on the type of your Vector 4 system—whether it has a hard disk or two diskette drives—and on whether your Vector 4 is brand new or has been in use under Vector 4 CP/M. Separate sections appear below for each situation.

- If your Vector 4 system is new and includes a hard disk, changing from Vector 4 CP/M to CP/M-86 means **installing** CP/M-86 on the hard disk. See "Installing CP/M-86 on Hard-Disk Systems."

- If your Vector 4 system is not new—if you have been using Vector 4 CP/M and have just obtained CP/M-86 from your dealer--changing from Vector 4 CP/M to CP/M-86 means **converting** your hard disk from one to the other. See "Converting to CP/M-86 on Hard-Disk Systems" below for steps.

- If your system has floppy-diskette drives (no hard disk), changing from Vector 4 CP/M to CP/M-86 means making two changes to the hardware of the computer. After that, you can boot the CP/M-86 operating system from its System Diskette just as you would boot Vector 4 CP/M. See "Using CP/M-86 on Diskette-Based Systems" below.

Installing CP/M-86 on Hard-Disk Systems

Installing CP/M-86 on the hard disk of new Vector 4 systems involves four steps. You must:

1- Insert a new Executive PROM (Rev. 2.0 or greater) and cut two signal traces on the main circuit card of your computer

2- Format the entire disk, using the FORMAT function of the UTILITY program

3- Install CP/M-86, using the SYSGEN function of UTILITY

4- Configure your system, using the CONFIG program

Here is how to do each step.

Step 1: Making PROM and Jumper Changes

Two changes to the Vector 4's hardware are part of the CP/M-86 conversion. You must:

1- Install the new Executive PROM provided with your upgrade kit, and

2- Cut two signal traces that control how data for the screen display is stored

These changes should be performed by your dealer. They are described in the **CP/M-86 Hardware Installation Instructions.** This sheet is included in this documentation package, after the end of this manual. When these changes have been made, go on to Step 2.

Step 2: Formatting the Disk

Having made the needed hardware changes, you are ready to format the disk for CP/M-86.

In Vector's Disk Partition System, the FORMAT function is part of the UTILITY program. FORMAT writes on the disk a "framework" to be used by the operating system. For CP/M-86 and the Disk Partition System, it also defines a "partition" for the operating system and the size of the operating system's file storage area.

See Section IX for complete information on using UTILITY and the FORMAT function.

1. Press the RESET button at the rear of the console. The system banner will appear on the screen.
2. Insert the Vector Graphic CP/M-86 System diskette in the diskette drive and close the drive door.
3. Press [F].

Allow a few moments for the system to boot. Then the system message is displayed.

4. Format the hard disk following the steps outlined in Section IX.

Step 3: Installing CP/M-86 on the Disk

The disk is now formatted and ready for installation of the CP/M-86 operating system. This is done through another function of the UTILITY program, the System Generation (SYSGEN) function.

1. Recall the UTILITY program by typing

   A>**UTILITY [RETURN]**

2. Type [1] to select the System Generation program.
3. Sysgen the CP/M-86 operating system to the hard disk following the steps outlined in Section IX.

Converting to CP/M-86 on Hard-Disk Systems

If you have been using your Vector 4 under Vector 4 CP/M and already have files on the hard disk, moving to use of CP/M-86 means **converting** the disk rather than installing. The difference is that you must save your existing files on backup diskettes before formatting the disk in preparation for installing CP/M-86.

Conversion takes six steps. You must:

1- Make backup copies of all programs and files currently on the hard disk, in preparation for new formatting (which will erase these files)

2- Install a new Executive PROM and cut two jumper wires that control how data for the screen display is stored

3- Format the entire disk, using the FORMAT function of the UTILITY program

4- Install CP/M-86, using the SYSGEN function of UTILITY

5- Copy files from the backup diskettes back to the hard disk

6- Configure your system, using the CONFIG program

If your existing files are large or many, conversion can take time, because you must copy all these files first to backup media, then back to the disk. Once CP/M-86 is installed, however, you will have use not only of all your existing CP/M programs and files (through the Simulator), but also of all CP/M-86 software—in both cases, enjoying the added power of CP/M-86. In addition, once in place, the Disk Partition System will allow use of other 16-bit operating systems as these are made available by Vector.

Here are the steps for converting your hard disk to CP/M-86 and the Disk Partition System.

Step 1: Backing Up Existing Files

To copy your files from the hard disk to backup diskettes, use the STORE/RESTORE commands.

STORE is specially designed for copying files from the hard disk to backup diskettes, and it is the only way to do this for files too large to fit on one diskette. Note, however, that STORE and RESTORE can be used only with files having names of seven characters or less. It may be necessary for you to rename some files before STORE/RESTORE will work for the complete backup.

Backup diskettes must be formatted. If you plan to use previously formatted diskettes, note that STORE erases any information already on a diskette. Use diskettes that you are sure hold no valuable information, or new, freshly formatted diskettes. See Section IX for details on using the FORMAT function.)

To save time in the backup process, use double-sided diskettes if possible (they store twice as much information as can single-sided diskettes).

Make sure you have enough diskettes to handle the capacity of your hard disk. A double-sided diskette holds about 600 kbytes (six-hundred thousand bytes) of data. Depending on your system, the hard disk holds either five or ten million bytes of data. You can figure your needs this way:

> For each five megabytes of disk storage, plan to use nine double-sided diskettes.
>
> Five Mbyte disk = 9 diskettes
>
> Ten Mbyte disk = 18 diskettes

See Section X for steps on use of STORE and RESTORE.

Step 2: Making PROM and Jumper Changes

Two changes to the Vector 4's hardware are part of the CP/M-86 conversion. You must:

1- Install the new Executive PROM provided with your upgrade kit, and

2- Cut two jumper wires that control how data for the screen display is stored

These changes should be performed by your dealer. They are described in the **CP/M-86 Hardware Installation Instructions.** This sheet is included in this documentation package, after the end of this manual. When these changes have been made, go on to Step 3.

Step 3: Formatting the Disk

Having made the needed hardware changes, you are ready to format the disk for CP/M-86.

In Vector's Disk Partition System, the FORMAT function is part of the UTILITY program. FORMAT writes on the disk a "framework" to be used by the operating system. For CP/M-86 and the Disk Partition System, it also defines a "partition" for the operating system and the size of the operating system's file storage area.

See Section IX for complete information on using UTILITY and the FORMAT function.

1. Press the RESET button at the rear of the console. The system banner will appear on the screen.
2. Insert the Vector Graphic CP/M-86 System diskette in the diskette drive and close the drive door.
3. Press [F].

Allow a few moments for the system to boot. Then the system message is displayed.

4. Format the hard disk following the steps outlined in Section IX.

Step 4: Installing CP/M-86 on the Disk

The disk is now formatted and ready for installation of the CP/M-86 operating system. This is done through another function of the UTILITY program, the System Generation (SYSGEN) function.

1. Recall the UTILITY program by typing

   A>**UTILITY [RETURN]**

2. Type [1] to select the System Generation program.
3. Sysgen the CP/M-86 operating system to the hard disk following the steps outlined in Section IX.

Step 5: Restoring Your CP/M Files to the Disk

This is the last step in converting your system: copying the files saved by STORE back to the hard disk. Use the RESTORE command, following the steps shown in Section X.

Certain CP/M files should NOT be RESTOREd to your disk, even though these were copied by STORE at the beginning of the conversion process. These are CP/M system files which have been replaced by newer versions CP/M-86. They are:

PIP.COM
STAT.COM
FORMAT.COM
GENSYS.COM
CONFIG.COM
DISKCOPY.COM
VECTOR.CPM
VECTOR.LD1
VECTOR.LD2

You do not want to have both CP/M and CP/M-86 versions of these files on your hard disk. Once the RESTORE process is complete, use the ERA (Erase) command to delete the files listed above.

Using CP/M-86 on Diskette-Based Systems

If your system has floppy-diskette drives (no hard disk), changing from Vector 4 CP/M to CP/M-86 means making two changes to the hardware of the computer. After that, you can boot the CP/M-86 operating system from its System Diskette just as you would boot Vector 4 CP/M.

These hardware changes should be performed by your dealer. They are described in the **CP/M-86 Hardware Installation Instructions.** This sheet is included in this documentation package, after the end of this manual.

APPENDICES

SYSTEM SPECIFICATIONS

CPU

The Z80B and 8088-2 CPUs can be individually selected through the use of Port 0CH and Port 0DH.

Z80B	Data Lines	8
	Address Lines	16
	Instructions	158 (including 8080 instructions)
	Clock Speed	5.1 MHz
	Power Supply	+5 V
8088-2	Data Lines	8
	Address Lines	20
	Instructions	235
	Clock Speed	5.1 MHz
	Power Supply	+5 V

MEMORY

PROM - One 4 K 2732 (or an 8 K 2764) and two 2 K 2716s. The 2732 is used for the Executive system program and the 2716s house the Character Generators.

2732 Addressed from 0000H to 0FFFH. This 4 K block is enabled/disabled through use of Port 02H or 03H. PROM is enabled (RAM disabled) when bit "0" is 0 and is disabled (RAM enabled) when bit "0" is 1.

2716 Used as Character Generators for the standard alphanumeric mode.

RAM - Sixteen 4164s (128 K system), four 74LS189s and one 74LS670.

An additional RAM memory board (using sixteen more 4164s) can be added to increase main memory from 128 K to 256 K.

I/O

Serial Ports - Three 8251 USART Chips are used to interface with the keyboard, serial printer and modem.

Keyboard	01H (Status/Control) 00H (Data)
Serial Printer	06H (Data) 07H (Status/Control)

Modem	04H (Data)
	05H (Status/Control)

Parallel Ports - One 8255 programmable I/O chip (with supplementary decoder chips) is used to interface with parallel printers using either a Centronics or NEC/Qume protocol.

Port A	08H
Port B	09H
Port C	0AH
Control Port	0BH

Other Ports - Other ports are used for the programmable timer, video controller, tone generator and to set various subsystem parameters.

ROM Enable (and color)	02H
	03H
CPU Swap	0CH
	0DH
Video Controller	0EH
	0FH
Timer	10H (Timer 0)
	11H (Timer 1)
	12H (Timer 2)
	13H (Control)
RAM Address Map	16H
	17H
Tone Generator	18H
	19H
Color Map	1CH
	1DH
	1EH
	1FH

Expansion Slots - Two modified S-100 expansion slots are available for additional boards. Since the main system uses a Switching Regulated Power Supply these boards must not have on-board regulation.

