PERQ
Systems
Corporation

PERQ SYSTEM OVERVIEW

March 1984

This manual is for use with POS Release G.5 and subsequent releases until further notice.

PERQ Systems Corporation
2600 Liberty Avenue
P. O. Box 2600
Pittsburgh, PA 15230
(412) 355-0900

The information in this document is subject to change without notice and should not be construed as a commitment by PERQ Systems Corporation. The company assumes no responsibility for any errors that may appear in this document.

PERQ Systems Corporation will make every effort to keep customers apprised of all documentation changes as quickly as possible. The Reader's Comments card is distributed with this document to request users' critical evaluation to assist us in preparing future documentation.

PREFACE

This manual is intended for new users of PERQ and PERQ2. The manual will familiarize you with the machines and get you started using either system.

As you learn more about your system, you will find that it provides you with many facilities, both conveniences and necessities.

This manual consists of three chapters.

Chapter One describes the method to turn the system on and the bootstrap process.

Chapter Two describes the PERQ and PERQ2 hardware.

Chapter Three discusses the basic operation of the system software.

Distinctions between the PERQ and PERQ2 are explicit. Unless qualified, the simple term PERQ refers to both the PERQ and the PERQ2.

CHAPTER ONE

INTRODUCTION

The PERQ and PERQ2 are PERQ Systems Corporation's personal computers which provide an integrated computing system for a single user. A typical configuration of PERQ2 hardware is shown in Figure 1-1.

PERQ provides a large, 32-bit, segmented virtual address space. Virtual addresses are mapped into a 20-bit physical address.

The supplied software provides a functional program development environment. The system includes a text editor, Pascal and optional FORTRAN compilers, interactive stack dump for debugging, file management utilities, and support for micro program development.

When you receive your PERQ system, the hard disk contains a bootable copy of the most recent version of the software system. Successive versions provide software enhancements as well as corrections to some inevitable bugs. Floppy disks serve as the distribution media for the successive software versions.

The extent of the modifications determine the scope of the distribution. If the modifications are major, all users receive an entire new system on floppies. Major modifications change the microcode or enhance the user interface. Modifications that substantially alter the system or impact a large number of modules are also considered as major. Modifications that incorporate minor changes to the system are distributed to the user community in a timely fashion according to the urgency of the changes. Changes in this category include updates to utilities, additional utilities, as well as major changes to a software product that is standalone in nature. By design, minor releases far outnumber major releases.

The software versions are identified by a letter, to indicate the major level, and a number to indicate the minor level. Both the letter and number are successive. For example, version F.1 succeedes version F.0 and software version G.2 supercedes all version F software.

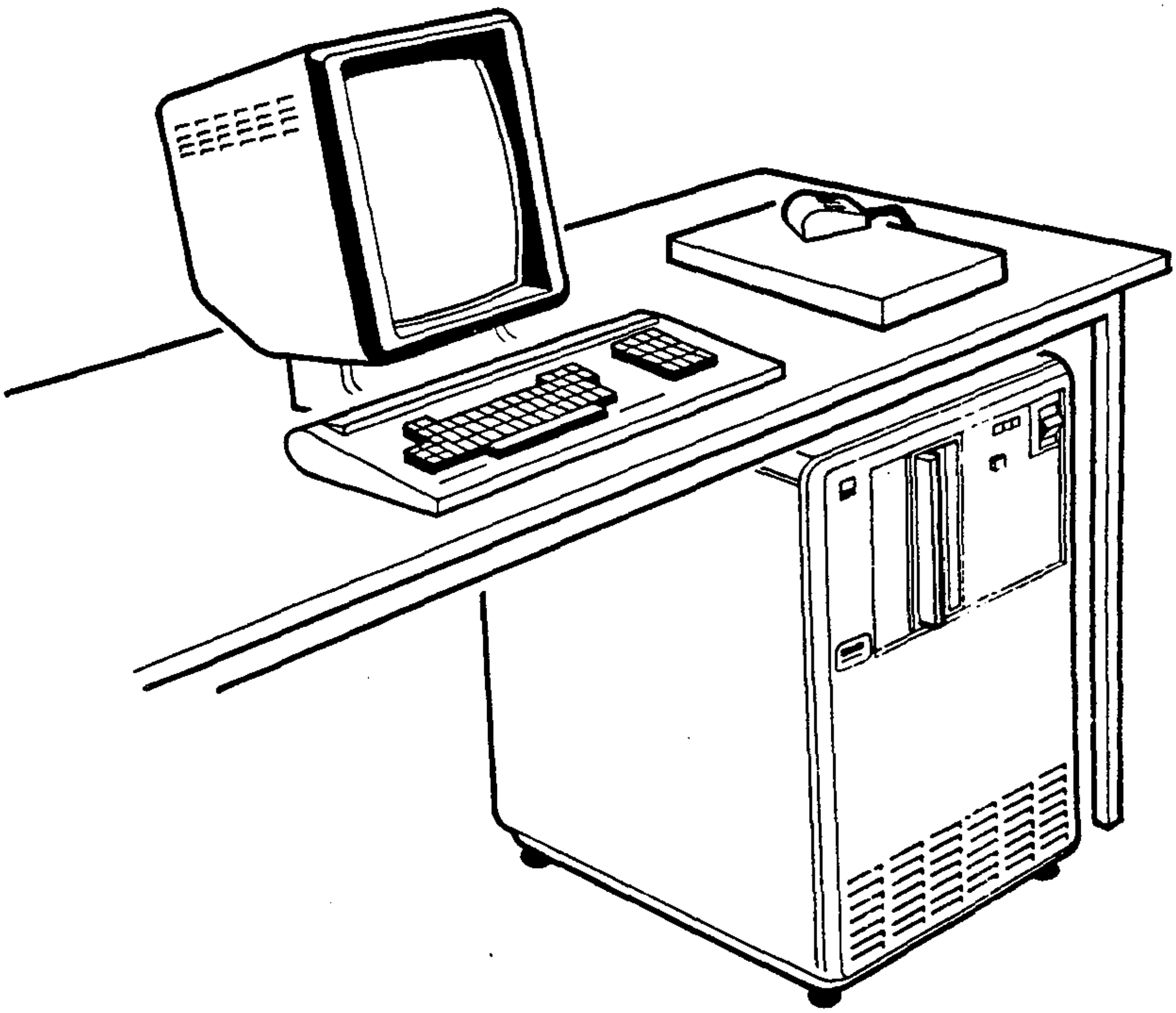

Figure 1-1
PERQ2 System

1.1 POWER ON/OFF

The PERQ power switch, labled ON/OFF, is located on the front panel of the processor box. When you turn on the power, fans start. If you do not hear the fans, check to see that the machine is plugged in and that there is power at the electrical outlet. If the machine is plugged into a live outlet and you still do not hear the fans, call your local service representative.

