

# HEWLETT-PACKARD JOURNAL

AUGUST 1984



# HEWLETT-PACKARD JOURNAL

August 1984 Volume 35 • Number 8

## Contents

- 4** Touchscreen Personal Computer Offers Ease of Use and Flexibility, by Srinivas Sukumar *Is it the easiest to use? Perhaps. Is it different? Yes.*

- 6** Operating System and Firmware of the HP 150 Personal Computer, by Laurie E. Pollero Wood and Charles H. Whelan *The industry standard MS-DOS operating system makes available a large amount of software.*

- 11** The HP 150 Touchscreen: An Interactive User Input Device for a Personal Computer, by Peter R. Stratton, Scott R. McClelland, and Thomas E. Kilbourn *It has adequate resolution, doesn't degrade the display, and is reliable.*

- 15** Applications Software for the Touchscreen Personal Computer, by Peter S. Showman, Karl W. Pettis, Karlie J. Arkin, Jeffrey A. Spoelstra, John Price, W. Bruce Culbertson, and Robert D. Shurtleff, Jr. *The objective is to make the computer less forbidding.*

- 25** Hardware Design of the HP 150 Personal Computer, by John E. Watkins, Patricia A. Brown, George Szeman, and Susan E. Carrie *It's really two products—a computer and a terminal.*

**28** Software Graphics in the HP 150

- 30** Personal Computer Printer Is User Installable, by Joseph D. Barbera *You just drop it into the top of the HP 150.*

**32** Authors

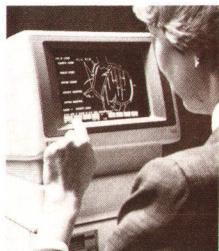
- 34** A Standard Keyboard Family for HP Computer Products, by Lorenzo Dunn and Michael R. Perkins *Its design was guided by many ergonomic studies.*

---

Editor, Richard P. Dolan • Associate Editor, Kenneth A. Shaw • Art Director, Photographer, Arvid A. Danielson • Illustrators, Nancy S. Vanderbloom, Susan E. Wright  
Administrative Services, Typography, Anne S. LoPresti, Susan E. Wright • European Production Supervisor, Michael Zandwijken • Publisher, Russell M. H. Berg

---

## In this Issue



We shot this month's cover photo over the shoulder of Diana Jillie, an HP medical sales representative, as she was demonstrating how to use the HP 150 Touchscreen Personal Computer to enter coronary angiography results. Seeing this demonstration convinced me that there are applications in which touching the screen of a computer is the *best way*—not just *another way*—to interact with the computer. Here's how *Advances for Medicine*, a publication of HP's Medical Products Group, described the application.

"First, by using a finger to touch the appropriate areas on the screen, the user modifies a generic coronary 'tree' diagram so that it accurately reflects the observed patient anatomy. The physician specifies the dominant appearance of the patient's circulation (left, right, or mixed) and indicates the presence and size of the numerous coronary artery segments. Next, a variety of lesions can be placed at the observed location in any segment in the coronary artery tree. Again, using only a finger, the physician may cite the presence and type of lesion (such as discrete or tubular) by pointing to the beginning and end of the lesion and noting the degree of occlusion (the percentage of narrowing). The workstation responds rapidly with a diagram that shows the lesion on the tree, with a shaded area indicating the degree of occlusion. Morphology distal to a lesion can also be documented. Depending upon the inputs of the user, the diagram is updated to show normal or small segments, or to indicate segments that are poorly visualized or not observed. A segment not observed, for example, is represented by dotted lines. The results of therapy can also be noted. After bypass surgery, for example, a second diagram can be created that shows the type, location, and patency of grafts. Single vein, internal mammary, Y, or jump grafts can be selected, and specific origins and anastomoses indicated. After all results are entered, a comprehensive report can be generated within minutes to include the graphic coronary trees and a complete printed summary. The data is then stored in a central data base for future retrieval."

It's impressive to watch, and it demonstrates not only the value of the touchscreen, but also the HP 150's interactive graphics and data communications capabilities. In this application, the HP 150 works with a host computer system—an HP Cath Data Management System, which happened to be nearly 3000 miles away in this instance—and acts like part computer, part terminal. Many owners, of course, will use their HP 150s as stand-alone personal computers, running not only medical and other specialized software, but also the widely used spreadsheet, data base management, and word processing software packages, or even video games. For a look at this powerful personal computer and graphics terminal from the designers' point of view, read the articles in this issue.

-R. P. Dolan

---

## What's Ahead

On the schedule for the September issue are articles on the design of the HP 4945A Transmission Impairment Measuring Set, an instrument used by telephone companies in the U.S.A. and Canada to test voice and data circuits, and on the Semiconductor Research Corporation, a joint effort of several U.S. companies to support university research in the semiconductor field. Two other articles deal with the translation of computer software into local languages. One is a general article on the problems of software localization, and the other describes a language-independent hyphenation algorithm developed for the Dutch version of HPWord, a word processing package for HP 3000 Computers.

---

# Touchscreen Personal Computer Offers Ease of Use and Flexibility

*This powerful 16-bit computer offers an industry standard operating system, many integrated software packages, high-resolution graphics, sophisticated data communications, built-in terminal capabilities, and of course—the touchscreen.*

by Srinivas Sukumar

ITS SMALL FOOTPRINT, ease of use, and overall flexibility differentiate the HP 150 Touchscreen Personal Computer (Fig. 1) from other PCs. This new personal computer consists of three modules: a system processing unit (SPU) with an integrated display, a disc unit, and a keyboard. The SPU and the disc unit occupy only one square foot of desk space. The entire computer including the keyboard occupies only 1.7 square feet. The SPU and keyboard are usable without the disc unit as an HP 2623A Graphics Terminal.

The SPU is based on an Intel 8088 microprocessor running at 8 MHz, and uses Microsoft's MS-DOS 2.11 operating system, an industry standard for which a wide range of popular applications software is available. Memory capacity is 256K bytes of RAM (expandable to 640K) and 160K bytes of ROM. A 1K-bit battery-backed CMOS RAM holds the system configurations.

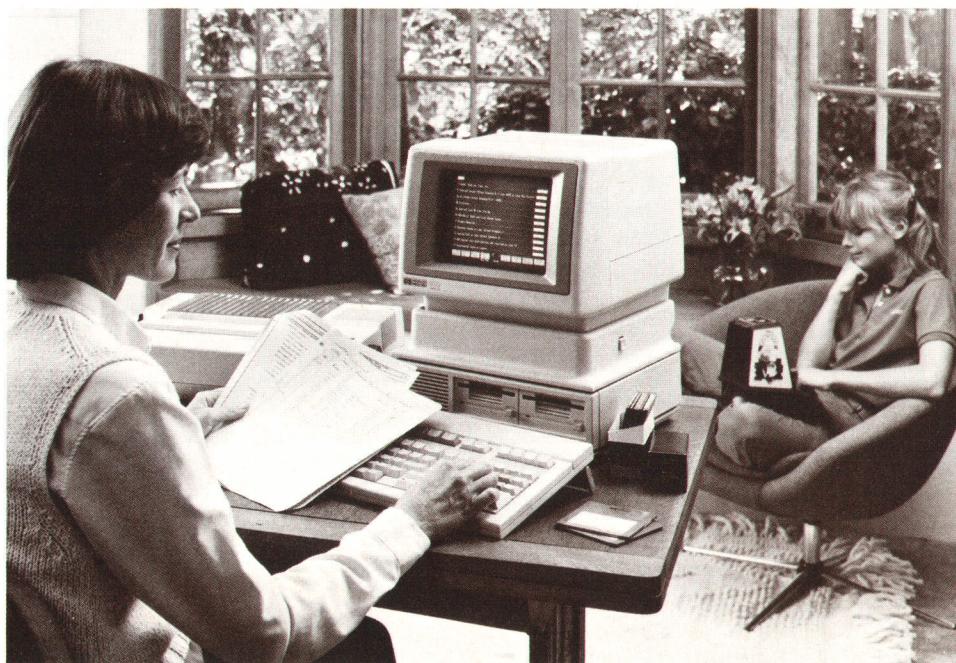
At the back of the SPU (Fig. 2) are three I/O ports: an RS-232-C/RS-422 port for host-computer communications, another RS-232-C port suitable for printers and other

peripherals, and an HP-IB (IEEE 488) port for printers, disc drives, and instruments. There are also two option slots for additional hardware cards.