CLOCK SIGNALS

A 32.640 MHz oscillator generates a signal which is used directly by the Video (Dot Clock) and other sub-systems.

INTERRUPTS

A programmable timer (8253) is used to generate one interrupt and two baud rates.

KEYBOARD

The Vector 4 uses a detachable keyboard which has 91 keys. that are accessed through a serial interface.

DRIVES

The Vector 4 can use several different disk drive configurations. These include one 5 1/4 inch floppy and one 5 1/4 inch hard disk, or two 5 1/4 inch floppies. The following chart gives the specifications for the individual drives.

Drive Type Specifications

5 1/4 inch Floppy - Hard Sectored	
	Capacity- 630 Kilobytes Transfer Rate- 250 Kilobytes/sec Rotation Speed- 300 RPM
5 1/4 inch Hard Disk	Capacity- 5 or 10 Megabytes Transfer Rate- 5 Megabytes/sec Rotation Speed- 3600 RPM

VIDEO MONITOR

Screen Size	12 in. diagonal CRT	
Resolution	312 lines	
	Green Phosphor	
	Alphanumeric Mode	24 lines x 80 characters Each character is a 16h x 13v pixels
	High Resolution Mode	640h x 312v pixels (B & W)
	Gray Scale Mode	160h x 312v pixels (16 levels) 320h x 312v pixels (any 4 or 16 levels)
	Color Mode (use external monitor)	160h x 312v pixels (8 colors) 320h x 312v pixels (any 4 of 8 colors)
Bandwidth	25 MHz	
External Controls	Contrast	
Internal Controls	Vertical Hold	
	Vertical Height	
	Vertical Linearity	
	Vertical Centering	
	Focus	
	Brightness	
	Horizontal Centering	

POWER REQUIREMENTS

Uses a Switching Power Supply which provides regulated, DC voltage to the Monitor (+12 V), main PCB (+5 V, -12 V, +12 V) and drives (+5 V, +12 V).

Voltage Option	115 AC +/- 15%	230 AC +/- 15%
Frequency	47 Hz to 63 Hz	47 Hz to 63 Hz
Current, Operating [1]	2 A	1 A
Current, Surge [1]	2 A	1 A
Power Dissipation [1]	190 W	190 W
Heat Generation [1]	60 BTUs	60 BTUs

[1] Maximum values.

DIMENSIONS AND WEIGHT

Main Enclosure (1)

Height	14.0 in	35.5 cm
Weight	37.5 lbs	17.0 kg
Depth	17.0 in	43.2 cm
Width	19.0 in	48.3 cm

Keyboard Subsystem

Height	1.9 in	4.8 cm
Weight (2)	4.0 lbs	1.8 kg
Depth	8.6 in	21.8 cm
Width	17.5 in	44.4 cm
Length (3)	40.0 in	101.6 cm

[1] Values given are for system which includes one 5 1/4 inch floppy drive, one 5 1/4 inch hard disk, one FD Controller Board and the external power cable. Two-floppy system will be slightly lighter.

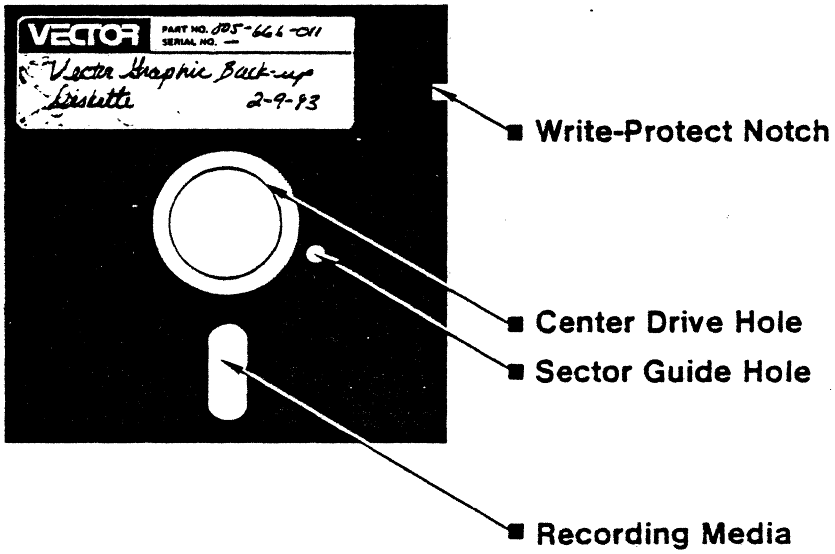
[2] Includes cable.

[3] Connecting cable fully extended.

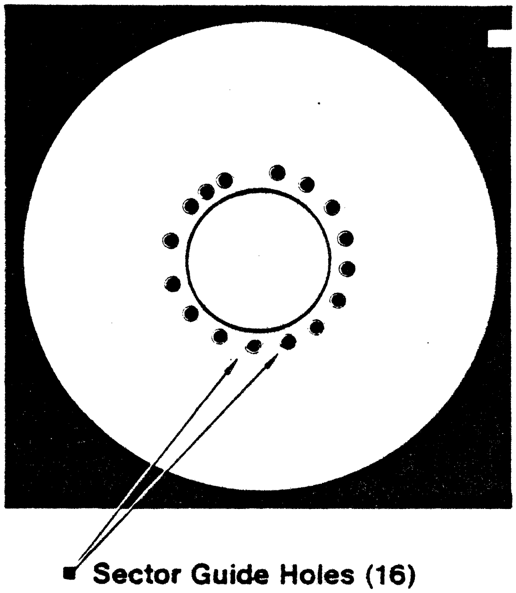
CARE AND HANDLING OF DISKETTES

Type Of Diskettes

All Vector 4 systems with floppy drives require an industry-standard 5-1/4" floppy diskette with 16 "hard sectors." This means that there are 16 sector guide holes around the center drive hole. These sector guide holes are used by the computer in reading from and writing to the disk.



When purchasing diskettes, be sure to specify "hard-sectored" diskettes in your purchase order or to your salesman. "Soft-sectored" diskettes (which do not have the sector guide holes) will not work on your machine. For best results, use Vector Graphic-Approved diskettes (Part Number 1ØØ1-Ø161).



Sector Guide Holes (16)

Handling And Safety Precautions

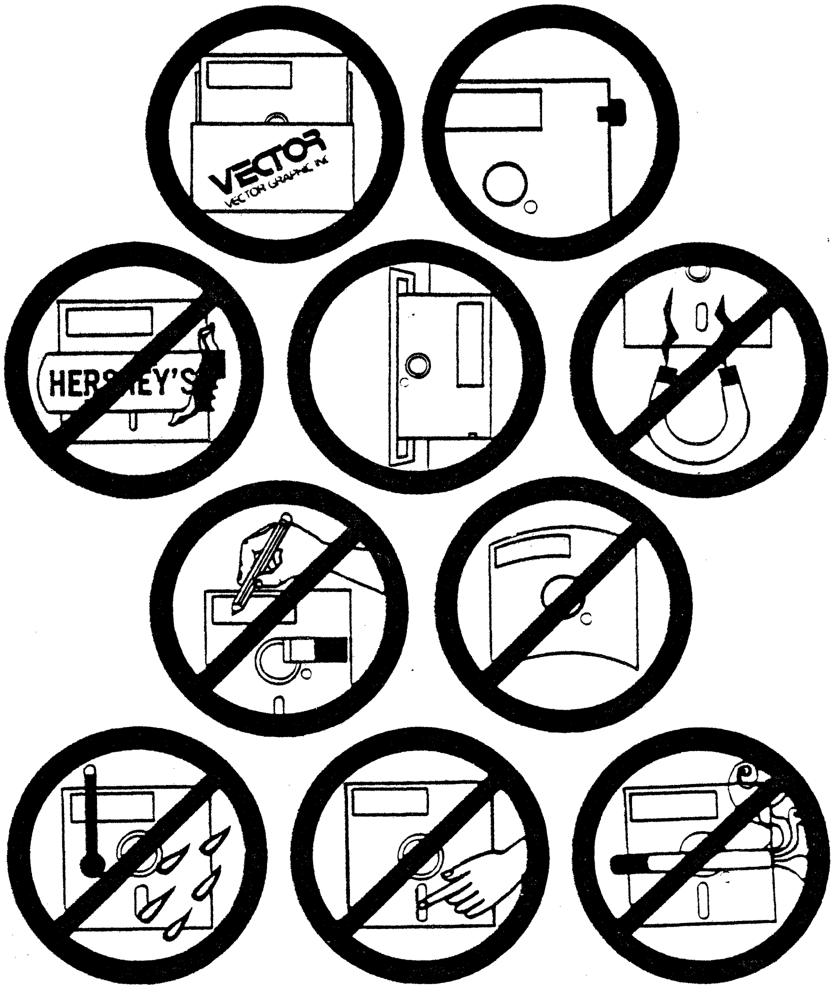
Guidelines

The ideal conditions for diskettes are the same as those for humans: a temperature of about 70 degrees, a relative humidity of 50%, and a dust- and dirt-free environment. (The maximum range of temperature and humidity for diskettes is 50-125 degrees, 8%-80% relative humidity. Avoid the extremes if possible.) For most effective use of your diskettes, observe the following guidelines.

- Do not touch or try to clean the surface of the diskette itself.
- Do not smoke, eat, or drink while handling the diskette. (A particle of smoke is larger than a particle of the magnetic oxide on the recording surface.)
- Do not place heavy objects on the diskette.
- Do not expose the diskette to direct sunlight or excessive heat.
- Do not use rubber bands or paperclips on the diskette.
- Write on diskettes only with fiber-tipped pens. Ball-point pens can distort the surface of the diskette. Even with a fiber-tipped pen, use minimal pressure when writing and make sure that the diskette is placed in its protective envelope before writing on the case.
- Do not use labels which have been written in pencil. Graphite particles from the pencil marks can harm the diskette recording surface.
- The best way to label a diskette is to write or type the information on an adhesive label, then apply the label carefully to the diskette case. Be careful not to let the adhesive touch the disk recording surface.
- Do not expose the diskette to magnetic fields. A strong field can erase all the information on a diskette. Avoid static electricity for the same reason.
- After use, always return the diskette to its protective envelope or to some other protective container, such as the Vector Graphic Diskette storage case.
- Store diskettes vertically. Like phonograph records, diskettes are susceptible to warping and compression if stacked.
- If a diskette becomes damaged or contaminated, replace it. Placing a contaminated diskette in the disk drive can pass the contaminant on to the read/write head, damaging it in turn.