1.2 BOOTING THE MACHINE

You can boot a PERQ system in one of two ways. First, when the machine is powered up it cycles through the boot sequence. The machine takes about two minutes to boot after it is turned on allowing the disk to spin up to speed. Second, you can boot the machine by pressing the Boot button. On a PERQ, the boot button is centered at the rear of the keyboard. On a PERQ2, the boot button is located either on the right side of the display base or on the processor box, depending on the model you have. In all cases the same sequence of events, described below, occurs.

The boot sequence for a PERQ system has three steps. As each of these steps progresses, the Diagnostic Display (DDS) increments. On the PERQ, the DDS is located on the underside of the keyboard. On the PERQ2, the DDS is located on the processor box or on the base of the display. The DDS shows a three-digit number that provides a code to show progress through the boot sequence. If any step fails, you can look at the Diagnostic Display and see where in the boot sequence the failure occurred. The PERQ Fault Dictionary provides a list of the DDS values and explanations of their meanings.

In the first part of the boot sequence, microcode executes out of a small ROM. This ROM covers the lower 2k of standard control store during this part of the boot sequence. The microcode runs a simple diagnostic on the processor and memory systems. If there are any errors, the microcode halts. The value in the DDS gives the reason that the machine halted. Once these diagnostics complete, the microcode makes a decision about which device is to be used for booting. Currently, there are three possible boot devices.

1. The first choice of boot device is another PERQ or PERQ2. The microcode determines if there is a PERQ Link Board plugged into the I/O Option slot of the machine. If the board is plugged in and there is another PERQ or PERQ2 on the other end of the link, the booting machine waits for commands from the link.

2. The second choice of boot devices is a floppy disk. The microcode checks to see if there is a floppy in the floppy drive. If there is a floppy in the drive and the floppy is a boot floppy, the second part of the boot is done from the floppy.

3. As the third alternative, the microcode tries to boot from the hard disk.

   If all of these fail, then the DDS contains an indication of the error. See the PERQ Fault Dictionary for an explanation of the display number.

After choosing the boot device, the second part of the boot sequence begins. In this part, the machine reads 3k words of microcode from the selected boot device. This microcode is in two sections, a more extensive diagnostic (VFY) and a system boot loader (SYSB). VFY attempts to verify that all of the CPU and Memory systems are working. VFY displays failures on the DDS. If no failures are detected, the microcode loads the PERQ operating system (POS) and transfers control to the third portion of the boot sequence.

In the third portion of the boot sequence, the interpreter microcode performs system initialization and then starts to execute the operating system. POS also increments the DDS. If there were no errors during the boot sequence, the machine will be running and the DDS reads 999.

If the machine has not booted after two or three minutes, try pressing the Boot button. If pressing the boot button does not cause the machine to boot, call your local service representative.

CHAPTER TWO

PERQ HARDWARE

This chapter briefly describes the PERQ and PERQ2 hardware. For further information on your particular machine, see the manuals that were shipped with the machine (Installing Your PERQ and PERQ User Guide).

The standard hardware system consists of a high resolution display, a keyboard, and a pointing device, all connected to the processor box. The processor box houses the processor, a fixed disk, and a floppy disk.

2.1 DISPLAY

The PERQ and PERQ2 displays are free-standing screens, connected to the processor box by a cable about ten feet long.

You can adjust the angle of the display to suit your needs. To tilt the PERQ display, turn the thumbscrew at the rear of the display. On the PERQ2, the display pivots on its base; simply tilt the display to the desired angle. (Note that on some PERQ2 models a boot button and a diagnostic indicator reside on the base of the display. To re-boot or read the DDS, you may have to tilt the display. On later models, these items are located on the processor box.)

The brilliance of the display can be adjusted to suit ambient lighting. On both the PERQ and PERQ2, the control knob is on the back of the display.

The image on the display is known as a raster and consists of 1024 horizontal lines. Each line contains individual picture elements, referred to as pixels.

The standard display, referred to as a portrait display, has its long side vertical and resembles a sheet of 8 1/2 by 11 inch paper. Each of the 1024 horizontal lines contain 768 pixels.

An optional display, referred to as a landscape display, has its long side horizontal and resembles a sheet of 14 by 11 inch paper. Each of the 1024 horizontal lines contain 1280 pixels.

Each pixel can be either black or white. A program can display various shades of gray by varying the proportions of black and white pixels in different areas.

One bit in the machine's main memory is used for each pixel. Note that you can select the background color of the display screen. Refer to the PERQ Utility Programs manual for details. The default is black letters on a white background, but if you prefer, specify white letters on a black background. Thus, depending on the background color, one setting of a bit makes the corresponding pixel white, and the other makes it black. (Module IO_Others.Pas includes the procedure IOSetFunction to control the setting; refer to the POS Operating System Manual for details.)

The area of memory containing these bits is known as the bit map. Since the bit map is simply a part of main memory, the whole of the bit map is directly accessible to user programs. For a portrait display, the bit map is 96K bytes. For a landscape display, the bit map is 160K bytes.

The display controller (an integral part of the memory board) transfers the bit map to the display 60 times per second, referred to as a 60Hz refresh rate. The refresh system is non-interlaced and thus provides a genuine 60 complete frames per second. This provides a flicker free display.

The processor contains an instruction called RasterOp, implemented by special hardware, that enables programs to quickly and easily alter the bit map. In fact, RasterOp makes real animation possible.