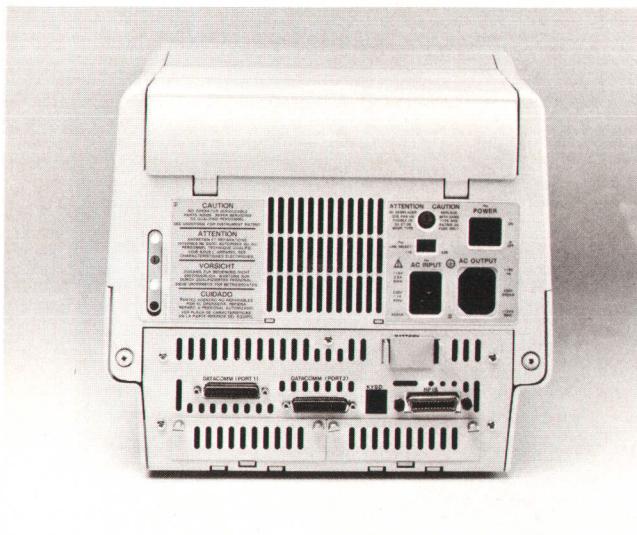
The salient feature of the HP 150 is the touchscreen, which is fully integrated into the system hardware and firmware. The touchscreen allows users to run computer programs with the touch of a finger or a pen. Instead of memorizing commands, typing in menu-selection numbers, or using a mouse, one simply touches the display screen to operate this new personal computer and its application programs. The computer may also be operated from the keyboard without using the touchscreen.

The disc unit has two Sony 3.5-in flexible disc drives and provides 540K bytes of storage. Two hard-disc options are available. The HP 45655A, a 5-megabyte disc drive, and the HP 45660A, a 15-megabyte disc drive, both with 3.5-in flexible disc backup, fit in the same desk space with the HP 150.

The keyboard has a low profile and sculptured keys and sets the standard for all new HP personal computers and



**Fig. 1.** The HP 150 Touchscreen Personal Computer consists of three modules: a system processing unit with integral display, a disc unit, and a keyboard. Without the disc unit, it operates as an HP graphics terminal.



**Fig. 2.** The rear panel of the HP 150 has three I/O ports and two slots (behind slotted panels at the bottom) for hardware options.

terminals. An integral thermal printer can be installed into the unit by the user or the dealer. The printer fits in the top of the display and occupies no additional space.

The green-phosphor CRT displays 80 columns by 24 rows of text (see Fig. 3), and has user-adjustable focus and brightness. The display also functions as a bit-mapped graphics display with a resolution of 512 by 390 pixels (Fig. 4).

### Design Objectives

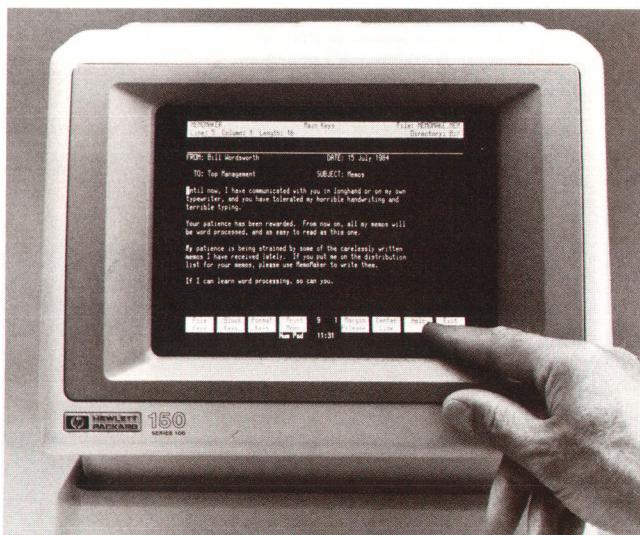
The objectives that guided the design of the HP 150 are reflected in the characteristics of this personal computer. An important objective was the ability to run industry-standard applications programs. This led to the choice of MS-DOS from Microsoft as the primary operating system for the machine. The touchscreen is there to make the machine very friendly to the first-time user, another objective.

The machine had to be a terminal when operating without its disc drives, so it carries all the necessary terminal code in ROM. The rich terminal feature set built into the machine is accessible through operating system extensions that let a programmer create alpha and graphics screens with ease.

The machine is also reliable and easy to build, check out, and service. An extensive set of built-in self-tests lets the factory or field isolate failing components and assemblies. The factory builds the product with no options, so production can be streamlined. The two option slots are for cards installed by the user.

All PCs need printing capability, and in this machine this is provided with no additional burden on scarce desk space. The optional thermal printer sits inside the SPU/display unit and provides fast and quiet printing of everything that can be displayed on the screen (alpha and graphics information). Printing can also be accomplished by connecting an external printer to the RS-232-C port provided on the rear panel.

The ability to adapt the HP 150 to local markets was a fundamental objective. Hence the HP 150 supports 17 dif-

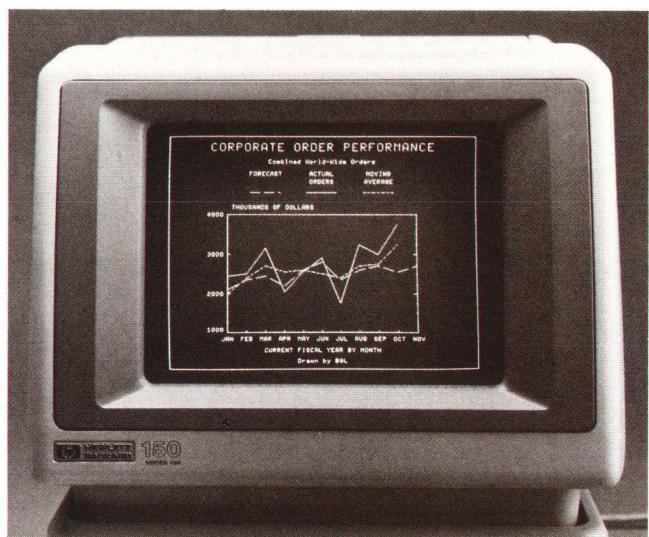


**Fig. 3.** The green-phosphor CRT displays 24 80-character rows of text, plus three rows of status and softkey information.

ferent local-language keyboards through its firmware. All error messages and manuals had to be locally adaptable for the international market. The ease with which the product was localized is a testimony to how well this objective was met.

The small package constraint led to very compact internal printed circuit board design and to the development of a custom gate array for graphics display.

The primary customer for the HP 150 is the business user. The HP 150's 2623A Terminal feature set provides a clean bridge from terminal to computer as the user makes this transition. An optional plug-in card and the built-in terminal features also enable the HP 150 to emulate IBM 3278, IBM 3276, and VT100 terminals. Since the disc unit is modular, the HP 150 can be an economical diskless system with a local area network connection.



**Fig. 4.** The graphics display has a resolution of 512×390 pixels.

## Acknowledgments

I would like to thank the firmware team led by John Lee, consisting of Thuy Nguyen, Nina Chuang, Quan Vu, Becky Smith, Carl Peterson, Peter Stratton and Zvika Bronstein. Stan Boyd was the project manager for the operating system, and his team included Chuck Whelan, Laurie Wood, Vicky Spilman and Dirk Maasen. My thanks are also due to Howard Lee, the hardware project manager, and his team—John Watkins, George Szeman, Rich Wales, Susan Carrie, Tom Kilbourn, Robert Ko, Trish Brown, and Robert Heyde. The mechanical design team was led by Rick Nelson and consisted of Lee Winick, Jun Kato, and Ralph Ropp. My thanks also to the many people in Personal Office Computer Division who made it all come together including

Darrell Scherbarth, IHV project manager, and his team, and Mark Della Bona and his team on the keyboard.

The product would have not been as successful without the software contribution of Personal Software Division. My thanks to section managers Steve Ng, Bill Finch, and Jim Sutton, and project managers Rob Shurtleff, Sherry Ramsey, Barbara Packard, John Alburger, Sharon Mead, and Ken Mintz, and their teams.

I would also like to thank Bill Johnson, program manager for the HP 150 introduction, Carol Mills, product manager, for her great enthusiasm and support, Pierre Patkay, production engineering manager, and Bob Bowden, materials manager, and their teams.