See "Using Diskettes" in Section III for instructions on how to load and mount diskettes for use.

RULES FOR GOOD HANDLING OF DISKETTES



Diskette Backup, Replacement, And Protection

From time to time, you will find a diskette that simply doesn't work. Computer users have met with batches of diskettes with the wrong number of sector holes, two diskettes mounted in one jacket, and various other manufacturing defects. If a diskette does not work reliably, replace it immediately.

If you have any very old records or tapes, you know that any electronic recording medium deteriorates with time. But whereas an old tape or album begins to hiss and crackle after a few years, an old diskette produces errors. You can avoid the problem of bad data by making backup copies on a regular schedule. Both program diskettes and data base diskettes should be backed up regularly.

Copying To The Backup Diskette

To copy information to a backup diskette, use any of these programs:

UTILITY: Diskcopy Function
STORE/RESTORE
PIP

Use the Diskcopy Function and STORE to copy the entire contents of a disk (the Diskcopy Function for floppy to floppy, STORE for hard to floppy). Use PIP for individual files, files with similar names or extensions (allowing you to use ambiguous file names, as described in the Sections V and X), or files to be copied from one user area on a disk to another.

The Diskcopy Function of UTILITY is described in Section IX. The PIP and STORE/RESTORE programs are explained in detail in Section X, "The Transient Commands."

Replacement

Be sure to replace frequently-used diskettes with fresh ones every six months. Do not wait until the old disk fails to replace it, especially if you do frequent copying from the disk to your computer. One suggestion is to use the backup diskette, while still fairly fresh, as the new front-line diskette, and to create a fresh backup. This minimizes both the chance of diskette failure and the cost of buying replacement diskettes.

To create a backup diskette for either a diskette or the hard disk:

1. Using the Format Function of UTILITY, format a blank diskette or a diskette that does not contain important information.

CAUTION

Do NOT format the CP/M-86 System diskette or any other diskette containing needed information. Formatting completely erases the contents of a diskette.

2. Using the Diskcopy Function or STORE, move the contents of the source disk or diskette onto the formatted target diskette. (Note: STORE transfers only from hard disk to floppy diskette.)

3. If necessary, PIP any other desired files to the target diskette.

Protecting Diskettes From Unauthorized Changes

To protect information on a diskette from change, use the write-protect tabs included in each box of new diskettes. When you place a tab over the write-protect cutout on the diskette cover, the disk drive will not allow you to erase or change any information on the diskette. The tab may be removed later, if required.

It is a good idea to place one of these tabs on the system diskette before working with it, if not already attached.

You can also use the STAT command to protect a diskette or individual files on the diskette from change or erasure. See the instructions for the STAT command in Section X, "The Transient Commands."

USER'S GUIDE TO TROUBLESHOOTING THE VECTOR 4

BEFORE YOU BEGIN

This section outlines a few simple procedures for finding the causes of problems you might have with your Vector 4. The Troubleshooting Guide is divided into three parts:

- a description of the desirable environmental conditions for operation of your Vector 4;
- a flowchart showing how to run the system in a controlled environment to try to duplicate a symptom;
- a series of symptom-related check-out procedures, allowing the noted symptoms to be checked against possible causes.

If you have never used diskettes, be sure to read "Using Diskettes" in Section III and "Care And Handling of Diskettes," Appendix B, before operating your Vector 4. This will ensure that any problems that arise are not the result of inadvertent damage to the system diskette.

ENVIRONMENTAL CONSIDERATIONS

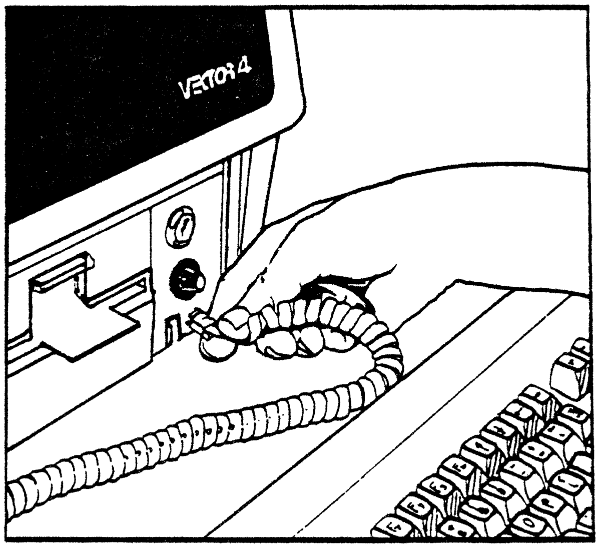
Vector 4 SBC Power Requirements

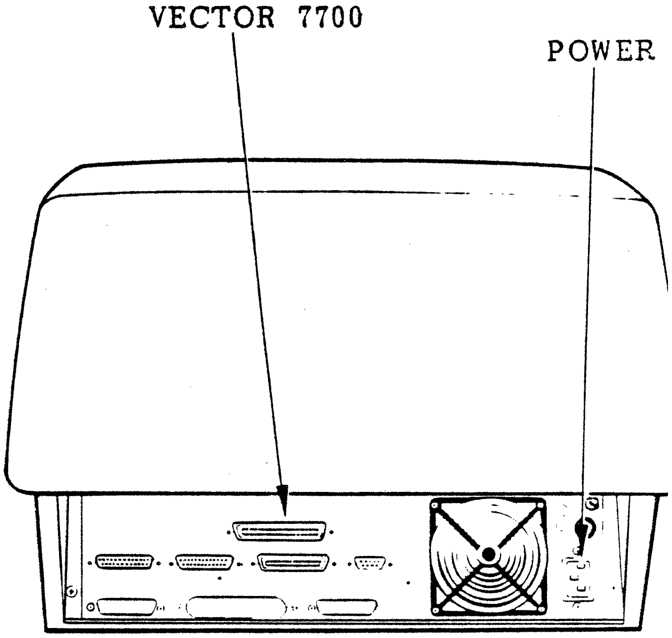
For accurate operation and troubleshooting, make sure that the Vector 4 is supplied with current meeting the specifications outlined in Appendix A. Use another appliance, if necessary, to test the power outlet in use for correct functioning. A typewriter, clock, or desk lamp will suffice. Be sure to remove the appliance before installing the Vector 4.

Make sure that no machinery or motor-driven equipment drawing large amounts of current, such as a copier or air conditioner, is connected to the same line as your Vector 4. Any equipment which pulls a heavy and intermittent load from the power line can cause the Vector 4 to lose current and shut down. This may cause loss of data during operation.

Connectors

Make sure that all cables and connectors on the Vector 4 are tightly inserted into the appropriate socket. See the illustration below for locations.





In particular, check the tightness of the keyboard cable and the power cord. These two connectors are crucial to the operation of the Vector 4. If necessary, remove the keyboard connector, then reinsert it into the socket. Be sure to squeeze the locking tab toward the connector during insertion. When inserting, listen for the locking mechanism to "click." This indicates a good connection.

General Instructions

If your Vector 4 has been set up properly, the following troubleshooting procedures will provide a valid and accurate guide to user-serviceable malfunctions. At this point, the system should be:

- o turned OFF (with the key in a vertical position),
- o plugged into a verified power outlet,
- o with all connectors (power cord, keyboard cable, printer or peripheral strips, etc.) checked for correct positioning and tight fit.

Next identify your Vector Graphic System Diskette. If you have a floppy-only system, insert the diskette into the right-hand drive (Drive A). If it has already been inserted, remove it and reinsert it according to the procedures outlined in Section III, "Using Diskettes." If you have a hard-disk system, keep the System Diskette nearby for possible use.

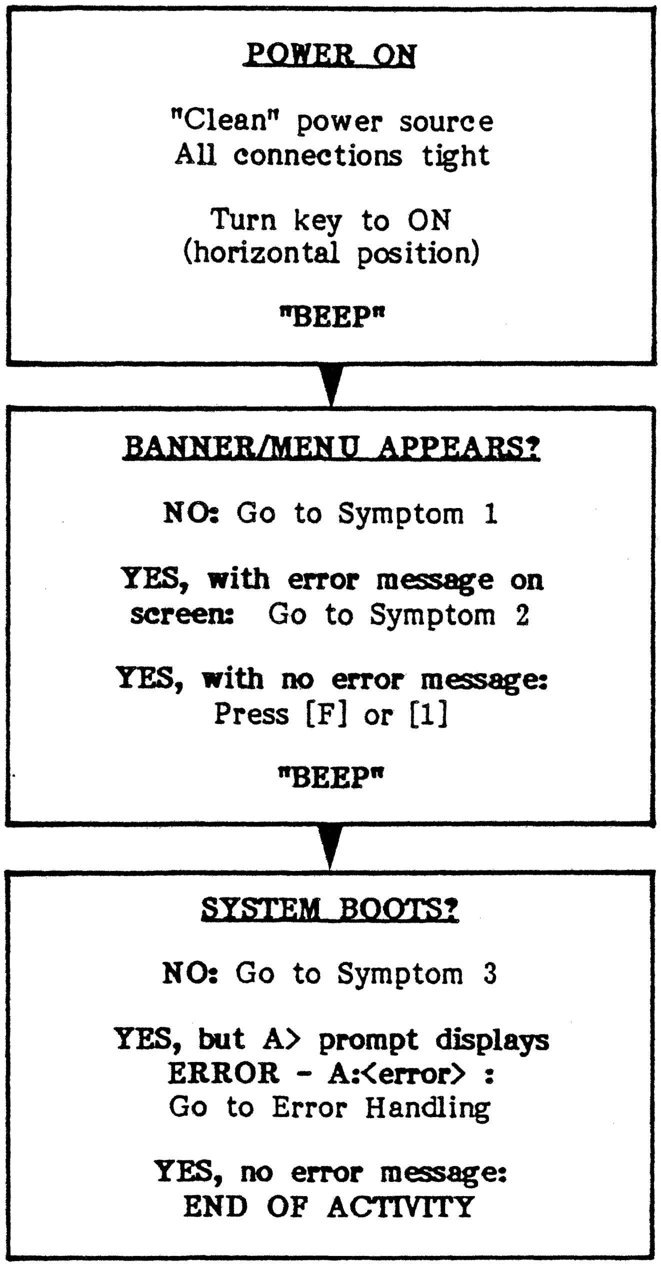
Now proceed to the Troubleshooting Flowchart. Follow the checks outlined there while trying to make the symptom repeat in the controlled environment.