The Q-code RASTER-OP and the corresponding Pascal statement RASTEROP merge a section of the bit map, which corresponds to a rectangular section of the display (the destination), with an identically sized bit pattern (the source) which is often held in another part of main memory.

The first parameter to RASTEROP is the RASTEROP function. There are eight RASTEROP functions, of which RRpl (function 0) is probably the most commonly used. Function RRpl causes the source to entirely replace the destination. Most of the other functions cause various mergers between the source and the destination bit patterns. Refer to the PERQ Pascal Extensions manual for a complete description of RASTEROP and the RASTEROP functions.

For a portrait display, the first 48 words constitute the top line of pixels. For a landscape display, the first 80 words constitute the top line of pixels. The most significant bit (bit 15) of each word is the first pixel. The following diagram illustrates the pixels in the top left corner of a portrait display:

Line	Words(bits)
0	0(15,14, ... 1,0) 1(15,14 ... 1,0) 2 ... 47
1	48 ... 95
2	96 ...

The displays can also provide different fonts.

2.2 KEYBOARD

The PERQ and PERQ2 use a detached, serial interface terminal keyboard. The PERQ keyboard connects to the processor box by a cable about eight feet long. The PERQ2 keyboard connects to the display by a coiled cable.

The keys are all solid state, including the LOCK key which contains a red light that is lit when the LOCK is on. On the PERQ keyboard, the LOCK key functions like a shift lock--the key shifts alphabetic characters and numerals to their upper case equivalents. On the PERQ2 keyboard, the LOCK key functions like a caps lock--the key shifts only the alphabetic characters to upper case.

All keys on the keyboards provide auto-repeat after the key has been held down for about half a second. On the PERQ, the repeat continues at the rate of about 10 per second. On the PERQ2, the repeat continues at the rate of about 16 per second. N-key rollover ensures that a keystroke is not lost even if you have not released the previous key.

The keyboard generates ASCII characters.

The keyboard employs standard typewriter layout with the addition of special keys. Figure 2-1 depicts the PERQ keyboard and Figure 2-2 depicts the PERQ2 keyboard.

2.3 POINTING DEVICE

The PERQ and PERQ2 use an electromagnetic tablet, referred to as the EM tablet, and three-button puck as a pointing device. Figure 2-3 depicts the standard three-button pucks. Optionally, you can choose a Summagraphics bitpad, referred to as the bitpad, and four-button puck. Figure 2-4 depicts the four-button puck supplied with the optional bitpad.

In the following paragraphs, the term tablet refers to both the EM tablet and the Summagraphics bitpad.

The puck position is read when the puck is in contact with the tablet. The system displays a cursor on the screen to depict puck position. The standard shape of this cursor is a small arrow pointing up and left, but you can redefine its shape. Note that this cursor is not placed in the bit map, but is superimposed on the display by the display controller at each refresh.

When you move the puck on the tablet, this cursor moves to follow the puck. To point the cursor to some area on the screen, look at the screen and move the puck. When you press a puck button, a small switch closes and causes an action. Different programs use the puck buttons in different ways. For example, the Editor assigns distinct functions to each button and performs an action dependent on the button you press. Other programs do not distinguish between the buttons. These programs perform the same function regardless of the button you press. Read the specific program documentation for information on how a given program uses the tablet and puck buttons.

The puck position can be read in one of two modes: absolute mode or relative mode. In both modes, the cursor on the screen follows the movement of the puck while it is in contact with the tablet.

In absolute mode, cursor position on the screen is determined by the actual (absolute) tablet coordinates of the puck; if the puck is in the upper-left corner of the tablet, the cursor is in the upper-left corner of the screen.

In relative mode, cursor position is determined by the difference between previous and present tablet coordinates; the absolute position of the puck does not matter. Motion of the puck on the surface of tablet causes motion of the cursor on the screen in the same direction and with proportional distance. Thus, if you move the puck left, the cursor moves left. When you stop the puck, the cursor stops. However, you may slide the puck all the way to the edge of the tablet and find that the cursor is only halfway across the screen. To move the cursor further left, raise the puck from the surface, retract it to the opposite side, replace it on the surface, and slide it in the original direction again.

Relative mode has several advantages over absolute mode. The total extent of arm movement is less in relative mode and programs can control the ratio of puck movement to cursor movement in relative mode. Thus, you can have very fast movement with coarse positioning, or very fine movement with highly accurate positioning.

Most programs let you specify the tracking mode.

2.4 PROCESSOR BOX

This section describes the components contained in the processor box. Figure 2-5 presents a conceptual diagram of the components.