# Operating System and Firmware of the HP 150 Personal Computer

by Laurie E. Pollero Wood and Charles H. Whelan

**A** FEW SECONDS after the HP 150 Personal Computer is turned on, the Personal Applications Manager (PAM) displays the applications that can be run and prompts the user for input. PAM is the HP 150's user interface. It makes the personal computer easy to use, which is a real bonus for novice users. Beneath the friendly user interface is a supporting structure of operating system and firmware.

### Personalities of the HP 150

The HP 150 has two built-in personalities. The personal computer personality requires access to the operating system, which is usually on a disc in a disc unit connected to the HP 150. The terminal personality requires connection to a remote computer via one of the RS-232-C ports. The user selects the machine's personality after turning it on, and can move between personalities once the system is up and running.

As a personal computer, the HP 150 runs Microsoft's disc operating system, MS-DOS 2.11. This industry standard operating system is single-tasking and written to be ported to various hardware systems. Porting it to a new piece of hardware requires writing device drivers and linking them into the MS-DOS code supplied by Microsoft.

The HP graphics terminal capabilities are always available (they are in read-only memory) and emulate the HP 2623A Terminal. The terminal emulator is controlled by a multitasking operating system called the TOS (terminal operating system).

### System Structure

The system can be thought of as concentric layers of code (see Fig. 1). The center is the HP 150 hardware. Firmware (code permanently recorded in ROMs) provides the drivers that control the hardware and the TOS. The code in RAM (loaded from disc) provides the personal computer personality. This RAM code contains two logical parts, the BIOS and the MS-DOS. The BIOS is the connecting link between the MS-DOS operating system and the firmware code. PAM is actually an MS-DOS program that knows how to initiate other MS-DOS applications.

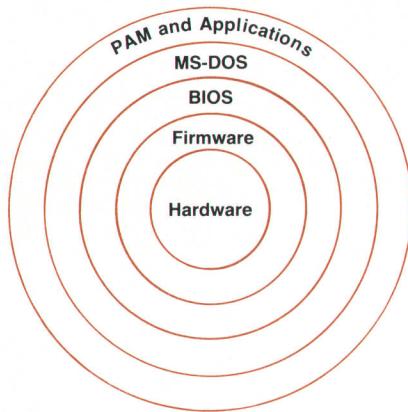
The HP 150 powers up as a terminal with the capability of loading the operating system from disc. Whenever the HP 150 is being used as a terminal, the user can always get to the operating system by pressing the (**Shift**)**Stop** key. When the HP 150 is being used as a computer, the user can access the terminal through a program such as PAM or by configuring the HP 150 to be a terminal and doing a hard reset.

### What Is MS-DOS?

MS-DOS 2.11 is an operating system developed by Microsoft Corporation that can be ported to computers using the Intel 8086 family of microprocessors.

MS-DOS provides a set of software services for programs running in a single-user environment. Basically, only one program can be running at a time, although it is possible for one program to run another and wait until it completes before resuming.

The major facility provided by MS-DOS is a set of services for managing files and I/O devices. The programmatic interface to MS-DOS files and I/O comes in two flavors. For the



**Fig. 1.** The HP 150 operating system can be thought of as layers of code. Firmware in ROM provides the hardware drivers and the terminal personality. BIOS and the MS-DOS operating system are loaded into RAM from disc. The Personal Applications Manager (PAM) is an MS-DOS program that knows how to initiate other applications.

CP/M® devotee there is a set of calls that have evolved directly from the CP/M operating system. With the 2.11 version of MS-DOS, a set of file and I/O capabilities that are very much like those of the UNIX™ operating system were added to the system. HP has chosen the latter set of calls for its own internally developed applications, while the CP/M-like calls are typically used by applications written for machines such as the IBM PC.

The hierarchical file system of MS-DOS 2.11 provides a directory tree structure similar to UNIX, so that users can logically organize their data on the disc. This structure allows files to be grouped in more manageable subdirectories, which is particularly important on the larger fixed discs.

#### What Is BIOS?

The HP 150 BIOS is a layer of software between MS-DOS and the HP 150 firmware/hardware. On system boot-up, the BIOS gains control from the boot firmware, and after initialization, passes information and control to MS-DOS so it too can configure and initiate itself. The BIOS is then invoked by MS-DOS whenever it requires a physical I/O service.

The BIOS uses information stored in nonvolatile RAM by the device configuration utility to determine the current mapping of logical MS-DOS device names and logical disc drive identifiers to physical peripheral devices. This information can typically be changed dynamically while the system is running.

The operating system stored on bootable disc media is actually two binary files (BIOS and MS-DOS) which are loaded into memory on boot-up. The BIOS consists of a section of initialization code to set up the HP 150 as a personal computer (rather than a terminal), the various modules for interfacing high-level I/O requests to the HP 150 firmware drivers, and a SYSINIT module supplied by Microsoft to which the BIOS passes control and data after it performs its own initialization. The BIOS gives SYSINIT

UNIX is a trademark of AT&T Bell Laboratories.

such information as a linked list of resident devices and their invocation addresses, the current address of the booted MS-DOS module, and the address where that module can be relocated. The latter allows the MS-DOS resident code to overlay and reuse the memory space required for BIOS initialization and SYSINIT.

#### MS-DOS/BIOS Interface

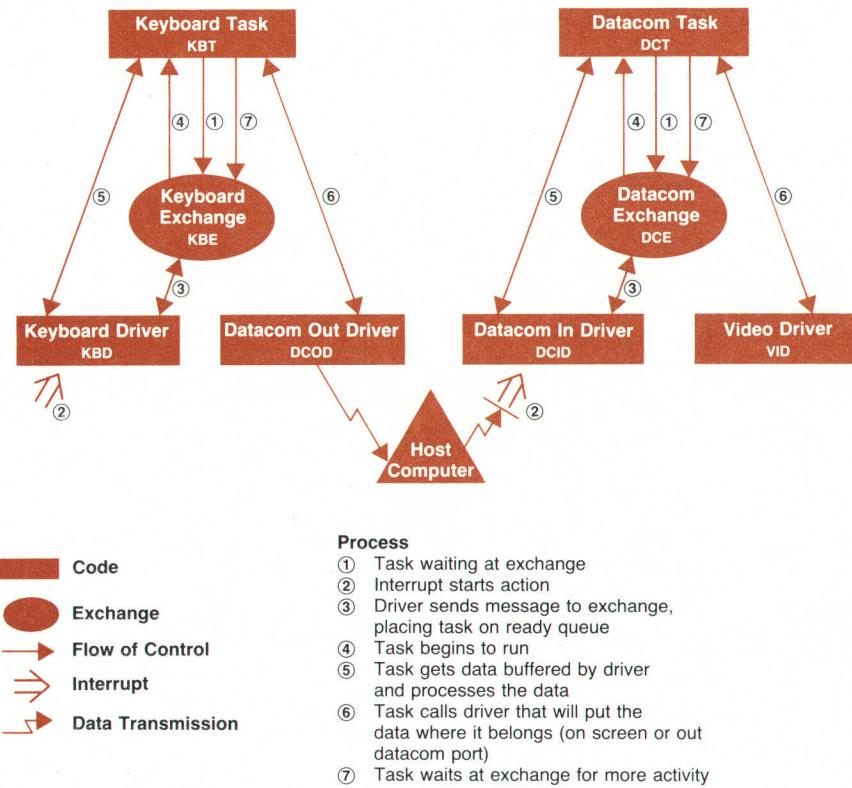
All requests for I/O, including disc operations, are presented by MS-DOS to the BIOS in a packet structure that includes information such as the type of operation and the I/O buffer address. The BIOS is responsible for performing the physical I/O operation and posting status information in the packet for MS-DOS upon completion. Once a system is up and running, the only interaction between MS-DOS and the BIOS is by way of these packets.

Two types of peripheral devices are supported by MS-DOS: character and block. Block devices are devices such as discs, which do their I/O in fixed-size blocks. For example, application program file I/O requests to MS-DOS are passed to the BIOS as reads and writes of specific absolute disc sectors. Before returning, the BIOS will update the packet with the error status and the number of sectors successfully transferred.