If the symptom does not repeat, the problem must have been due either to an improperly-connected power cord or cable, some characteristic of the power line itself, an improperly-inserted diskette, or some environmental problem. Note the current operating conditions (power outlets, temperature, etc.) and try to maintain them for regular operation of the system.

If the symptom does repeat, check the indicated symptom (isolated by number) against the Symptom Checks. Follow these procedures closely until the Vector 4 is functioning.

If after taking these steps you still cannot get your Vector 4 to work, check your service agreement to determine the appropriate action.

TROUBLESHOOTING FLOWCHART



POWER ON

"Clean" power source
All connections tight

Turn key to ON
(horizontal position)

"BEEP"

BANNER/MENU APPEARS?

NO: Go to Symptom 1

YES, with error message on screen: Go to Symptom 2

YES, with no error message:
Press [F] or [1]

"BEEP"

SYSTEM BOOTS?

NO: Go to Symptom 3

YES, but A> prompt displays
ERROR - A:<error> :
Go to Error Handling

YES, no error message:
END OF ACTIVITY

SYMPTOM CHECKOUT PROCEDURES

SYMPTOM 1 - NO BANNER OR MENU AT POWER-ON

Symptom Review: Your Vector 4 shows no Executive banner or booting menu, and no other screen display, at power-on or RESET.

1. Turn system OFF. Wait 60 seconds, then turn system ON. Make sure that key is at fully horizontal position. Listen for tone generator to "beep".

   If tone generator "beeps", go to Step 3.

   If tone generator does not "beep", feel for air coming out of fan at rear of terminal. If no air comes out of fan, continue.

2. Make sure that Vector 4 is correctly plugged into working electric socket, and that connector of power cord is fully plugged into Vector 4 with no obstructions. If necessary, verify electric socket with other appliance.

   If other appliance does not work in electric socket, install Vector 4 in known working socket (and call electrician regarding other socket).

   If other appliance works in electric socket, Vector 4 power supply needs inspection. Check your service agreement for the appropriate action; the power supply is not user-serviceable.

3. Wait 30 to 60 seconds for terminal to warm up. If no banner displays after 60 seconds, turn contrast knob all the way counterclockwise, then back to full clockwise position.

   If banner displays, return to Operating Instructions.

   If banner still does not display, Vector 4 monitor or power supply will need inspection. Check your service agreement for the appropriate action; this problem is not user-serviceable.

END OF ACTIVITY

SYMPTOM 2 - ERROR MESSAGES ON SCREEN

Symptom Review: Your Vector 4 turns on with the key, generates the tone to indicate power-on, and displays the Executive banner with an error message.

The Automatic System Test runs each time the Vector 4 is powered or reset. It checks the memory, the booting program, and the keyboard. If the automatic test uncovers any errors, it displays one of the following messages:

MEM ERR, BLK xx, EVEN/ODD ADDR, MASK = xx*
PROM CS ERROR
KBD ERROR

* "xx" represents a two-digit number.

These errors may be the result of a temporary condition.

1. Press the RESET button at the rear of the console cabinet.

2. If the error message again displays, turn the system OFF with the key, wait 60 seconds, and then turn the power ON.

   If the error message still repeats, repeat Step 1 once. If the screen still displays the same error message, write down the message exactly as it appears on your screen and check your service agreement for the appropriate action.

   If **KBD ERROR** displays, continue.

3. The **KBD ERROR** message may appear simply because the keyboard connectors are not sufficiently tight. If you get this message, remove the keyboard cable and connect it firmly, listening for the "click" to indicate a positive connection. Then repeat Step 1 once.

   If the **KBD ERROR** message repeats, check your service agreement for the appropriate service action.

END OF ACTIVITY

SYMPTOM 3 - VECTOR 4 WON'T BOOT

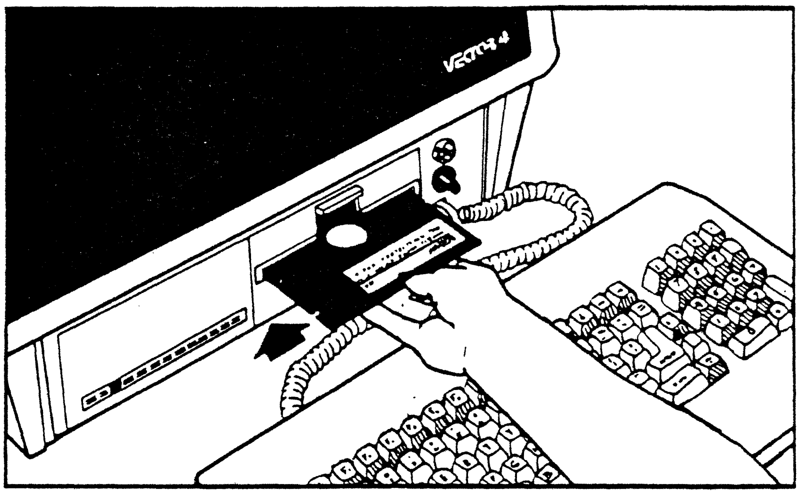
Symptom Review: Your Vector 4 has successfully powered on, and the screen displays either the Basic Menu or the Systems Menu. No error messages appear as a result of the Automatic System Test, but when a selection key is pressed, the screen does not (after several seconds) clear and display:

```
VECTOR GRAPHIC CP/M-86 VERSION V.VV (XX)

A>
```

For A Floppy-only System:

1. Determine which diskette supplied with your system is the Vector Graphic System Diskette #1.
2. Make sure that this diskette is inserted in the right-hand floppy drive, in the position shown below.



3. Press [F] to boot from the System Diskette.

If system still will not boot, check your service agreement for the appropriate action.

For A Hard-disk System

If the banner disappears, and the screen displays the letter "e" on a light background, press the RESET button. See Section IX, "UTILITY and CONFIG," and Section XI, "Installing CP/M-86 on Your Hard Disk," for instructions on how to install the CP/M-86 operating system on your hard disk.

END OF ACTIVITY

ERROR HANDLING

Your system is designed to detect and report different kinds of errors. Sometimes errors occur in the regular operation of hardware components—for example, when you ask for data to be read from or written to a diskette or the hard disk. The operating system automatically tries to correct these, performing the needed function again until it can be done successfully. Other errors are user-related: they stem from wrongly entered commands or mistakes in the use of system functions.

The two sections below show error messages that are generated by the Disk Partition System, including the CONFIG Utility, the CP/M Simulator, and the Disk System. Except for the last, which is a WARNING, each message shown is displayed preceded by "ERROR - ".

There also are error messages produced by the CP/M-86 operating system itself. These are described in your **CP/M-86 User's Guide** (Digital Research), Appendix D. See that section for information on what steps to take when one of these messages appears.

CONFIG Error Messages

CONFIG REVISION NOT COMPATIBLE WITH SYSTEM REVISION

The revision level of the system is below that required to run the particular version of the CONFIG Utility.

UNABLE TO SET LIST DEVICE

This message can have three causes:

1 - The requested device (printer or communications device) was busy at the time an attempt was made to load the driver (using a CONFIG PLOAD or CLOAD command);

2 - The device was busy or disconnected at the time of an attempt to select it;

3 - The device was busy or disconnected at the time of an attempt to change the current list device.

INPUT FILE ALREADY SPECIFIED

A second input file was specified when only one input file is allowed.

INPUT FILE NOT FOUND

The input command file was not found.

DRIVER FILE ERROR

The driver file is empty or contains bad code.

UNABLE TO GET PARAMETERS

The executing command file was unable to get the specified parameters. This error should not occur, because a PSEL command, specifying the parameters, must be executed first.

UNABLE TO SET PARAMETERS

The executing command file was unable to set the specified parameters. This error also should not occur because it must be preceded by a PSEL command, which sets the parameters.

UNABLE TO INITIALIZE DRIVER / DEVICE

The program was unable to initialize a printer or communications driver after loading it. This error normally should not occur.

*** BAD VIDEO DRIVER REPLY ***

The wrong ESC sequence was received from the video driver after a call was made to it. This error will not occur unless there is a problem with the video driver or the system has been corrupted, in which case you should reboot.

DRIVER FILE NOT FOUND

The printer or communications driver file was not found on an attempt to load it from a specified disk drive.

KEYBOARD REASSIGNMENT FILE NOT FOUND

A file specified for keyboard conversion could not be found.

KEYBOARD REASSIGNMENT FILE ERROR

The keyboard reassignment file could not be closed.

INVALID KEY CODE FOR REASSIGNMENT

An invalid code was specified.

TOO MANY KEY CODES FOR REASSIGNMENT

More than 80 codes, including the one for the key to be reassigned, have been specified.

KEYBOARD REASSIGNMENT TABLE FULL

The conversion table holding the reassignment codes is full.

WARNING - ORIGINAL LIST DEVICE NOT RESTORED

An attempt was made to set the list device back to its original state after a printer or communications driver command was executed (except PSEL) and was not successful. This could occur only in a multi-user environment.

CP/M Simulator Error Messages

The following error messages can appear during use of the CP/M Simulator.

NOT ENOUGH MEMORY

The internal memory (RAM) available for use is not large enough to hold the Simulator program. Normally, this message appears only in advanced programming, when memory used for a previous program has not been freed to make room for following programs.

ERROR WHILE PARSING COMMAND LINE

The Simulator has been unable to understand your command line. Check your command for mistakes in syntax or spelling. The command for using a program through the Simulator is the same as it would be if you were running the program under Extended CP/M or Vector 4 CP/M.

See Section X for more information on the RUN8 command file.

EXECUTION FILE NOT FOUND

The Simulator could not find the CP/M program or command file named in your command. Use DIR to check that the program is available on your user area. The program must exist on your local user area or must be in User Area Ø and have been set to SYS status through the STAT command.

FILE IS NOT AN EXECUTABLE Z8Ø PROGRAM

This message appears when the program name in your command includes an extension (usually .COM). Enter the program name by itself, without its extension.

ERROR - NO PROGRAM NAME SPECIFIED

Appears when "RUN8" is entered instead of the name of a CP/M (.COM) program. You can enter a "RUN8" command line instead of a program name when you want to see the version number of the Simulator. A copyright message including the version number is displayed below the error message.

EXAMPLE:

```
A>RUN8 [RETURN]
```

The error message appears, then this copyright statement:

```
RUN8 - VECTOR GRAPHIC EXTENDED CP/M SIMULATOR - REV. X.XX (AA)*
COPYRIGHT (C) 1983, VECTOR GRAPHIC INC.
```

(*"X.XX" is replaced by the actual version number; "AA" is replaced by the actual release letters.)