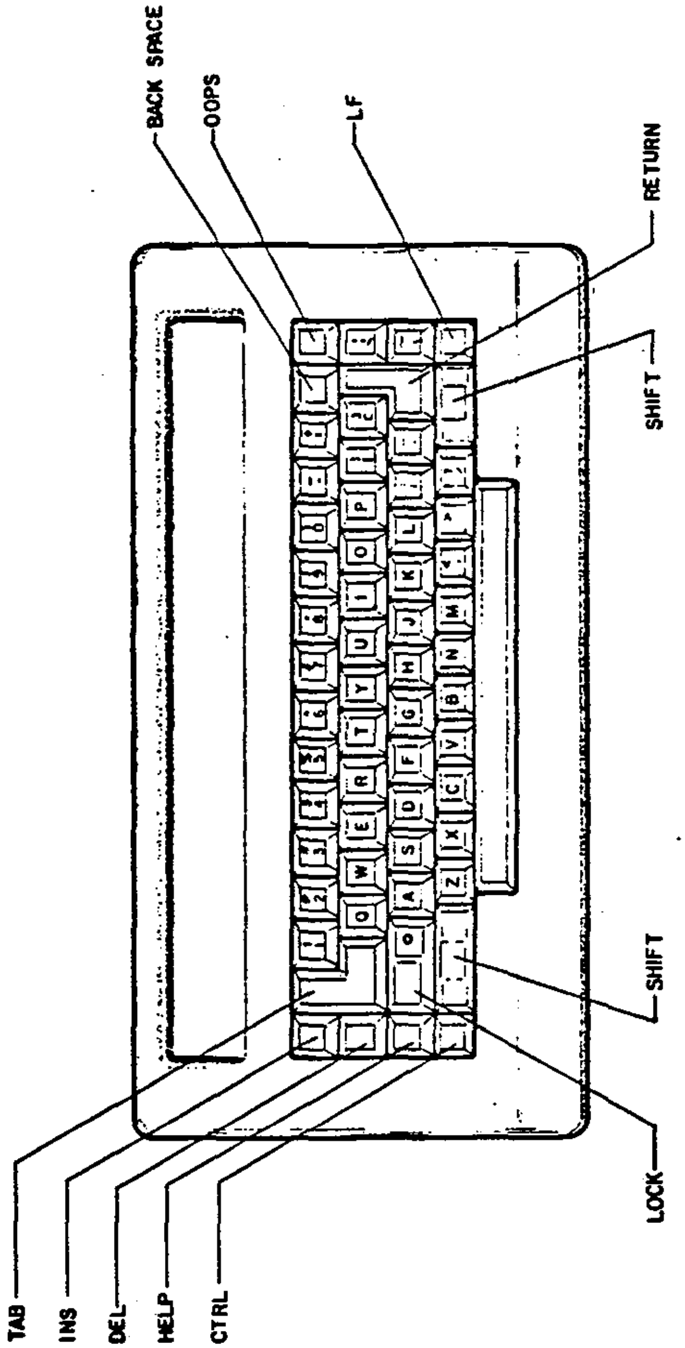

Figure 2-1
PERQ Keyboard

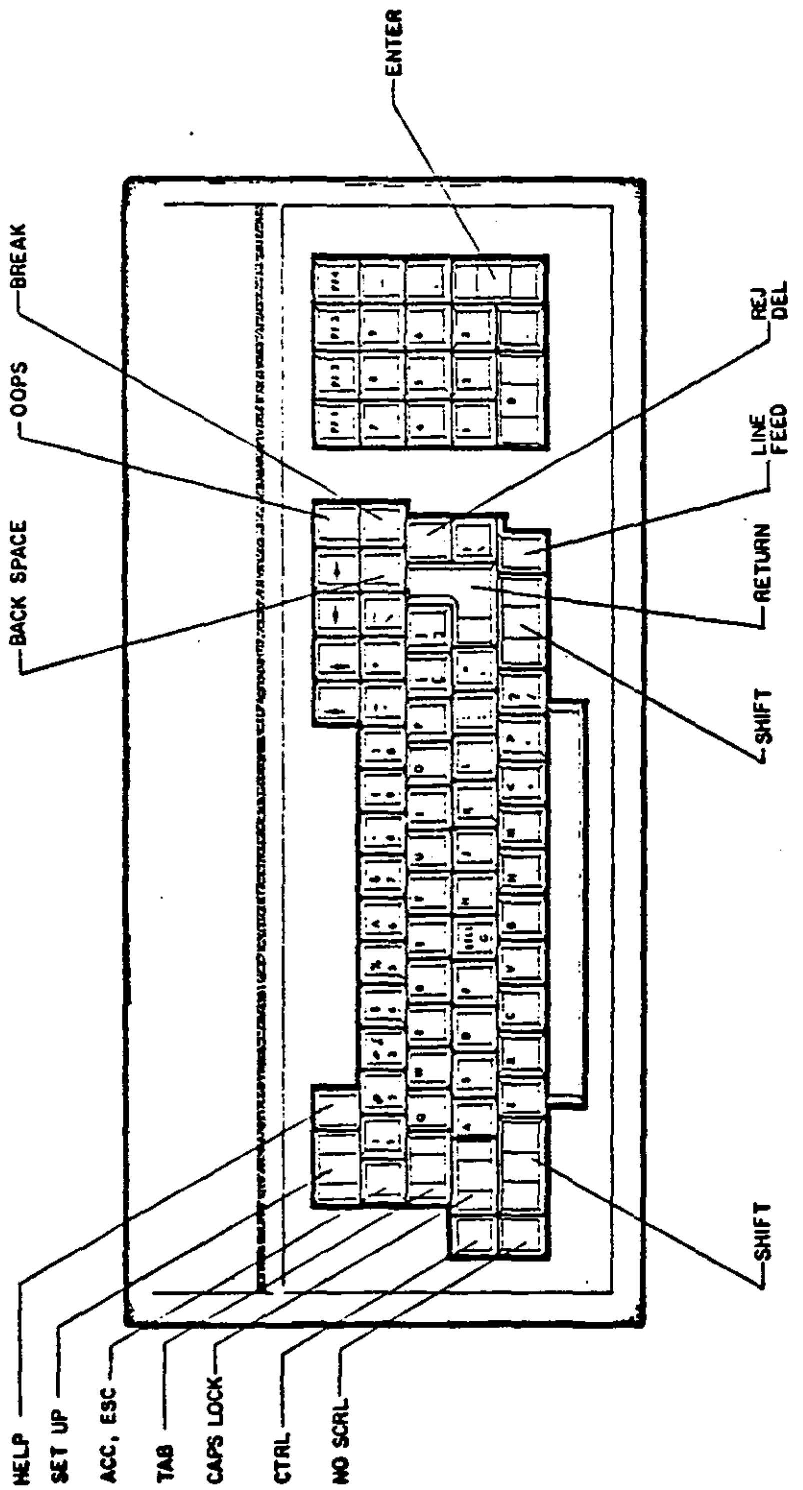

Figure 2-2
PERQ2 Keyboard

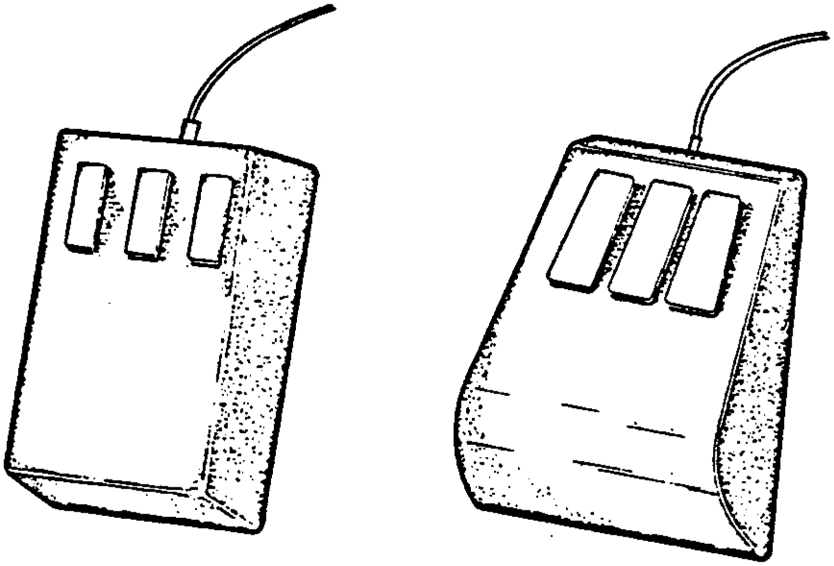

Figure 2-3
Three-Button Puck

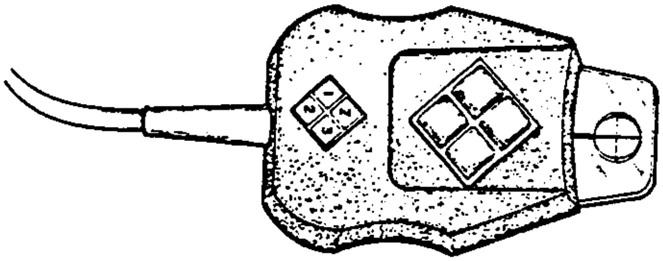

Figure 2-4
Four-Button Puck

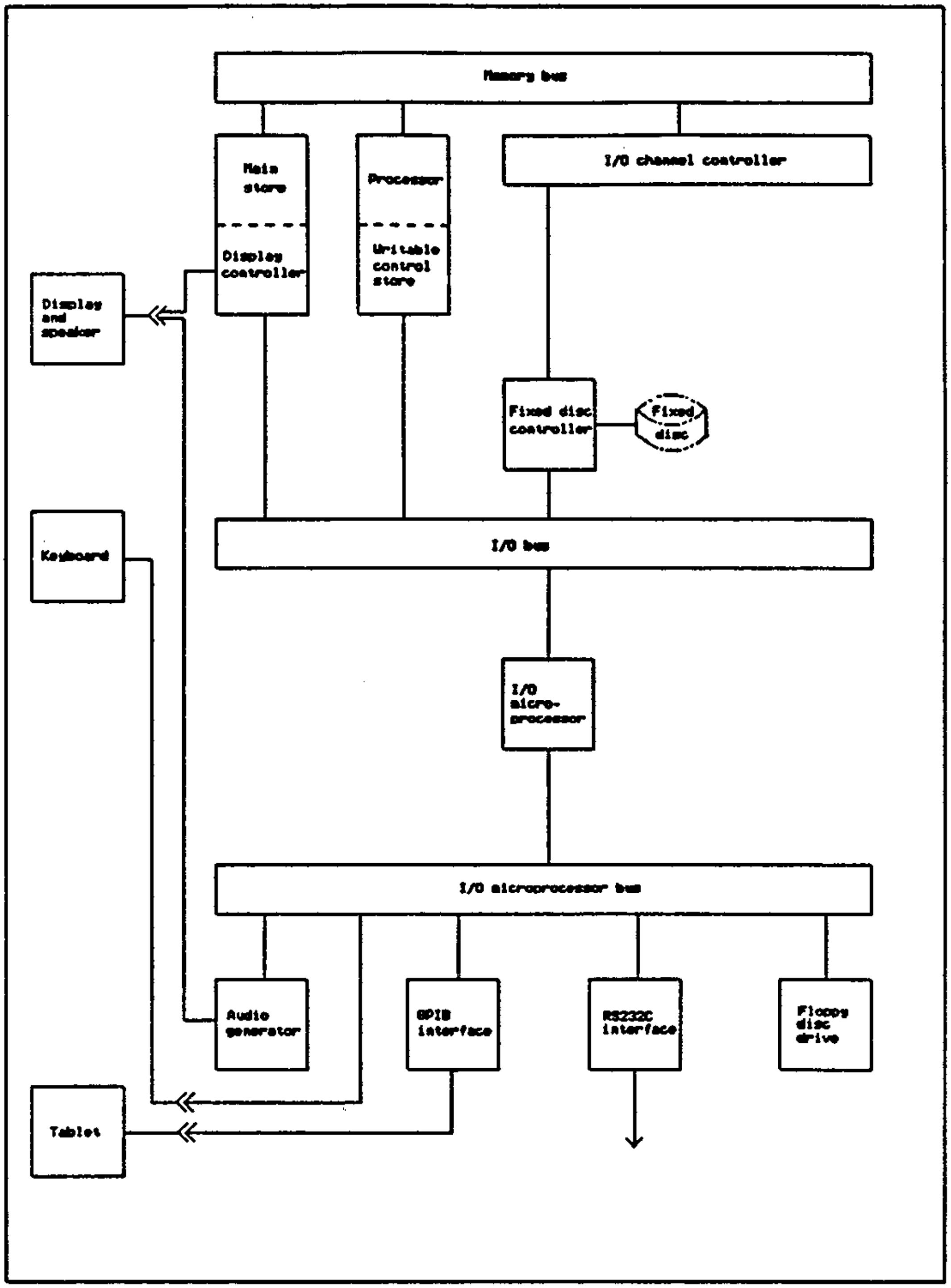

Figure 2-5
Components of Processor Box

2.4.1 Central Processor Unit

The processor is controlled by the PERQ Operating System (POS), which is written is Pascal. You can write programs to run under POS in Pascal or in POS FORTRAN 77.

Like most high-level languages, both Pascal and POS FORTRAN 77 programs must be translated into coded instructions (often called machine code) before they can be executed. A code called P-code has been devised for standard Pascal programs and is also suitable for FORTRTAN.

The PERQ and PERQ2 machine code is known as Qcode. Qcode is P-code with some additions to cater to extra instructions which facilitate display manipulations such as RasterOp. The processors can execute Q-code at up to 1 million Q-codes per second.