As the name implies, character devices do character-oriented I/O. Character devices have names that can be opened and accessed like MS-DOS files. Through the HP 150 device configuration utility, DEVCONFIG, many of the standard named character devices can be mapped to different devices. For example, the standard printer PRN can be mapped to the internal printer, HP-IB printers, serial printers, or parallel printers.

MS-DOS provides a facility to allow device drivers to be added to a system when it is booted. This allows new block or character devices to be added or old drivers to be replaced without having to create a new system. In the latest version of the HP 150, this concept is extended. An installed driver may optionally interact with the system so that it can be dynamically configured and accessed using either the standard named character devices or the block device addresses commonly allocated to HP disc drives. For example, a character device driver named PRN would replace the PRN driver in the BIOS at boot-up time. In this case, the logical device driver has been replaced. Or, a new device driver can link into the chain of physical device drivers available for a particular logical device such as PRN. The DEVCONFIG program allows the user to select the physical interface from several choices, one of which is the newly added physical device.

The CON device handles I/O input requests from the keyboard and output requests to the video screen. In its simplest fashion, reading and writing data to the console is much like teletypewriter operation. Standard HP escape sequences are forwarded to the firmware driver. To provide greater console performance and functionality, the alpha/graphics I/O system, AGIOS, was added to the HP 150 console handler. The AGIOS is implemented by passing commands to the CON driver using the I/O control programmatic interface of MS-DOS. These commands provide field-oriented alpha screen handling, graphics intrinsics, keyboard control, and touchscreen management. They



**Fig. 2.** Data flow in a typical terminal-to-remote-host-computer configuration.

allow an application program to take over the keyboard and screen.

The HP 150 supports a wide range of HP disc devices. The BIOS determines the physical attributes of each drive dynamically by reading the first sector on each disc. These attributes are recorded there by the HP 150 FORMAT program. Because of this method, as new disc drives are developed they can be supported by the existing BIOS if they follow the existing disc protocols and if the FORMAT program is revised to initialize the first sector properly. When the device configuration program changes configurations, or when a disc device is powered on, the BIOS will reread sector 0 at the time of the next disc access. This allows drives to be added or changed on-line without rebooting the system.

The HP 150 BIOS supports I/O requests to access HP-IB (IEEE 488) or RS-232-C peripherals. Application programs can communicate directly with HP-IB devices by passing templates to the firmware HP-IB driver. A template is a data structure containing HP-IB commands to perform such functions as bus control and device addressing. This facility gives the application the ability to exercise complete control over the commands and data communicated to HP-IB peripherals.

The latest version of the HP 150 BIOS also supports a RAM disc, which allows a portion of memory to be treated as a disc. This can provide greatly improved performance when accessing frequently used programs and data files. The device configuration utility program allows the user to specify the amount of RAM allocated and which drive identifier (if any) addresses the RAM disc. The fact that a given drive is in RAM is transparent to application pro-

grams. Of course the user must realize that any files stored on the RAM disc will be lost if the system must be rebooted.

### TOS Characteristics

The TOS (terminal operating system) in firmware is used by both the terminal and the personal computer personalities.

Drivers in the TOS control the interface with the hardware. Examples of drivers are HP-IB, datacom, keyboard, alpha video, graphics video, and printer. More particularly, drivers provide data to the system and remove data from it. Drivers are sources and/or destinations for data.

In the TOS, a task is a unit of work. For example, one task, the keyboard task KBT, is responsible for picking up keycodes from the keyboard driver, translating each keycode into a character or a function to be performed, and deciding what to do with the data obtained. A similar task exists for processing datacom input. When the system is initialized, approximately ten tasks exist.

The TOS maintains a task queue consisting of two FIFO (first in, first out) subqueues. One subqueue, the background subqueue, contains tasks that will be executed only when the foreground subqueue is empty. Only one task can be executing at a time; it is called the running task and is at the head of the task queue. All tasks that are capable of running are in the task queue and are said to be in the ready state. The task queue is referred to as the ready queue.

Frequently, the terminal tasks do not have any work to do. For example, the keyboard task has nothing to do if the user hasn't used the keyboard recently. A task is in the waiting state when it has finished its work and is waiting

for more work to do. Waiting tasks are not in the ready queue.

The TOS does not predetermine the amount of processor time a task will receive. Once a task is running, it has control until it decides to relinquish the processor. A task can yield, which puts it at the end of the ready queue and allows the other ready tasks to run.

Communications in the system are handled by sending messages via exchanges (mailboxes). Task-to-task communication is always handled via exchanges and messages. Drivers talk to tasks through exchanges too, although tasks can call drivers directly. Exchanges are places a task can wait for data or commands. When a task is waiting, it is associated with an exchange. When a message is sent to the exchange, the task is added to the ready queue and eventually gets to run. It then has the opportunity to process the data or the command specified in the message. The only way information can be sent to a task is by sending a message to the task's exchange.

For example, in a typical terminal-to-remote-host-computer configuration, the general flow of data is as follows. Keyboard characters are transmitted to the host. The remote host echos the characters it receives back to the terminal and also sends other characters to the terminal. The terminal takes datacom input and puts it on the display. Hence, when a user types at the keyboard, the characters that appear on the display have gone to the remote computer and come back again. In the TOS this situation is described by Fig. 2.

When the machine is waiting for input, both the keyboard task KBT and the datacom task DCT are waiting at their respective exchanges (KBE, DCE) for a message. When the user starts to type, an interrupt is generated and the keyboard driver KBD sends a message to the KBT's exchange KBE. This puts the KBT on the ready queue. Once the KBT is running, it calls the driver and asks for the keyboard data. Then KBT processes the data (What is the character? Is it shifted? Is this a special function key?) and determines what should be done with it (send to datacom or the display, or handle the special function). Because of the current configuration of the machine, the KBT calls the datacom out driver DCOD. The DCOD transmits the characters to the

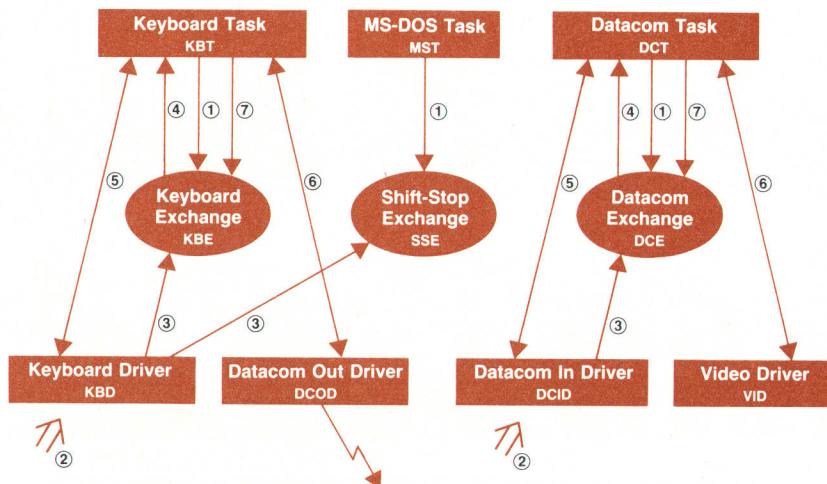
remote host and returns to KBT. Now the KBT is finished and it goes back into the waiting state, waiting for another message at its exchange KBE.

The remote host echos the characters back to the machine. Receipt of datacom input generates an interrupt and the datacom in driver DCID sends a message to the DCT. This puts DCT on the ready queue. When DCT begins to run, it calls the DCID and asks for the data. The DCT checks the data for escape sequences and processes them if they are present. The DCT then calls the video driver, passing it the characters to be displayed. The video driver puts the characters on the screen and returns to DCT. The DCT has finished its task. It now waits at its exchange for another message, that is, more data from the remote host.