ERROR - INCOMPATIBLE REVISIONS EXIST BETWEEN SIMULATOR AND HOST SYSTEM. CP/M-86 REVISION 1.Ø OR GREATER REQUIRED

The CP/M-86 system installed in your Vector 4 and the Simulator program will not work together; they are of different versions. The operating system version message appears when you first boot CP/M-86. The Simulator version message can be seen by typing "RUN8", as described above.

Disk System Error Messages

Soft and Hard Errors

Error messages for the disk system can report "soft" or "hard" errors.

- Soft errors are those which the operating system has been able to correct, usually by trying the requested function again until it can be done. Soft error messages report this result, letting you know that an error occurred but does not affect your data. For these messages to appear, reporting of soft errors must be enabled by the DSOFT command of CONFIG (CONFIG DSOFT 1).

- Hard errors are those which the operating system has been unable to correct. For these messages to appear, prompting must be enabled by the DPROMPT command (CONFIG DPROMPT 1). Each error message displayed offers three responses to the error condition:

  - [A]bort: If you type [A], you abort (end without success) the system's efforts to complete the function that produced the error.

  - [R]etry: Type [R] to ask that the function be done again.

  - [I]gnore: Type [I] to indicate that the function should be completed in spite of the reported error. You might choose this option if you thought you could fix the error yourself—for example, repair a text file that had been read from the disk with minor errors (several characters changed).

Soft Error Messages

DISK ERROR CORRECTION DONE

RETRY ON DISK READ

WRITE VERIFY ERROR - SECTOR RE-WRITTEN

DISK DRIVE FAILURE

Hard Error Messages

DRIVE NOT READY / READING FROM DRIVE D

DRIVE NOT READY / WRITING TO DRIVE D

DISK ERROR / READING FROM DRIVE D

DISK ERROR / WRITING TO DRIVE D

**PHYSICAL ERROR <error code>
<ON WRITE / VERIFY> , SECTOR <sector #>**

DRIVE NOT READY messages can occur if you try to access a floppy drive when the disk is not properly seated, the drive door is open, or there is no diskette in the drive. Check for these conditions, then enter the desired response from the three shown above. Sometimes all it takes to correct one of these errors is that you remove the diskette, remount it, and enter [R] to try the operation again.

DISK ERROR messages occur when the disk system's built-in checks show that data has not been read or written correctly. These errors can be the result of temporary conditions affecting the hardware of the system. In this case, a retry may correct the fault. But if an error continues to appear, the problem may be in the hardware itself. In this case, seek technical assistance as provided for in your service agreement.

A PHYSICAL ERROR message can appear during use of the FORMAT function of UTILITY. The message includes an error code pointing to the kind of error that has been detected. The meaning of these codes are:

Code	Type of Disk	Type of Error
11	Hard	Drive not ready
12	Floppy	Write-protected floppy
13	Hard	Write-fault
14	Hard or Floppy	Track Ø not found
15	Floppy	Floppy door open or no floppy inserted
21	Hard or floppy	Controller-busy timeout
22	Hard	Seek-complete timeout
23	Hard	Loss-of-sync timeout
31	Hard or Floppy	Bad command
32	Hard or Floppy	Drive error
33	Hard or Floppy	Bad record value or record count
41	Hard or Floppy	Track values don't match

Code	Type of Disk	Type of Error
42	Hard or Floppy	Sector values don't match
43	Hard	Head values don't match
44	Hard or Floppy	Data verification error
51	Floppy	Checksums don't match
52	Floppy	Post-write checksums don't match
53	Hard or Floppy	ECC error
54	Hard or Floppy	Uncorrectable ECC error
55	Hard or Floppy	ECC sync error
56	Hard or Floppy	Sync byte missing

Some of these errors are easily interpreted and quickly corrected. Others may require the assistance of your dealer. For instance, if this message comes up—

physical error 12
on write, sector NN

(*"NN" will be replaced on screen by the actual sector numbers)

—you probably can correct the problem. Check the diskette's write-protect notch (the cut-out in the upper right-hand corner). If it is covered by a foil write-protect tab, the diskette has been made read-only. Unless the diskette is write-protected for good reason, you can remove the tab and try the operation again. (See Appendix B, "Care and Handling of Diskettes," for more information on using diskettes; see Section X, "The STAT Command," to learn how to set the Read/Write status of a diskette in other ways.)

In the same way, it's likely that Physical Error 15 ("Floppy door open or no floppy inserted") can be easily corrected.

When any error message occurs, first take whatever steps might correct the problem (such as closing a drive door or inserting a diskette). Then try running the operation that produced the error again. If an error appears repeatedly, or if it is one not easily understood and corrected, contact your dealer for assistance.

Recovery Techniques

If error messages appear continually in the use of one diskette, the diskette probably is defective, and it should be replaced. Error messages sometimes appear when you are formatting new diskettes for use in backup. If the errors persist after several tries with a diskette, return or discard it. Remember to unload and reload a diskette any time you repeat a disk operation following an error. This ensures that that the diskette is seated correctly.

If you meet with errors in working with a diskette already holding valuable data, and you cannot access the needed files, try copying all files to another diskette using the DISKCOPY function of UTILITY. Or, if yours is a hard-disk system, try to PIP the contents of the diskette to your hard disk. If the error still occurs, try switching source and destination drives. Sometimes a repositioning of diskettes within drives can give a successful copy.

If you cannot make a complete copy of a diskette in these ways, try to PIP its files one-by-one to another formatted diskette. If the problem is limited to one or two files, the other files should copy successfully. Then, by enabling user prompting (through the DPROMPT command of CONFIG), you can direct the PIP program to ignore the errors in the faulty file, and the file can be copied. Unless they are very numerous, the errors can be corrected later through editing of the file on screen.

It is important to use a good, reliable brand of diskettes, such as Vector Graphic-approved diskettes (Part Number 1ØØ1-Ø161), and to replace each diskette when necessary.

CP/M Programs Not Supported Under the CP/M Simulator

The following utility programs are not supported under the CP/M Simulator:

PROGRAM NAME

GENSYS.COM
GENLIST.COM
FORMAT.COM
DISKCOPY.COM
CONFIG.COM
- -P Option
- -S Option
- -F Option
- -D Option
- -A Option
- -O Option

DISKTEST.COM
MEMTEST.COM*

*NOTE: MEMTEST.COM is replaced by MEMTEST.CMD. A version of DISKTEST.COM suitable for CP/M-86 is in development.

The following programs are partially supported under the CP/M Simulator:

PROGRAM NAME

RECLAIM.COM:
- -$S Option
- -$E Option

ERAX.COM
EDIR.COM

GLOSSARY OF TERMS

Terms in **BOLD FACE** within a definition are defined in separate entries in this Glossary.

AMBIGUOUS FILENAME (AFN) - A filename that includes a **WILDCARD** character, either * or ?, and so refers to more than one file. Ambiguous filenames can be used with some CP/M-86 commands but not with others. Example: the command DIR *.MEM displays a listing of all files ending with the .MEM extension. See also **UNAMBIGUOUS FILENAME.**

APPLICATION PROGRAM - A special program created to direct the computer in performing a specific business task, such as word processing, accounting, or payroll. See also **SOFTWARE.**

ASCII - American Standard Convention for Information Interchange: the most widely accepted set of codes used to represent letters, numbers, and control characters in microcomputers. Each ASCII code is made up of eight **BITs**. The Vector 4 can work with 256 different ASCII codes.

BOOTING - The process of loading the operating system from the disk into the computer's internal memory after power has been turned on. The booting command, given by the user, is "F" when booting from a floppy disk and "1" for for CP/M-86 when booting from the hard disk.

BDOS - Basic Disk Operating System. The portion of the CP/M-86 Operating System responsible for locating and recording data on the disk drives.

BIOS - Basic Input-Output System. The portion of CP/M-86 responsible for controlling input and output. The BIOS opens and closes circuits to move data within the system.

BIT - A "Binary digIT", the smallest single piece of information that can be stored in the computer's memory. Usually represented as "1" or "0," a bit is actually the relative presence or absence of electrical current at a particular location in some a memory device.

BUILT-IN COMMANDS - A group of commands loaded into the internal memory of the computer during the booting process. These commands are part of the **CCP.**

BYTE - For the Vector 4's operating systems, a coded group of eight **BITs** that represents one character. For example, the character [CTRL C] is represented in the computer by the byte 00000011.

CCP - Console Command Processor. The portion of CP/M-86 responsible for interpreting input from the keyboard and sending commands to the rest of the operating system. The **BUILT-IN COMMANDS** are part of the CCP.

COMMAND - An instruction the user gives to the computer.

CONTROL KEYS - Special keys on a computer's keyboard that, by themselves or in combination with one other key, send commands to the computer. For example, **[ESC]**, **[DEL]** and **[CTRL-H]** are control keys.

CONSOLE - The part of your Vector 4 that contains the video screen, central processing unit (**CPU**), and other electronic components.

CP/M-86 - Control Program for Microprocessors ("86" refers to the type of **MICROPROCESSOR** used). An extremely popular operating system used extensively in the microcomputer industry. (CP/M-86 is a registered trademark of Digital Research.)

CPU - Central Processing Unit. The "brains" of a computer. The CPU is responsible for making decisions, performing calculations, and executing instructions. Includes and sometimes is the same as the **MICROPROCESSOR.**

CURSOR - A flashing box, arrow, or line seen on the video screen, used to indicate the location of the computer's "attention."

DISK - Magnetic media used to store data and programs used on the computer. A disk may be flexible, called a "floppy diskette," or rigid, called a "hard disk" or "Winchester."

DISK DRIVE - The hardware device that houses either the hard disk or the floppy diskette. The computer's component for mass storage of information.

EXTENSION - The three characters after the period in a **FILE SPEC.** The extension describes the type of the file. Example: <file name>.MEM. "MEM" is the extension, and indicates that the file is a MEMORITE file.

FILE - A collection of related information recorded on disk under a specific name.

FILENAME - The first part of a **FILESPEC.** The filename may be up to eight characters long, and identifies a file of data stored on the disk. Example: in the filespec LETTERS.MEM, the filename is "LETTERS."

FILESPEC - The way to access or refer to a **FILE.** The filespec consists of two parts: the **FILENAME**, which can be up to eight characters long, and the **EXTENSION,** which is three characters long.