The PERQ and PERQ2 processors are actually microprogrammed to execute Q-code, whereas a conventional processor contains hardware that directly executes the machine code. On a microprogrammed processor, that hardware is replaced by a microcode engine controlled by a microprogram. The microprogram is written in microcode which the microcode engine executes directly. Figure 2-6 illustrates this concept. The net effect is that the whole appears to be a processor capable of executing the machine code (Q-code), even though the microcode is different from the machine code.

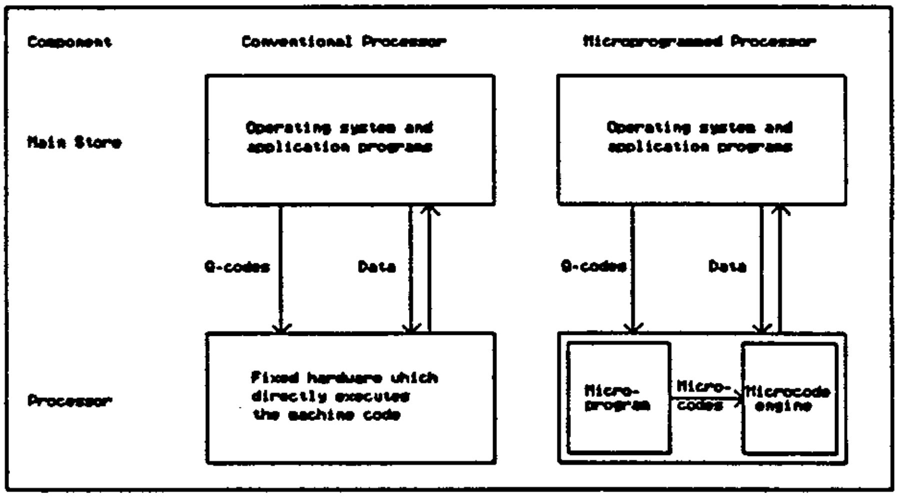
Component
Conventional Processor
Microprogrammed Processor
Main Store
Operating system and application programs
Operating system and application programs
Q-codes
Data
Q-codes
Data
Processor
Fixed hardware which directly executes the machine code
Micro-program
Micro-codes
Microcode engine

Figure 2-6
Microcode Executed by Microcode Engine

A microprogrammable processor provides several advantages. If Q-code is enhanced, a new microprogram enables an existing PERQ or PERQ2 to execute the enhanced Q-code without hardware alterations. If a completely different machine code proves desirable for some application, a new microprogram could implement the new machine code. Also, a microprogrammed machine permits the transfer of some of the hardware controller's complexity into the microprogram and thus makes the controller simpler, less expensive, and more reliable.

The microprogram is loaded during boot into Writable Control Store (WCS) contained on the processor board. The standard WCS consists of 4K words. An optional WCS consists of 16K words. Each WCS word is 48 bits long.

The WCS cycle time (that is, the time to transfer one word from the WCS to the processor) is 170 nanoseconds.

Data comes into and out of the processor on 16 bit wide paths. Internally, the processor uses paths that are 20 bits wide.

The PERQ and PERQ2 processors have:

> 256 REGISTERS - these are 20 bit wide general purpose registers, which can hold main memory words or addresses
>
> OP FILE - this is an 8 byte buffer of Q-codes and parameters. It is loaded by copying the appropriate part of a code segment from main memory. Thus, the OP FILE gives the processor quicker byte by byte access to the current part of the program than if it had to access each byte separately from main memory.
>
> EXPRESSION STACK - this is a push-down stack containing 16 registers, each 20 bits, and is used for evaluating expressions by holding intermediate values.

2.4.2 Main Memory

The standard PERQ main memory consists of 0.5 megabyte of Randomly Accessible Memory (RAM). Optionally, the PERQ can use a full megabyte of RAM.

The standard PERQ2 main memory consists of 1.0 megabytes of Randomly Accessible Memory (RAM). Optionally, the PERQ2 can use 2.0 megabytes of RAM.

A single word (two bytes) of memory can hold an address in the range 0 through 65535 (that is, 0 through 64K-1). Thus, no single piece, or segment, of a program can be larger than 64K words, since the program cannot address beyond this size. If a single data area larger than 64K words is required, it must be constructed from more than one segment.

There are two types of segments:

CODE SEGMENTS - contain only Q-codes and their parameters. Once a program is running, its code segments are fixed in size and can only be read. Code segments are byte-addressed and therefore have a maximum size of 64K bytes (32K words)

DATA SEGMENTS - contain data only, no Q-codes. Data segments can be read, written, and varied in size while the program executes. Data segments are word-addressed and therefore have a maximum size of 64K words.

A single word can also be used to hold a segment number. Thus, there can be up to 64K segments. Segments are allocated in 256 word groups and loaded contiguously into main memory on 256, word boundaries.

When more software is required than will fit in main memory, that software can be loaded into virtual store. Virtual store is partly the hard disk and partly main memory. Those parts of the software that need to be available at all times are kept in memory; the parts that are in use are kept temporarily in memory; and those parts which are not in use are kept on the hard disk.

When a segment that is on disk is required, it is copied or swapped into memory. Segments that are not in use are swapped out to make room. The swapping system only updates the disk copy of a swapped out data segment that has been written to, since the disk copy for other segments is accurate.

2.4.3 Input and Output Control

The PERQ and PERQ2 have two systems for handling data: the Input Output (I/O) Channel Controller for fast data streams and the I/O microprocessor for slower data streams.

The I/O channel controller has direct access to the main memory. This controller is attached to the hard disk.

The I/O microprocessor controls the slower data streams keyboard, floppy disk drive, GPIB interface, and RS232 interface.

2.4.3.1 Hard Disk

Each PERQ and PERQ2 is equipped with a high-speed, non-removable hard disk. On the PERQ, the standard disk is a Shugart 24 megabyte disk. On the PERQ2, the standard disk is either a Micropolis 8" 35 megabyte or a Micropolis 5.25" 35 megabyte disk (check the <u>User</u>

Guide that accompanied your machine).

The disks are mounted in the processor box. The disks take approximately two minutes on a PERQ and 30 seconds on a PERQ2 to spin-up and stabilize following power on.