### MS-DOS/TOS Interface

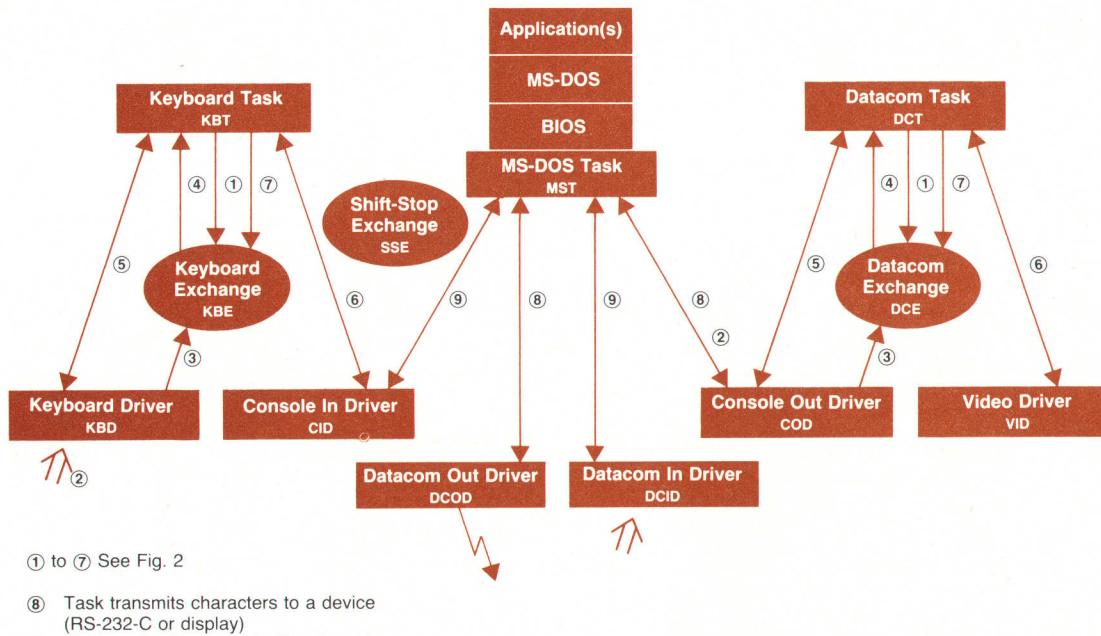
MS-DOS runs as the TOS task MST. MS-DOS makes use of the other TOS tasks and some of the TOS's drivers and processing routines. During MS-DOS initialization, a new driver, the console, is added and the existing drivers are reconfigured for use as a personal computer. For example, the datacom drivers are detached from the keyboard and display and associated instead with the MS-DOS logical drivers, COM1 and COM2.

The MS-DOS task MST is initialized at power-on and its purpose is to boot the operating system from disc at the appropriate time. MST waits until the terminal personality tasks complete their initialization. This puts the HP 150 into a known state before the MS-DOS initialization begins. MST then checks to see if the HP 150 should power up with a terminal personality or a personal computer personality. If the personal computer is the personality of choice, then the booting takes place immediately. If the terminal personality is selected, then MST waits at the shift-stop exchange SSE for a message. The keyboard driver sends a message to this exchange whenever the user hits **(Shift)Stop**. When MST receives the message, the booting process is continued. Once it is determined that the personal computer personality should be loaded into memory from disc, MST reads the second sector on the disc. This sector contains such vital information as which sectors to read, where to place the bytes in memory, and where to start executing. After



① to ⑦ See Fig. 2

**Fig. 3.** HP 150 system before BIOS initialization.



**Fig. 4.** HP 150 system after BIOS initialization. BIOS initialization's main purpose is to change the personality from a terminal to a personal computer.

the sectors are read into memory, MST jumps to the execution start address, which is the BIOS initialization entry point. Now the task's function is to run MS-DOS and the programs run by MS-DOS.

The BIOS initialization's main purpose is to change the HP 150's personality from a terminal into a personal computer. Pictorially, the system changes from Fig. 3 to Fig. 4. Previously, data from the user (keyboard and touchscreen) was funneled out the RS-232-C port via the DCOD. Now the data is sent to the console input driver CID. The CID buffers the characters and waits for MST to request them. When MST wishes to send characters to the display, MST passes the characters to the console out driver. The COD sends a message to the datacom task at its exchange and buffers the characters. Then when the MST relinquishes the processor, the DCT is ready to run. The datacom task contains the control mechanism for processing computer input (escape sequences are recognized in the DCT), and the characters are eventually sent to the video driver, which puts them on the screen. The MST task can send as many characters as it likes to the console out driver, but nothing will be displayed until the MST gives up control of the processor, which allows the DCT to run. The TOS system is optimized for handling buffers of data rather than single characters. MST can control when the task swap occurs and optimize use of the machine.

The datacom drivers have now been detached from the keyboard and datacom tasks. The DCOD and the DCID are now associated with the MS-DOS task directly. MST sends characters to datacom by calling the DCOD, which sends them out the RS-232-C port immediately. The DCID buffers incoming characters, waiting for MST to request them. Note that DCID waits for a request from MST for the characters, instead of sending a message to an exchange. The use of

the exchange is inappropriate for data input in the personal computer personality. The computer personality gives control to the application, which continues running except when it has explicitly made a request for I/O service. In the terminal personality, on the other hand, incoming characters cause the DCID to send a message to the associated task (DCT). This is because the terminal tasks are usually waiting for input. They expect to be interrupt-driven, whereas the personal computer personality does not.

### Conclusion

The HP 150 is a successful attempt at creating a friendly dual-personality machine. The merging of personal computer capabilities with a full-featured terminal personality required painstaking engineering effort to ensure that each capability was achieved without sacrificing completeness, friendliness, or performance. We had to do some time-consuming fine tuning to achieve satisfactory performance in a system with several structured layers of software between an application program and the hardware.

The HP 150 has far more functionality than the average purchaser realizes. It is a powerful machine that provides a simple yet effective environment for the beginning user, but like an iceberg, it has a huge mass of capability below the surface, waiting to be tapped by the sophisticated user.

### Acknowledgments

We wish to acknowledge the contributions of the following people to the HP 150 operating system and firmware: Stan Boyd, Zvika Bronstein, Nina Chuang, Tom Hall, John Lee, Dirk Maasen, Michael McGrath, Thuy Nyugen, Carl Peterson, Rebecca Smith, Vicky Spilman, Peter Stratton, Quon Vu, Franc Woods, and the HP 2628A firmware project team.

# The HP 150 Touchscreen: An Interactive User Input Device for a Personal Computer

by Peter R. Stratton, Scott R. McClelland, and Thomas E. Kilbourn

**T**HE HP 150 TOUCHSCREEN is an interactive user input device that provides a close, intuitive match between the user and the action performed by the computer. The objectives of the touchscreen design included medium resolution, manufacturability, low cost, good appearance, and reliability.

Various technologies have been used in touchscreen designs. Capacitive and resistive films have been used to provide information about the location of a stylus touching the screen. These work by providing a repeatable varying capacitance or resistance across the sensitive area. One effect of these devices is a degradation of the display quality, since the user must look through the film. Additionally, the lifetime of these films is limited.

High resolution can be obtained with ultrasound. By measuring the change in sound transmission through the glass when a stylus touches it, or by measuring the time of reflection of sound impulses from the stylus, the position of the stylus can be determined. These devices tend to be expensive. Furthermore, they can be very sensitive to the dust particles on the display, which may cause false reflections.

Another technique is to place pressure transducers on the corners of the display. These provide data that can be

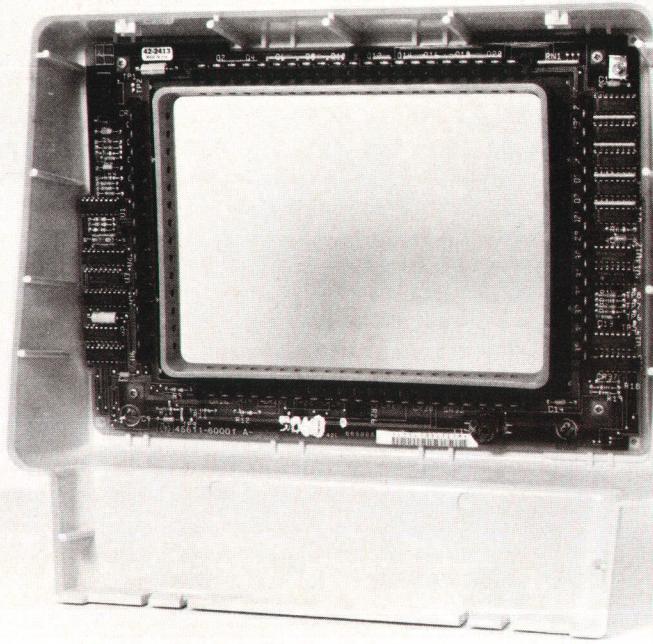
used to calculate the position of a stylus pushing on the display. A disadvantage of this is the necessity of providing a mechanically sound mount for the display without making the transducers ineffective.