FORMAT - A UTILITY function used to prepare a **DISK** for use. FORMAT puts track and sector markings on the recording surface to prepare it for use. For the hard disk, FORMAT also defines partitions for the operating system or systems that are to be installed on the disk.

FUNCTION KEYS - The keys at the top of the keyboard marked "f1" through "f14." They are used for special functions in application programs.

HARDWARE - The physical components of the computer. Disk drives, consoles, printers, and keyboards are all examples of hardware components.

KILOBYTE - A term that describes 1,024 bytes, a **BYTE** being equivalent to one character. The abbreviation is "k", so a diskette may be said to hold 630k or approximately 630,000 characters.

MEGABYTE - A term that describes a million (actually 1,048,576) **BYTES** or characters. The abbreviation for megabyte is "M", so a hard disk may be said to hold 5M or approximately 5 million characters.

MICROPROCESSOR - The "brain" of the computer. A complex of miniaturized electronic circuits etched on a tiny silicon chip, the microprocessor performs calculations, determines logical values, and guides the functions of the computer following the instructions of **SOFTWARE** and the user's **COMMANDS.**

OPERATING SYSTEM - A program that controls the flow of data into and out of the components of the computer, manages **FILES**, and allows the user to load application **PROGRAMS.**

PIP - Peripheral Interchange Program. A **TRANSIENT COMMAND** used to copy **FILES** from one disk (or device) to another.

PRINTER - The hardware component of the computer used to print copies of files or records when a "hard copy" is needed.

PROGRAM - A series of instructions that directs the computer in performing some task. See **SOFTWARE** below.

RAM - Random Access Memory. A type of volatile (changeable) memory used by the microprocessor to store data or instructions temporarily. Also used to store the operating system while it is in use.

ROM - Read Only Memory. A type of stable memory used to store certain frequently-used instructions even when power is off. The booting program, for example, is stored in ROM chips.

SOFTWARE - Programs of instructions written to direct the computer in specific functions. Software includes two kinds of programs: **APPLICATION PROGRAMs**, which usually are business-oriented (such as word processing or preparation of spreadsheets); and system programs, which control the computer **HARDWARE** to run the application programs.

TRANSIENT COMMANDS - A group of utility programs stored on the system disk; used for file maintenance and as support to the **OPERATING SYSTEM.** Transient commands are moved from the disk to internal memory each time they are called into use; they cannot be used unless the user has that disk's prompt on the screen.

TPA - Transient Program Area. The portion of RAM that is "left over" after booting. It is used to store **APPLICATION PROGRAMs, TRANSIENT COMMANDs,** and newly entered data. Also a **BUILT-IN COMMAND** that allows the user to return to the last task he or she was performing.

UNAMBIGUOUS FILENAME (UFN) - A specific filename that refers to only one file on a directory. It cannot contain **WILDCARD** characters. See also **AMBIGUOUS FILENAME.**

USER AREA - A space on the disk assigned by the **OPERATING SYSTEM** and used to store files. User areas divide large disk capacity into smaller work areas. Up to 16 user areas may be selected through the USER command.

UTILITY - Another name for any of the **TRANSIENT COMMANDS** used for file maintenance.

WILD CARD - One of two special characters that can be used to refer to multiple files using one **AMBIGUOUS FILE NAME.** The wild cards symbols are:

? - replace with any single character in the location entered

***** - replace with any character or group of characters needed to fill up that part of the file spec (up to eight for filename, three for extension)

DISKETTE DIRECTORIES

Vector Graphic System Diskette

The CP/M-80/86 System Diskette supplied with your Vector 4 contains the following files:

In User Area 0:

- CPM-86.SYS
- UTILITY.CMD
- SERIAL.DEV
- STORE.COM
- HELP.CMD
- DDT86.CMD
- TOD.CMD
- RUN8.CMD
- CONFIG.CMD
- PARALLEL.DEV
- RESTORE.COM
- HELP.HLP
- ASM86.CMD
- SUBMIT.CMD
- CPM-86.BEG
- PIP.CMD
- NEC.DEV
- STAT.CMD
- ED.CMD
- GENCMD.CMD
- ACOM.DEV

And in User Area 1:

- PRINTEST.COM
- KEYBDTST.COM
- CPUTEST.COM
- RECLAIM.COM
- MEMTEST.COM
- SCRNTEST.COM
- CRC.COM
- PORTEST.COM
- DISKTEST.COM

Each file is explained briefly below. See the preceding sections of this manual for more detailed description of the most important of these command files.

User Area 0 - Standard Operating Commands

CPM-86.SYS

The CP/M-86 system file.

RUN8.CMD

RUN8 is used to run program files with a .COM extension. It simulates the CP/M operating system to allow eight-bit programs to be executed in the sixteen-bit CP/M-86 environment. The RUN8 Simulator program is called into operating by a command of this format:

```
A>RUN8 <program> [RETURN]
```

"Program" is the name of the eight-bit program.

Example: To run Memorite III under CP/M-86:

```
A>RUN8 MEMORITE  [RETURN]
```

CPM-86.BEG

This is the "Begin" file of CONFIG commands which the booting program searches for and executes when the operating system is loaded. The file's contents are variable: Vector Graphic ships its systems with a .BEG file containing a basic set of CONFIG commands, but the file can be edited to suit it to your needs using SCOPE or ED.

UTILITY.CMD

A general-purpose utility that provides three functions: Format, SysGen, and DiskCopy. These functions do the following:

Format

Prepares the hard disk or a diskette for use. Erases existing information, then copies "road signs" (track and sector markings) for use by the operating system.

In the Disk Partition System, Format also sets out partitions for different operating systems to be used from the hard disk.

SysGen

Generates a system onto a newly-formatted disk or diskette, which can then be used to "boot" (start) the system.

DiskCopy

Makes an exact copy of a diskette onto another. The diskette that receives the copy need not be formatted first.

CONFIG.CMD

Informs the operating system of the configuration of your system. Used to (among other things) select printers, specify page size, set baud (transmission) rate for printers and modems, set modes of use for keyboard and video control programs, and create keyboard conversion tables for use with different application programs.

PIP.CMD

Copies files or groups of files (using either unambiguous or ambiguous file names) from one disk or user area to another. Can rename or combine files in copying them, and can also "copy" files to the printer or other output device, and from an input device to an output device.

SERIAL.DEV

This list-device file contains the programming for an optional serial-interface draft printer.

PARALLEL.DEV

This list-device file contains the programming for an optional parallel-interface letter-quality printer.

NEC.DEV

This list-device file contains the programming for the Vector Graphic 7700 and 3500 letter-quality printers.

STORE.COM (Use with CP/M Simulator: RUN8.CMD)

Copies files from hard disk to multiple diskettes for backup. It can copy only files with filenames of seven characters or less. STORE can continue files across more than one diskette, and will copy database (DBS) files. STORE is an eight-bit (.COM) program used with the RUN8 command.

RESTORE.COM (Use with CP/M Simulator: RUN8.CMD)

The "other side" of STORE: copies files back to the hard disk from the diskette or diskettes used for backup in the STORE program. STOREd files extending over more than one diskette must be RESTOREd in the order in which they were copied. RESTORE is an eight-bit (.COM) program used with the RUN8 command.

STAT.CMD

Determines statistical information on the amount and type of storage in disks or diskettes. It can also be used to set files or entire disks to Read-Only (R/O) status, protecting them from changes, and back to Read-Write (R/W), enabling you to again make changes.

HELP.CMD

Provides brief information on the use of all CP/M-86 commands, including the unique Vector Graphic programs, UTILITY and CONFIG. Can be entered alone to produce a display of available topics, or with a command name and options, for information about the specific command.

HELP.HLP

A storage file for information used in producing the HELP-command displays on screen.

ED.CMD

CP/M-86's text editor. Allows you to create and edit a file.

DDT86.CMD

The "Dynamic Debugging Tool," aids the assembly language programmer in locating errors uncovered in the assembly/loading process.

ASM86.CMD

Translates an assembly-language program (with an extension of .A86) into hexadecimal code (with an extension of .HEX). This HEX file in turn is translated by the GENCMD program into machine code (the specific instructions used by the computer to control hardware operation). The result is a command file, with an extension of .CMD.

The life cycle of a .CMD file is as follows:

Assembly language file (PROGRAM.A86) created through ED or SCOPE.

PROGRAM.A86 file is "assembled" through the ASM86 program, by the command:

ASM86 PROGRAM [RETURN]

This produces three files: PROGRAM.HEX (the hexadecimal code file, ready to be translated into machine code), PROGRAM.LST (a printer-ready file, paginated for documentation), and PROGRAM.SYM (contains all symbols used in the program and their relative locations).

The .HEX file is translated into machine code by the GENCMD program:

GENCMD <PROGRAM> [RETURN]

The GENCMD program creates the file PROGRAM.CMD, ready to be used with the command line:

PROGRAM [RETURN]

Any errors in programming uncovered either by ASM86 or GENCMD are indicated to the programmer.

GENCMD.CMD

Uses the HEX output of ASM86 (and other language processors) to produce a .CMD (command) file of machine-executable code. GENCMD is one of the set of commands used to create programs.

TOD.CMD

Allows you to examine and set the time-of-day clock within the system.

SUBMIT.CMD

Allows you to create a "string" (list) of any commands that can be entered from the keyboard to be executed by the operating system automatically. For instance, needed utility programs can be run each morning through creation of a SUBMIT file, using ED or Vector Graphic's SCOPE.

(The job is simple: determine the format for each command line in the routine, enter the commands in your ED or SCOPE file, and give that file the file type .SUB. Then, to run the program, enter this command:

SUBMIT <filename> [RETURN]

The screen displays each line of the SUBMIT file as the command is executed and the results of the program.

ACOM.DEV

ACOM.DEV is the control program, or driver, for communications devices (modems) that can be used to connect your Vector 4 to distant computers systems via the telephone line.

User Area 1 - The RUN8 Simulator and Tests And Diagnostics

User Area 1 contains the RUN8 Simulator program and a series of test and diagnostic programs.

The test and diagnostic programs should be used only by service technicians or personnel experienced in testing and manipulating Vector Graphic computer systems.

PRINTEST.COM (Use with CP/M Simulator: RUN8.CMD)

Allows the technician to check five different functions of the printer attached to the Vector 4 for troubleshooting purposes.