2.4.3.2 Floppy Disk

Each PERQ and PERQ2 is optionally equipped with an IBM-compatible floppy disk drive. The drive uses an 8-inch, double-sided, single or double density diskette. The capacity of the diskette is 1 megabyte (double-sided, double-density).

You can store data on the diskettes in one of two formats: the POS file system format, which is not compatible with any other system, or RT-11 format, which is compatible with PDP-11 and VAX floppy disks. (RT-11, PDP-11, and VAX are registered trademarks of Digital Equipment Corporation). Note that a diskette formatted for RT-11 use is not interchangeable with a diskette formatted for POS file system use and vice versa.

The primary purposes of the floppy disk systems are the receipt of software and the exchange of programs and data with other systems.

On the PERQ, the floppy drive is mounted horizontally at the top of the processor box. Insert the labeled side of the floppy at the front of the drive with the label facing up.

On the PERQ2, the floppy drive is mounted vertically at the top left of the processor box. Insert the labeled side of the floppy at the front of the drive with the label facing right.

2.4.3.3 GPIB

Each machine is equipped with an IEEE-488 interface, known as the General Purpose Instrumentation Bus or GPIB. The GPIB is an 8-bit parallel, medium speed, industry standard bus. You can connect peripherals (for example, tape drives, printers, and plotters) to a PERQ or PERQ2 via the GPIB. The GPIB can also acquire data from and control many types of laboratory instrumentation.

The GPIB interface can perform Talker, Listener, and Controller functions.

2.4.3.4 RS232

The PERQ and PERQ2 are equipped with a fully programmable, serial line interface conforming to RS232 standards. Baud rate, character size, parity options, and modem control are all programmable.

The PERQ uses a single RS232 port capable of transfer rates up to 9600 baud. The PERQ2 includes two RS232 ports (port A and port B) capable of transfer rates up to 19.2K baud. However, system software uses only one RS232 port, port A.

2.4.3.5 ETHERnet

You can optionally equip a PERQ or PERQ2 with a Local Area Network interface conforming to Ethernet standards (Ethernet is a registered trademark of the Xerox Corporation). This optional interface also conforms to ECMA and IEEE-802 standards.

The Ethernet interface provides high speed interconnection of computers at 10 megabits per second along a coaxial cable. Up to 2.5 kilometers of station separation is possible and up to 1024 stations can share the network.

On the PERQ, a separate board is required for Ethernet support. On the PERQ2, Ethernet support is included on the IO board. A multiple pair cable connects the interface to a transceiver (a small box usually located in the ceiling), which in turn connects via a tap to the coaxial cable.

The network uses a baseband transmission with Carrier Sense, Multiple Access/with Collision Detect (CSMA/CD).

To use an ETHERNET connection, you assign an Ethernet node name for each machine on the network in the file ETHERNET.NAMES. Each machine on the network must include the file ETHERNET.NAMES. Refer to the FTP command description in the PERQ Utility Programs Manual for details.

CHAPTER THREE

SYSTEM OPERATION

The software supplied with the PERQ system provides a functional program development environment. The system includes a text editor, a Pascal compiler, an interactive stack dump routine for debugging, file management utilities, and support for micro program development. An optional FORTRAN 77 compiler is also available.

The sections that follow discuss the basic operation of the PERQ Operating System (POS) software. For complete details on an individual software entity, consult the specific documentation.

3.1 USING THE KEYBOARD

You communicate with the PERQ Operating System, POS, by entering commands at the keyboard. The keyboard sends the commands to the command line interpreter, known as the Shell, which either executes the commands or calls a system or user written program to execute the command.

Both the PERQ and PERQ2 keyboards employ standard typewriter layout with the addition of special function keys. The special function keys are in different positions on the two keyboards and are not labeled identically (refer to Chapter 2), but the keys have the same effect on various programs.

Some of the special function keys that appear on the keyboards are described below. Consult the documentation for the system program you are using for information on a how a particular program uses each of the special function keys.

The PERQ and PERQ2 also recognize a number of special control characters. These control characters perform simple input line editing and program control. You can enter a control character by pressing the appropriate letter while holding down the CTRL key. All of the PERQ documentation depicts a control character as CTRL/x where x is the variable letter key. POS responds to some control characters by displaying an up arrow (^) followed by the letter you typed (for example, ^C). Depending on their function, other control characters are not echoed.

The following lists the valid control characters and some of the special function keys:

BACK SPACE or CTRL/H - erases the last character you typed.

CTRL/W or CTRL/BACK SPACE - erases the last word you typed.

OOPS or CTRL/U - erases the last line you typed.

CTRL/C - typed once, causes a current program to abort the next time the program asks the operating system for input.

CTRL/C - typed twice, causes the current program to abort immediately. However, this does not cause user command files to exit; see CTRL/SHIFT C.

CTRL/S - suspends program output to the display.

CTRL/Q - resumes output to the display after a CTRL/S.

CTRL/SHIFT C - displays a dump of the runtime stack and immediately returns to the Shell. During command file execution, CTRL/SHIFT/C aborts the command file and returns control to the keyboard.

CTRL/SHIFT D - displays a dump of the runtime stack and allows you to enter the preliminary debugger (called Scrounge). If you choose not to enter the debugger, the original program resumes execution. If you do enter the debugger, you can request that the program resume after investigating the state. For a more complete description of the debugger, see the "PERQ Utility Programs Manual."

CTRL/SHIFT P - dumps an image of the screen to a printer connected to the Perq directly or via the Ethernet network.

HELP - pressing the HELP key by itself displays a general help message. If you press the key after typing a command, the display contains specific help information for the command.

3.2 ISSUING COMMANDS

The POS command interpreter runs as a separate user program. The name of the command interpreter is the Shell. It takes commands from the terminal or from a user command file and executes them. The Shell does not distinguish between upper and lower case letters; use whichever you prefer. The commands in a user command file can be any sequence of commands as would be typed to the Shell at the keyboard. However, the cursor is a lighter shade so that you can distinguish command file execution from a typed command. The PERQ Utility Programs manual describes both the Shell and the use of command files.