## Infrared Array

The method chosen for the HP 150 design is a scanned infrared array. Capable of medium resolution at a moderate cost, the nonvisible light does not degrade the display, and the solid-state design provides high reliability.

Infrared emitting diodes are arranged in lines along the bottom and one side of the display. Directly opposite each emitter is a phototransistor. By pulsing an emitter and looking at the output of the phototransistor directly opposite, it can be determined whether that path is blocked. By scanning along the top and then along the side, the (x,y) coordinates of any object within the array can be determined.

Given that the design would use infrared pairs, several challenges presented themselves. One was to squeeze the design into the bezel of a nine-inch display. Volume was somewhat limited. Another challenge was to design the system without multiple connectors. These concerns helped drive the choice of the infrared diodes and the phototransistors. The parts chosen have a 90° angle be-



**Fig. 1.** The HP 150 touchscreen uses a scanned infrared array mounted on a printed circuit board with a cutout for the display. Infrared emitting diodes are arranged along the bottom and one side of the display. Directly opposite each emitter is a phototransistor. The user's finger blocks the infrared light seen by one or more phototransistors.

tween the leads and the light source or sense direction, so they can be mounted on a single printed circuit board with a cutout for the CRT (see Fig. 1).

As shown in the block diagram, Fig. 2, the scanning is accomplished under the control of a single-chip 8041A microcomputer. The microcomputer provides a clock to a 6-bit counter, the output of which is decoded as an address for an IRED (infrared-emitting diode). A pulse generator provides an on pulse after a suitable delay. The counter output is also decoded to gate the output of the appropriate phototransistor into the detector. The data from the detector is transmitted to the 8041A. By keeping a software count synchronized with the hardware counter, the address of the blocked pairs can be determined with only the two interface lines, clock and data. The 8041A need only toggle the clock line and then read the data in successive cycles. Thirteen scans are completed each second.

An additional line, sync, is provided for error recovery. In the case of transients on the clock line, the software and hardware counts can get out of phase. The 8041A can detect this by checking the sync line at the end of a cycle.

### Scanning Algorithm

After completing a scan, the microcomputer must determine whether a valid touch, or hit, has occurred, and report its position to the HP 150's main 8088 processor. The requirements for a valid hit are simple. There must be one contiguous blocked group of pairs in each direction (horizontal and vertical). The x or column coordinate then is the center of the horizontal blocked area and the y or row coordinate is the center of the vertical blocked area. If the resulting (x,y) pair is different from the last valid coordinates (or there was no touch on the previous scan) and the synchronization pulse occurs at the right time, then it is a valid touch and is reported to the 8088.

Fig. 3 is a flow chart of the scanning algorithm. The count is divided into two segments, row addresses and column addresses. A valid hit consists of at least one column pair interrupted and at least one row pair interrupted. Interruption of more than one row or column is also accepted unless they are separated by an unblocked pair. This would indicate multiple objects touching the display, an ambiguous situation.

In determining how many pairs were required to achieve the desired resolution, it was assumed that the spacing between adjacent pairs would be small enough that a per-

son's finger could block two pairs. This assumption resulted in the effective resolution being nearly twice the number of pairs. If the finger blocks an odd number of pairs, then the effective hit address is the address of the center pair. If the number of blocked pairs is even, then the effective address is halfway between the two pairs at the middle of the object. This is easily calculated by the 8041A.

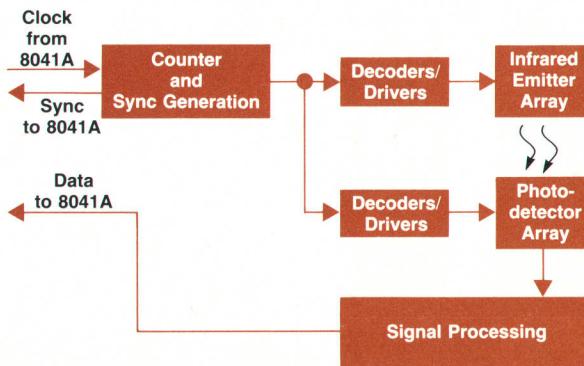
During the scan, the 8041A first looks for an interrupted pair as it counts through the column addresses. If one is found, that pair number is stored and the count continues, now looking for an unblocked pair. Once an unblocked column pair is found, the two numbers are added. This results in a mapping of the average of the blocked pairs into a column number with twice the range of the pair numbers. The same algorithm is followed for the row pairs. This algorithm yields a position with a resolution of nearly twice the number of pairs, or 41 columns by 27 rows. The 8088 CPU uses the row number (0-26) directly and maps the column number (0-40) to alpha screen coordinates (0-79).

### Detector Design

The detector design is based on an assumption as to the spectrum of the modulation on the light received by the phototransistors. There are three principal sources of light driving the bases of the phototransistors. Sunlight, overhead or desk lighting, and the IREDS. Sunlight consists of energy spread throughout the electromagnetic spectrum. The light waves at different frequencies are essentially at a constant amplitude. That is, the energy is modulated at a frequency close to zero. Indoor lighting is most often produced by ac light current, resulting in amplitude modulation at 120 Hz (100 Hz in Europe.) The output of the IREDS is modulated at the frequency of the clock, about 700 Hz. All of this leads to the idea of building the detector with a high-pass filter that has a cutoff frequency between 120 and 700 Hz. The driving pulse to the IREDS is nearly square, which results in large-amplitude harmonics. Thus, the cutoff frequency can be somewhat higher than 700 Hz. In the HP 150 touchscreen design, the filter is implemented with a large inductor that provides a low-pass path to ground at the output of the phototransistors. This is followed by a bandpass filter, the output of which goes into a comparator. The comparator converts the signal to a CMOS-compatible level, which can be gated to the 8041A microcomputer.

### Initialization

The power-on initialization sequence for the scanner starts with a self-test which checks the on-chip memory and then the array hardware. The scanner first attempts to synchronize with the hardware by rapidly pulsing the clock and sampling the sync line. If the sync doesn't appear after a reasonable number of clocks, the scanner assumes the touchscreen is defective or not connected and no more interaction is attempted. If the sync signal is found, the scanner does one complete scan of the pairs to make sure they all appear unblocked. Experience has shown that defective pairs fail such that they appear blocked instead of open, so this is a fairly good verification of hardware integrity. If there are any bad pairs, the scanner will compensate



**Fig. 2.** Touchscreen block diagram. A single-chip 8041A microcomputer controls the scanning of the array.

for up to two of them to allow continued operation with slightly degraded performance. It compensates by merely ignoring the blocked indication of those pairs suspected to be bad.

At this point, the 8088 system processor requests the results of the self-test and the synchronization attempt and the addresses of any bad pairs found. Normal scanning for touches commences when the 8088 commands it.

### Touchscreen Firmware

The HP 150 touchscreen firmware module consists of a set of firmware routines that define a simple-to-use yet powerful virtual machine for touchscreen operations. The virtual machine accepts data input from the touchscreen hardware and control input from the target program and outputs data to the target program and control operations to other parts of the system firmware (Fig. 4).

The target program is usually a software application program running under the MS-DOS operating system. However, it can also be another firmware module (e.g., the configuration firmware), system software (e.g. the Personal Applications Manager), software written by the user, or a pro-

gram running on a remote computer, communicating via a data communications port.

The 8041A microcomputer detects an initial touch on the display surface, a change of touch position on the display surface, and the disengagement of touch with the display surface, called the "release," and sends this information to the touchscreen firmware running on the 8088 system processor. In the first two cases, the 8041A sends position information to the firmware, and in the third case, it sends the information that a release has occurred. In all cases, the firmware checks its internal coordinate map to determine the operation that should be performed for that position on the screen (Fig. 5).

### Firmware Control Inputs

The touchscreen firmware may be called by the target program to establish certain environmental parameters. The overall effect is to define the function to be performed when a particular character cell on the display is touched by the user. The two main control functions are as follows.