MEMTEST.COM (Use with CP/M Simulator: RUN8.CMD)

Performs a test of the Vector 4's RAM (memory). MEMTEST can be used when the Automatic System Test uncovers a memory error; it can test a specific section of the RAM or the entire range. It automatically determines the maximum memory for both 128k and 256k byte systems. After operation, the system must be rebooted since the operating system is cleared to test the memory.

KEYBDTST.COM (Use with CP/M Simulator: RUN8.CMD)

Allows the technician to check for problems in the keyboard. KEYBDTST can check the value of any individual key or read the values of all keys in sequence from the keyboard.

SCRNTEST.COM (Use with CP/M Simulator: RUN8.CMD)

Used by service technicians to adjust the focus and intensity of the Vector 4 monitor.

PORTEST.COM (Use with CP/M Simulator: RUN8.CMD)

Used to determine the correct operation of the various output ports of the Vector 4.

CPUTEST.COM (Use with CP/M Simulator: RUN8.CMD)

Verifies the CPU by writing a known value into a known location in memory and displaying it on the screen.

CRC.COM (Use with CP/M Simulator: RUN8.CMD)

Uses a mathematical formula to produce a constant value for a file on disk or diskette. If a program is giving errors, the CRC (for Cyclic Redundancy Check) value can be checked against the CRC done for the same program file on a diskette that is known to be good. If the two CRC's do not match, it can be assumed that the program file has been damaged and that hardware failure should not be suspected as the cause of the errors.

DISKTEST.COM (Use with CP/M Simulator: RUN8.CMD)

Checks the accuracy of the disk recording media and the disk controller board. It also can be used by a service technician to align floppy disk drives.

CAUTION

The DISKTEST program may change data on the disk and should be used only by experienced service technicians or programmers. Use of this program by inexperienced operators can lead to erasure of data.

RECLAIM.COM (Use with CP/M Simulator: RUN8.CMD)

Allows an experienced user to rewrite portions of the disk which may become inaccessible due to a "Bad Sector" disk error.

CAUTION

The RECLAIM program may change the data stored in good areas of the disk or diskette and should be used only by experienced operators.

For more detailed information on these diagnostic test programs, see the Vector 4 CP/M Programmer's Guide.

Vector 4 Programmer's Diskette

Although the Programmer's Diskette is part of the **Programmer's Guide to CP/M-86**, a description of its contents has been included here to explain some of the other capabilities available on your Vector 4. The **Programmer's Guide** provides complete documentation of these CP/M-86 utilities for program development and productivity enhancement.

The diskette contains the following files:

ED.CMD
ZSM.COM
LOAD.COM
SERIAL.DEV
PARALLEL.A86
DUMP.ASM
EDIR.COM
RUN8.CMD
RAID.COM

SC.COM
GENCMD.CMD
DDT86.CMD
SERIAL.A86
NEC.DEV
DUMP.COM
ERAX.COM
ACOM.A86

ASM86.CMD
LMCMD.CMD
DDT.COM
PARALLEL.DEV
RANDOM.A86
MBASIC5.COM
SORTDIR.COM
ACOM.DEV

Many of these files also are included on the Vector Graphic CP/M-86 System Diskette and are described above. Those that are not are explained briefly below. See the **CP/M-86 Programmer's Guide** and **Vector 4 CP/M Programmer's Guide** for complete information about these programs. Generally, those files in the list with a **.CMD** extension are described in the first of these manuals; files with a **.COM** extension are described in the second.

SC.COM (Use with CP/M Simulator: RUN8.CMD)

Vector Graphic's SCOPE program (SCreen Oriented Program Editor). It is a sophisticated, programming-oriented text editor offering the many functions required in program development, such as searching for and replacing text, moving sections of text, and appending text from another file. Unlike a true word-processor, the SCOPE program is oriented more towards creating and manipulating text than towards printing it.

ZSM.COM (Use with CP/M Simulator: RUN8.CMD)

The eight-bit (Vector 4 CP/M) version of ASM86.CMD. Translates an assembly-language program (with an extension of .ASM) into hexadecimal code (with an extension of .HEX).

LMCMD.CMD

LMCMD.CMD generates a .CMD file from an Intel HEX-format H86 file.

LOAD.COM (Use with CP/M Simulator: RUN8.CMD)

The eight-bit (Vector 4 CP/M) version of GENCMD.CMD. Uses the HEX output of ZSM (and other language processors) to produce a .COD (command) file of machine-executable code. One of the set of commands used to create programs.

DDT.COM (Use with CP/M Simulator: RUN8.CMD)

The eight-bit (Vector 4 CP/M) version of DDT86.CMD ("Dynamic Debugging Tool"). Aids the assembly language programmer in locating errors uncovered in the assembly/loading process.

SERIAL.A86

The assembly language source code for SERIAL.DEV, the control program for an optional serial-interface printer.

PARALLEL.A86

The assembly language source code for PARALLEL.DEV, the control program for an optional parallel-interface printer.

ACOM.A86

The source code for ACOM.DEV, the driver for communications devices used with the system.

ACOM.DEV

ACOM.DEV is the control program, or driver, for communications devices (modems) that can be used to connect your Vector 4 to distant computers systems via the telephone line.

RANDOM.A86

A sample .A86 (assembly language) program using calls to the BDOS (Basic Disk Operating System) for services.

DUMP.COM (Use with CP/M Simulator: RUN8.CMD)

Produces a hexadecimal listing of any file on disk, including machine executable code used for COM files. DUMP.COM is another tool for assembly language programming. It allow the programmer to see exactly what code is used at any step in program development. To print out a copy of a file, configure the system for the connected printer and enter:

[CTRL P] (to send output to the printer), then

A>**DUMP <filename> [RETURN]**.

DUMP.ASM

Gives an example of what an eight-bit .ASM (assembly language) file looks like. In addition, experienced users can reassemble the DUMP.COM program from the assembly language file included.

MBASIC5.COM (Use with CP/M Simulator: RUN8.CMD)

One version of Microsoft's BASIC-80 interpreter program. MBASIC5 allows you to create and run programs written in the BASIC programming language. MBASIC5 is Version 5.20. Refer to the **BASIC Programmer's Manual** for a complete presentation of the BASIC programming language.

EDIR.COM (Use with CP/M Simulator: RUN8.CMD)

EDIR is a variation on the Built-In command DIR. It allows for several other options, including presenting an alphabetized directory, displaying ambiguous file names for several user areas, displaying the directory at the printer, and displaying the attributes (R/W, R/O, et cetera) on the screen. All options can be combined in one command line.

ERAX.COM (Use with CP/M Simulator: RUN8.CMD)

This is a variation on the resident command ERA. It allows extended (X) options such as erasing all files by individual query (the screen prompts with **Y-N?** for each file before deletion); erasing files from several user areas (using an ambiguous user-area statement); and erasing files set to $SYS through the STAT command.

SORTDIR.COM (Use with CP/M Simulator: RUN8.CMD)

This program sorts the directory into alphabetical order. Whereas the EDIR program **displays** the directory in alphabetical order, the SORTDIR program actually reorganizes the directory entries on the disk for all files in the current user area.

RAID.COM (Use with CP/M Simulator: RUN8.CMD)

Vector Graphic's "RApid Interactive Debugger." RAID checks user-defined programs to eliminate unseen errors.

DEALER'S GUIDE TO THE DISK PARTITION SYSTEM

Introduction

Vector Graphic's Disk Partition System has been designed to allow installation of multiple sixteen-bit operating systems on the hard disk of Vector 4 computers. Up to ten different operating systems may be resident on the disk; one of these may be selected for use from an Options Menu that appears on booting. For system services, the Disk Partition System provides common functional modules that are shared among all installed operating systems.

Initially, the Disk Partition System is offered with the CP/M-86 operating system. The implementation looks forward to subsequent installation of MS-DOS and Concurrent CP/M-86 as these operating systems are made available. In addition, the CP/M-86 system includes a translator program, the CP/M Simulator, that allows programs designed for Extended CP/M and Vector 4 CP/M to be executed under CP/M-86. The Simulator supports use of previously released Vector Graphic software. It also allows most other eight-bit software making conventional use of system functions to be executed in the sixteen-bit CP/M-86 operating system environment.

Details on use of the Disk Partition System, on use of the CP/M Simulator, and on installing CP/M-86 are given in Sections IX and XI. Separate manuals are provided for MS-DOS, Concurrent CP/M-86, and other systems to be provided by Vector Graphic.

Overview

The Disk Partition System was created in response to the need to run different sixteen-bit operating systems in a single hardware environment. One obstacle that must be overcome in a design of this kind is maintenance of conflicting file structures on one disk. The Disk Partition System handles the problem in this way:

- In use of floppy-diskettes, an entire diskette contains only one file structure format, and the diskette may be read from or written to only by an operating system whose file structure is compatible with that of the diskette.

- In use of a Winchester disk, the physical disk is partitioned to allow different file structure formats to co-exist on the same disk. Partitioning is performed at the time the Winchester is formatted (through the FORMAT function of the UTILITY program), and it remains static afterwards.

Hardware Environment

The Disk Partition System is designed to execute CP/M-86 on all Vector 4 hardware configurations offered by Vector Graphic. The Project can be implemented in systems of either 128K or 256K of RAM without modification. The video driver assumes the video map RAM to be at location 3F000H. In systems with 128K of RAM, the second 128K of logical address space is overlaid on the lower 128K, allowing the video controller to utilize the RAM at location 1F000H as the video mapping area.

The Operating System Image

The operating system image for floppy diskette and Winchester disk systems is the same: in both cases, the operating system image is file-based.

The operating system file contains a loader for floppy diskettes, a copy of the operating system, and tables giving parameter information for the supported Winchester and floppy diskette drives. When the operating system is booted, the information about the drives is passed to the operating system from the loader.

System Executive PROM

The Vector 4 Version 2.0 System Executive PROM is responsible for initialization of the system and for booting and also contains information on the configuration of the system's drives. There is a unique PROM for each configuration of the Vector 4 offered by Vector Graphic. This frees the user from having to determine and specify the physical configuration of his hardware when generating the operating system. It also requires that when a drive upgrade is performed, the new configuration must be matched by a new Executive PROM.

A properly configured Winchester disk containing a loader program is automatically booted upon power-up, under control of the executive PROM's monitor program. The Winchester loader then takes control, presenting a menu of selections for booting from all available resources. For systems using CP/M-86, the menu looks like this:

```
[F]         LOAD SYSTEM FROM FLOPPY

[1]         LOAD CP/M-86 FROM LOCAL WINCHESTER
```

For systems with added installed options, the menu might look like this:

```
[F]         LOAD SYSTEM FROM FLOPPY

[N]         LOAD SYSTEM FROM NETWORK

[1]         LOAD CP/M-86 FROM LOCAL WINCHESTER

[2]         LOAD MS-DOS 2.0 FROM LOCAL WINCHESTER
```

If the system has no Winchester drive the system resets and control returns to the monitor for floppy booting.