The general form of a command line is a command or program name, followed by any number of optional parameters, followed by any number of optional switches. Some of the switches take parameters. For example:

```
     PRINT myfile/Copies=5
```

prints five copies of a file named myfile on a printer. All switches begin with a "/".

An example of a program that takes a number of parameters on the command line is the Copy program. The form is:

```
     COPY SourceFile~DestinationFile
```

Print is a good example of a program that takes multiple switches. A Print command line might be:

```
     PRINT myfile/Copies=5/Break/Title/Shift=8
```

The Shell and some other utility programs allow you to enter commands and arguments by means of PopUp Menus. (This only works, however, if you have booted from the hard disk.) The menu holds a list of all legal commands or arguments. You use the pointing device, see Chapter 2, to specify the selection. Using a menu is sometimes more convenient than typing out the name of the desired command or argument.

PopUp menus appear when requested. When the selection is made, the menu disappears and restores the screen space that was covered by the menu. Using PopUp menus, therefore, does not require sacrificing any screen area. The PERQ Utility Programs manual provides complete details on the use of PopUp menus.

3.2.1 Logging On

After the PERQ system boots, you enter the date and time of day. The date and time use the form:

```
DD MMM YY  HH:MM:SS
```

When you specify the date, you cannot enter a year prior to 1980 (80). The time notation uses a 24-hour clock. If you were logging in on the 27th of March, 1983 at 2:30 in the afternoon you'd type

```
27 mar 83 14:30:00
```

The seconds are optional and you can replace the spaces in the date with hyphens (27-Mar-83). There is usually a default date and time given at login. If the date is accurate, enter only the time or, type a carriage return to confirm both date and time. If it is not accurate or if there is no default, enter the correct date and time.

At this point, you must supply a user name and password to be allowed access to the machine. This is referred to as logging on. After the PERQ system loads, the defined user names are "Guest" and "" (the empty string; just press the Carriage Return key). The password for both names is the empty string. Refer to the PERQ Utility Programs manual for information on the UserControl program and instructions on how to add other user names and associated passwords.

After you successfully log on, the command interpreter loads and you can execute commands.

3.2.2 Logging Off

Before turning the machine off, you should always log off using the Bye command, also described in the PERQ Utility Programs manual. If you simply power down, temporary files on the disk are not deleted. You can type:

```
BYE OFF
```

to log off and power off the machine automatically. If this is not possible, you can always turn the machine off by moving the power switch to the left.

3.2.3 Help

Help is available for all utilities. The system supplies Help through the Helper program. You can invoke Helper by typing help as a command, by pressing the HELP key, or by pointing to the Help command from a PopUp menu. Helper then prompts for the help topic. You can also invoke Helper by typing the Help command and supplying a topic, for example

```
Help Edit
```

invokes the Helper and provides help on the Editor.

3.3 SPECIFYING A FILE NAME

The route that the system travels to reach a filename is called a PATH. A full path name specifies device, partition, directories, and filename, in that order. The syntax of a path name is:

```
device:partition>[directory>]filename
```

The brackets surrounding the "directory" part indicate that there may be zero to nine occurences of ">directory". Note that if you specify a directory, the square brackets are NOT part of the syntax.

There are four methods to specify a filename. First, you can specify a full path name. For example:

```
device:partition>[directory>]filename
```

Second, you can use the default device that was determined at boot time. For example:

```
:partition>[directory>]filename
```

Using this syntax enables you to look for a file in a different partition.

Third, you can use the default device and partition that were set at boot time. For example:

```
>[directory>]filename
```

The search then starts at the first directory you specify.

Finally, you can just type:

```
filename
```

and the search begins at the current directory (which you can set using the PATH command, refer to the PERQ Utility Programs manual for details). The PATH command can use any device, any partition, and any directory in that partition.

Setting the default path doesn't affect the default device and partition used in the first three methods.

In parallel with the concept of a current directory, the file system provides a search list. This provides you the ability to specify a set of directories to search, in a specific order, whenever you simply specify a filename. Refer to the SetSearch command description in the PERQ Utility Programs manual for details.

3.4 PREPARING PROGRAMS

There are four steps in preparing a program to run on a POS system:

Create a source file

Compile the source file to produce a segment file

Link the segment file to create a .Run file

Run the program

Each of the steps involves manipulation of at least one file.

3.4.1 Creating a Source File

The PERQ text Editor is a "point, act" Editor; to perform an editing action, you first "point" to a piece of text in the file and then perform some action on it. Pointing is done with the tablet and puck.

Refer to the PERQ Editor User's Guide for details on the use and operation of the Editor.

3.4.2 Compiling a Source File

The compiler converts a source program into a file of Q-codes.....

The PERQ Pascal Extensions manual describes PERQ Pascal. For a description of the optional FORTRAN compiler, refer to the PERQ FORTRAN User's Guide.

The PERQ Utility Programs Manual describes both the Link and Run commands.

3.5 TAILORING THE SYSTEM

You can use Profiles to tailor your PERQ system. The Profile is a text file which contains commands that define characteristics of certain utility programs. For example, a profile can direct the Login program to initialize the default path and searchlist. Each user of the system can have his own profile; the UserControl program can assign each user a profile file.

To create a profile file, copy DEFAULT.PROFILE to your own directory. For example:

```
COPY SYS:BOOT>DEFAULT.PROFILE SYS:USER>MYDIR>MYPROFILE
```

Now run UserControl and specify SYS:USER>MYDIR>MYPROFILE for the profile file. You can then edit the file and establish your specific conventions.

Each entry in the profile file begins with a number sign (#), followed by the name of the subsystem. For example, #LOGIN. The subsequent lines contain switches or data for the subsystem. The general format of a profile file is as follows:

```
#program1 <switches or input for program1>
          <more data for program1>
#program2 <switches for program2>
  ...
```

For example:

```
#Login /Path=Sys:User>Mydir
       /SetSearch=Sys:Boot>Library
       /CursorFunction=7
 RandomUtility /MaxSize=100
```

The format of each list of entries in the profile is defined by the utility program that uses the profile. You should read the documentation for a particular utility to determine whether it reads the profile, and if so, what entries can be included in the profile.