**Enable/disable (row, column) touch data reporting mode.** This mode causes the report data from the touchscreen

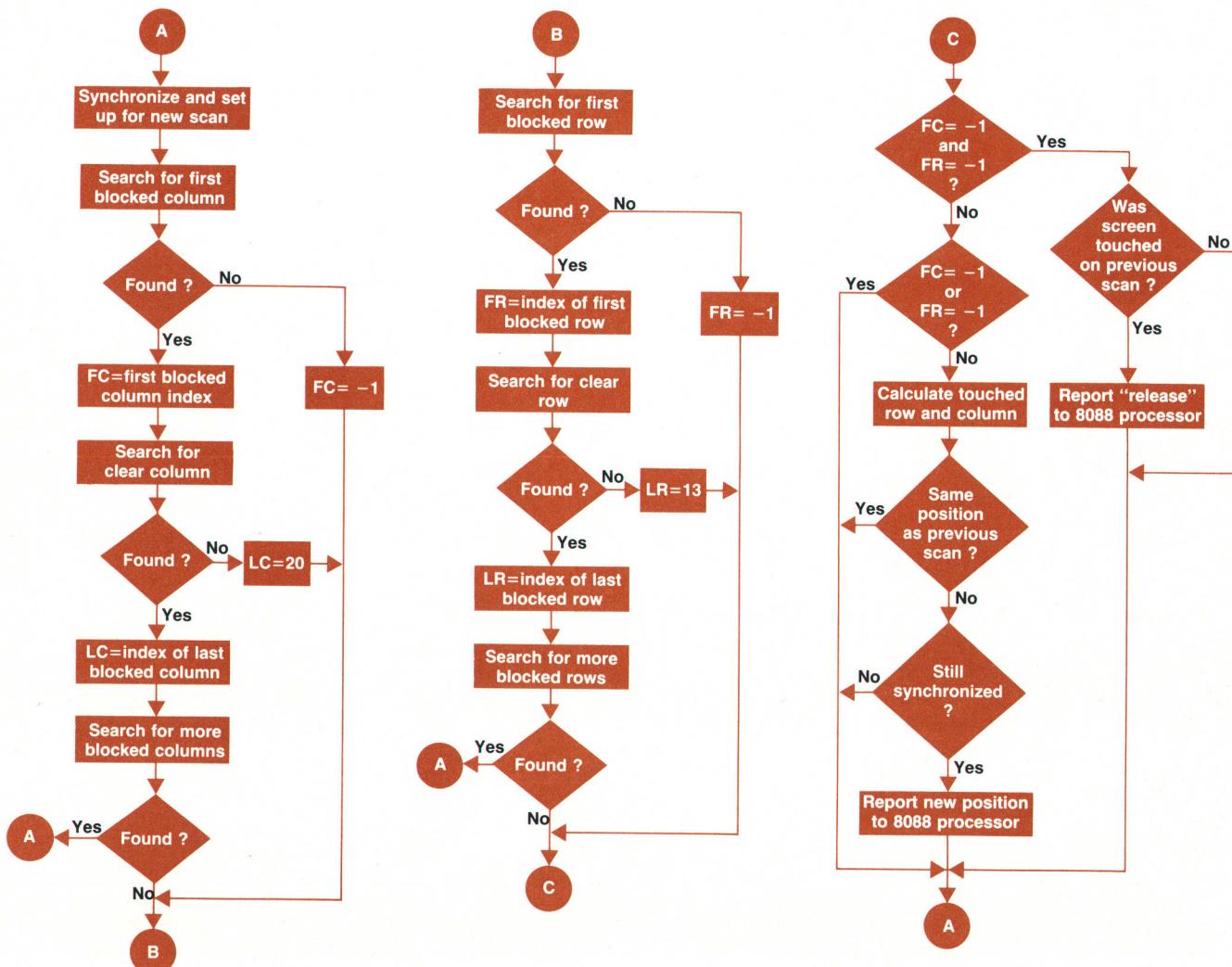
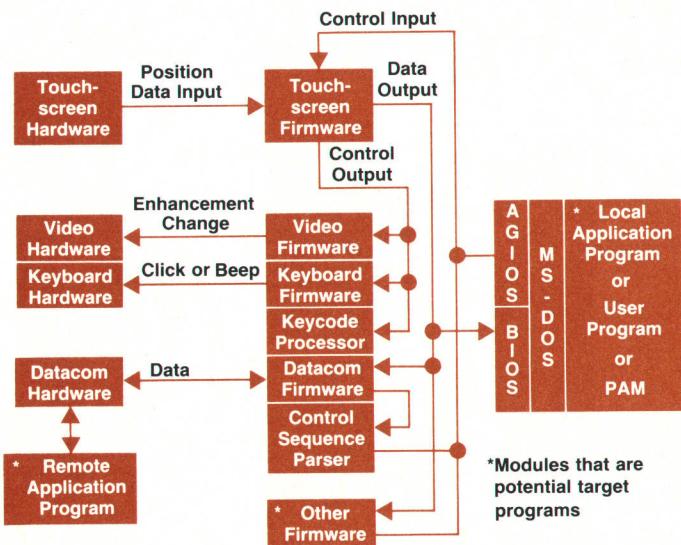


Fig. 3. Flow chart of the touchscreen scanning algorithm.



**Fig. 4.** Touchscreen control and data paths.

hardware to be translated into a screen coordinate pair corresponding to the row number and column number of the position touched by the user. The (row, column) data report is sent to the target program only if there is no field defined for the touched position.

**Define touch fields.** The target program can define rectangular areas of the screen so that they behave in a particular manner when touched and/or released. There are several different types of touch fields that can be defined. The different field types have behavior modes that differ somewhat from one type to another, but there is an underlying functionality that is common to them all:

- The rectangular area of the screen that is enclosed by the bounds of a particular field responds to touch as a whole entity. Touching one point within the area produces exactly the same effect as touching a different point within the area. If the field has been defined such that the display enhancement changes when touched and/or released, then the whole area will change its enhancement regardless of which point within it has been touched.
- The field is debounced. After the initial touch has been made in the field, further movement within the field is ignored until either a movement is made to a position outside the bounds of the field or the touchscreen is released.
- The field causes a particular operation to be initiated when it is touched and/or released by the user. There is a one-to-one mapping from an individual field to the operation that it initiates.
- Visual and audio user feedback mechanisms assist the user in determining when an attempt to touch and/or release a field has been successful.

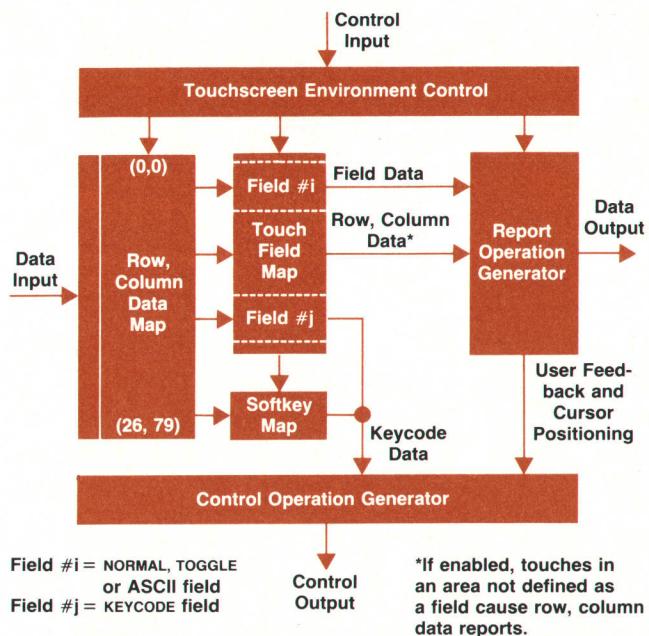
The softkey labels on the display screen are predefined touch fields. Touching a particular softkey label will cause the same action to be performed as would occur if the user pressed the corresponding function key on the physical keyboard.

#### Touchscreen Firmware Outputs

Touchscreen data may be sent to the target program by

the touchscreen firmware as a result of a touchscreen action performed by the user. With the exception of ASCII fields, the data generated is formatted so the target program can distinguish it from other types of data (e.g., keyboard data). The touchscreen firmware is capable of generating touchscreen data in both an internal binary form for communication with target programs running internally to the system and also in an escape-sequence form for communication with a remote target program. The data generated by ASCII fields is not distinguishable from data generated by operating the physical keyboard for either local or remote target programs.