The PROM's configuration information is passed to the operating system during booting, and the operating system then configures itself to the system's hardware environment.

System Generation

System generation is through the GENSYS function of the UTILITY program. Two kinds of systems may be generated: floppy-diskette based and Winchester-disk based. In both cases, the destination diskette or disk must previously have been initialized using the FORMAT function of UTILITY. For diskette-based systems, the system may be written only to double-sided diskettes.

For both floppy diskettes and Winchester disks, the operating system resides in a file in the data area. For hard-disk systems, the file is within the partition defined for the operating system.

One element of the MS-DOS operating system, the console command processor, also is a file (COMMAND.COM). This file, although separate from the other MS-DOS system records, is copied automatically during system generation.

In floppy-based systems, each of the operating systems implemented in the Disk Partition System has a unique loader program, and it must be stored with the operating system on the system diskette. In contrast, for Winchester-based systems, a general purpose Winchester loader is used for all of the operating systems. It is stored within the FORMAT segment of the UTILITY program.

See Section IX for details on using GENSYS and FORMAT.

Formatting of Diskettes and the Hard Disk

Formatting of diskettes and hard disks is done using the FORMAT function of UTILITY. In addition to the usual format activities, this program allows the user to define partitions on the hard disk for different operating systems and, within those partitions, logical drives.

The user may reserve space for up to ten operating system areas. Each area also is used for file storage for the installed operating system, and the operating system itself is stored in a system file rather than in a system area of the physical disk.

Each operating system may be associated with a unique file structure area, or, if the file structures of several operating systems are compatible, the operating systems may reside in and use one "global" area. For example, CP/M-86 and Concurrent CP/M-86 use file structures that are compatible; these two operating systems could use the same file structure area and access the same physical files. In contrast, MS-DOS has a file structure not compatible with that of CP/M-86, and so it requires a separate file structure area.

During formatting, the user specifies the total size of each area, the identity of the operating system to be installed, and the size and number of logical drive assignments for the hard disk and any existing diskette drives.

Following the FORMAT operation, all information on how the Winchester disk is partitioned and on logical mapping of floppy drives is stored on the Winchester disk. For systems with two Winchester drives supported by two separate controllers, the system should be viewed as having one large, continuous drive. The only constraint is that spanning the two drives within one logical partition is not permitted.

The CP/M Simulator

The CP/M Simulator allows programs written for Extended CP/M and Vector 4 CP/M to be run under CP/M-86. The Simulator makes this possible by converting program calls for CP/M operating system functions into calls for the equivalent CP/M-86 functions.

The Simulator executes the actual eight-bit code in a program, just as would be done under Vector 4 CP/M, using the Z80 processor in the Vector 4. When the executing program makes an operating-system request, the parameters of the request are passed to the sixteen-bit system by a pseudo-BDOS in the Simulator. Control is then passed to the sixteen-bit system, which performs the requested function. The Simulator also contains a pseudo-BIOS that converts eight-bit BIOS calls to sixteen-bit BIOS calls.

Vector 4 CP/M contains many functions not available in CP/M-86, and these functions also are performed by the Simulator.

The Simulator is designed to execute eight-bit software in a manner as true to that of Vector 4 CP/M as possible. This does not mean, however, that any CP/M program will run under the Simulator. Only CP/M programs that make conventional interfaces to the operating system will execute properly. This is the same constraint that has existed with Vector 4 CP/M.

Using the Simulator

The Simulator is a program file under CP/M-86 named RUN8. The Simulator runs automatically when execution of a .COM file is requested. No command line other than that which would normally be used for the .COM file is needed.

Although the user never actively calls the RUN8 program, there are several things that should be done to let RUN8 work most effectively:

1- As shipped, RUN8.CMD is stored on Drive A, User Area Ø, and a copy of it must stay there for RUN8 to work. The file can be PIPped to other drives or user areas, but it must remain in it original location.

2- To be used from other user areas on the drive, the RUN8 file should be set to SYS status through the STAT command:

```
A>STAT RUN8.CMD SYS [RETURN]
```

3- Any .COM files—that is, eight-bit programs written for CP/M—also should be set to SYS status if they are to be run from user areas other than User Area Ø. (See "The STAT Command" in Section X for more information of the SYS attribute. It is recommended that all .CMD and .COM files be set to SYS to make them easily accessible from anywhere in a drive.)

When a command file is requested, the directory is searched first for a .CMD file, then for a .COM file. If the .COM file is found, the Simulator is loaded and goes to work. It saves all existing video and keyboard mode settings on entry, then sets the video command mode to "Vector-3 Compatible" (CONFIG VMODE Ø). This is the mode needed for execution of earlier Vector Graphic software packages. After the eight-bit program is run, the saved mode settings are restored, and the Simulator exits to CP/M-86.

Certain .COM files are not supported or are partially supported under the CP/M Simulator. These are listed at the back of Appendix C, "User's Guide to Troubleshooting the Vector 4."

Under CP/M-86, the Simulator does not support the Vector 4 CP/M functions of spooling and despooling.

INDEX

CP/M-86 HARDWARE

INSTALLATION INSTRUCTIONS

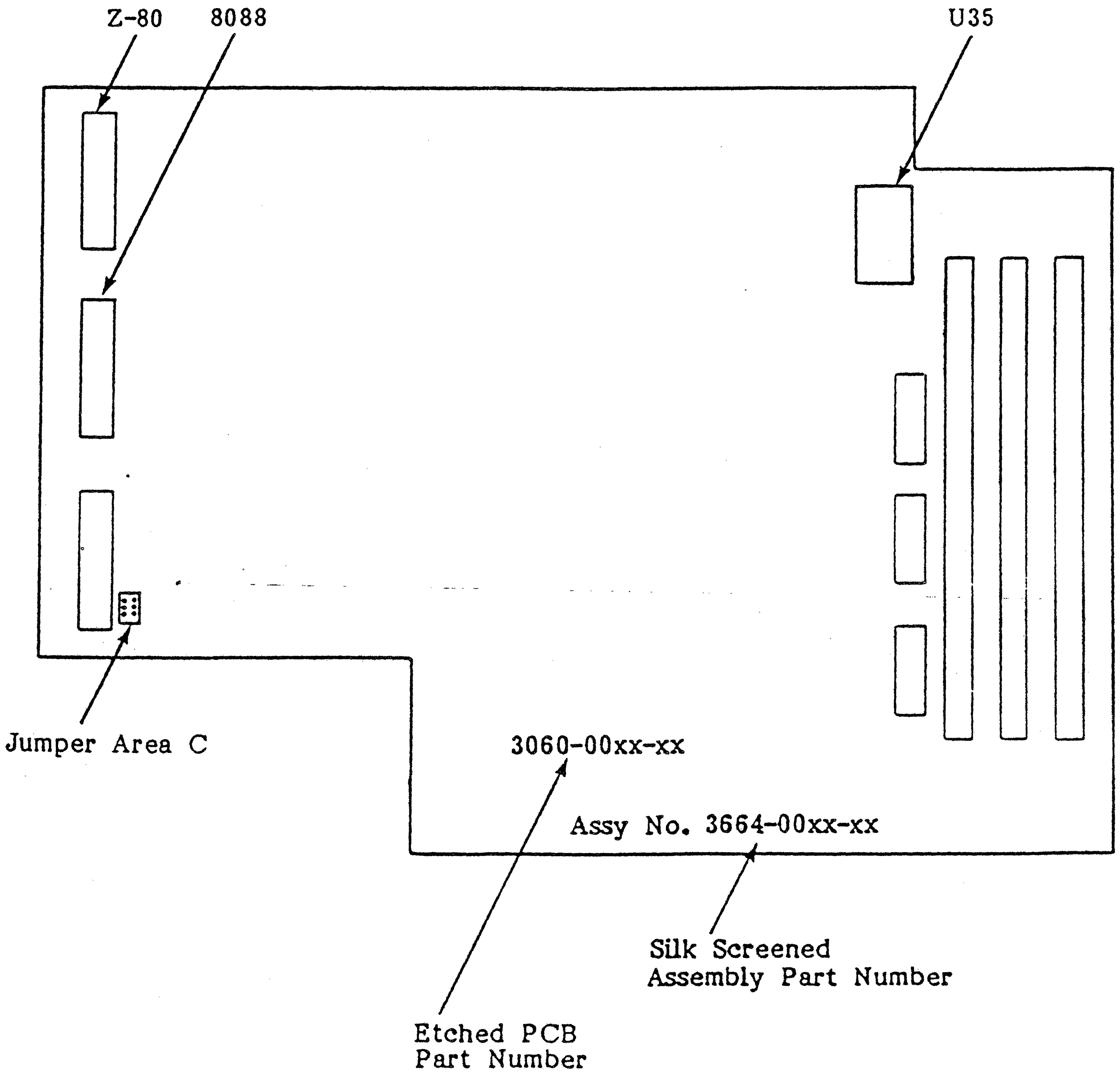
This instruction sheet describes how to modify your Vector 4 so that it can use the CP/M-86 Operating System. If your Vector 4 has already been modified, the PROM will not be included in the installation kit. If this occurs it is not necessary to implement the following procedures.

Tools and Materials Needed:

-One number 2 phillips screwdriver
-One cutting knife.
-One PROM (from your CP/M-86 Installation Kit)

1. Turn off all power to your system and remove the console cover from your Vector 4.
2. Remove the Disk Controller Board and Vector 4 back panel. Be careful not to touch the internal power supply or cables connecting the CRT.
3. Locate Jumper Area C on the SBC. See Figure 1 on next page. Cut the jumpers between pins 3/4 and pins 5/6.
4. Locate PROM socket. See Figure 1 on next page.
5. Remove PROM from U35 and insert new PROM.
6. Check all connections before remounting SBC, securing console cover and turning power back on.
7. Read the CP/M-86 User's Guide for instructions on how to install the CP/M-86 Operating System.

Figure 1: Layout of SBC



Z-80
8088
U35
Jumper Area C
3060-00xx-xx
Assy No. 3664-00xx-xx
Silk Screened
Assembly Part Number
Etched PCB
Part Number