The touchscreen firmware also generates control outputs. Touching a softkey label may cause the corresponding softkey function to be performed. Touch fields may be de-



**Fig. 5.** Block diagram of the mechanism for mapping touch position to control and data operations.

fined such that they behave exactly like a particular control key on the physical keyboard (e.g., cursor up, backspace, carriage return). All operations initiated by the touchscreen firmware as a result of the user touch and/or release action cause an audible simulated keyboard click to occur. In addition, touch fields may be defined such that they cause an audible beep to occur when touched, such that the cursor is positioned at the upper left corner of the field when the field is touched and/or released, or such that the display enhancement changes when the field is touched and/or released.

### Soft Keyboard

The main effect of providing this environment is that the target program can define a soft keyboard on the display that is totally independent of the physical keyboard and hence is not constrained by the limitations of the traditional physical keyboard. The keys in this environment are implemented using touch fields. The keys can be of any size, rectangular shape, and position on the display within the physical constraints of the dimensions and character resolution of the display.

The target program can define keys that are of different types (e.g., toggle keys or not), different user feedback modes, and different functional characteristics. The functional characteristics of a particular set of keys can change dynamically with the state of the target program, and the whole keyboard can change from one keyboard form to another with the state of the target program.

The input/output functionality constraints imposed upon the target program by the traditional keyboard/display environment can be greatly reduced by the use of the touchscreen firmware environment. This allows the target program to be a more powerful tool for the user than it otherwise would be.

### Acknowledgments

Many thanks to the people who were involved with the touchscreen design, in particular to Jun Kato, who designed the mechanicals and the plastic parts used for the touchscreen, and Srinivas Sukumar, who, as project manager and later section manager, championed the touchscreen concept throughout the project.

## Applications Software for the Touchscreen Personal Computer

*HP-developed text editing, card file, graphics, spreadsheet, and calculator packages are designed to maximize the benefits of the touchscreen.*

by Peter S. Showman, Karl W. Pettis, Karlie J. Arkin, Jeffrey A. Spoelstra, John Price, W. Bruce Culbertson, and Robert D. Shurtleff, Jr.

COMPUTERS ARE ENTERING the lives of more and more people who are not and don't wish to become computer experts. Various approaches have been tried to make computers less forbidding, ranging from using familiar visual metaphors (such as the Rolodex file simulation described elsewhere in this article) to voice output and input. Questions and responses printed by computer programs have been made less terse and more human-like, although many people become annoyed when they find the computer being polite but most unhelpful.

One consistent observation about how people perform tasks that are somewhat unfamiliar is that choosing from a set of possible alternatives is easier than trying to remember the alternatives. For example, people have a much larger vocabulary of words they can recognize than of words they can recall. It is also fairly obvious that you can more quickly choose an item from a small set of choices than from a large set, and that it is easier to find something in

a list of items that occur in some rational (or at least familiar) order than in an apparently random order.

To a novice user, one intimidating aspect of most computer systems is the need to learn several kinds of unfamiliar commands to begin using the system. Running application programs, getting and saving data to be used by the programs, and quitting are all necessary steps to begin using the system. Before long, listing directories and performing various manipulations on data files will also be needed to keep track of the user's work. A typical operating system has dozens of commands available, and even if the beginner doesn't use most of them, their existence adds to the confusion.

Another problem not limited to beginning users is the large number of files that seem to accumulate in any computer system. Some represent programs or program fragments; others contain data. Some filenames have been more or less explicitly created by the user (when data has been

saved by application programs, for example), but other files have been created without the user being aware of them until they are encountered in a directory listing. While operating systems (including the version of MS-DOS used in the HP 150) usually provide tools for sifting through these files, the novice user may not know how to use them.

## PAM

The HP 150 Personal Applications Manager (PAM) addresses these concerns by helping the computer user manage both application programs and other disc files. The PAM screen is what the user normally sees after turning on the HP 150 (see Fig. 1). It presents a list of the application programs available on the discs currently in the system, and allows the user to run any of them simply by touching the program name on the screen and then touching the Start Application softkey label. There is no need to remember either what programs are on the disc or what command sequence is needed to run each program. The user can even change the name displayed by PAM if another name seems to make more sense. The display is kept simple by not showing other filenames on this screen, an important consideration since they typically outnumber the applications by more than ten to one.

A utility program provided with PAM allows the user to "install" application programs onto the system discs. This performs two functions: it copies the files needed by the program onto the disc, ensuring that all are present, and it adds the program's name to the list used by PAM for displaying application names. The system is designed to remember what files constitute the program so they can also be removed easily if a new version is installed or if the user wishes to delete the program for some other reason.

Another utility program allows the user to change the screen labels used to represent the application programs and to reorder the labels for each disc—for example, to put

the most commonly used programs first.

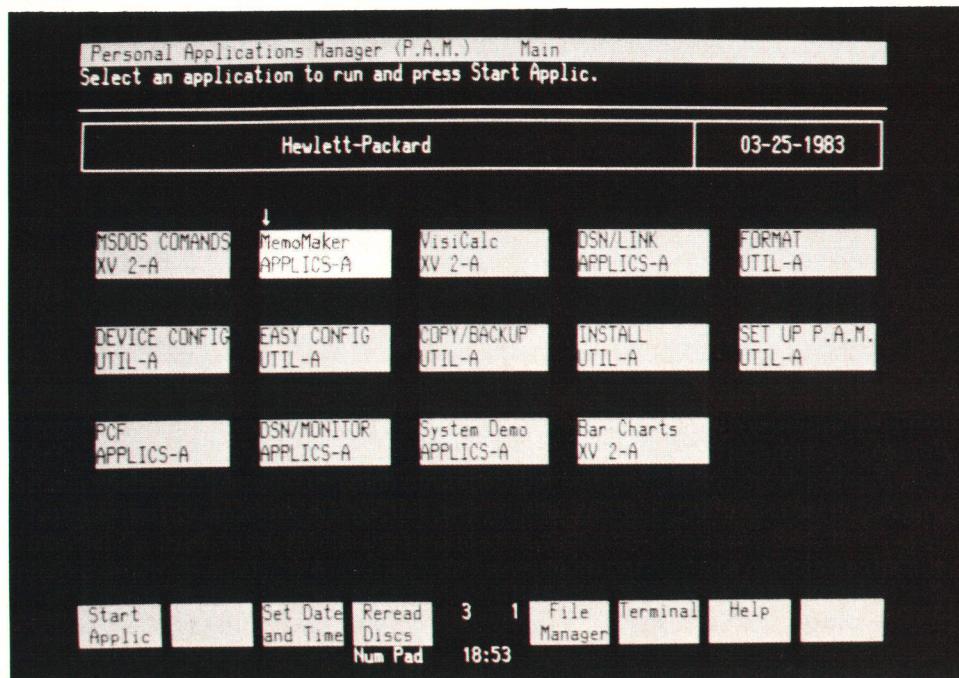
## File Manager

Another component of PAM is the File Manager, which provides the most common file manipulations in the same select-by-touch manner. The available commands are shown in touch labels on the screen, so the possibilities are always visible. Filenames are displayed in alphabetical order, with directories (which may contain other files) highlighted and listed first. The default display lists only the names of the files, but an expanded display is available which also shows the size and creation date and time for each file.

File Manager allows the user to copy files, to rename or delete them, and to browse through text files (reading them on the screen) or print them on any of the system printers. To aid in managing the large number of files that can be stored on a hard disc such as the HP 9133, File Manager also helps the user build, display, and (if empty) delete disc sub-directories. These can be thought of as file folders, each of which contains files and possibly other subdirectories.

Application programs chosen from the PAM screen can also access the File Manager. This gives the user access to the File Manager functions without terminating the application. It also provides a consistent file selection method for application programs, since file and directory names selected by the user can be passed back to the application. This is particularly useful when retrieving previously saved files. Since the File Manager displays the list of possible files, the user has less to remember. To help further, the application program can instruct the File Manager to show only files whose names match certain patterns. Since many application programs always use a consistent filename suffix as a default, this provides a simple method of limiting the files displayed to those most likely to be usable with the current program. The user can override this default to see other files.

To customize the File Manager display, application pro-



**Fig. 1.** The Personal Applications Manager screen is what the user normally sees after turning on the HP 150 Personal Computer.

grams can specify the contents of several of the File Manager screen labels; this helps remind the user of the specific task at hand while providing consistency among the application programs that use the File Manager.

### User Interface

While touch input is important to PAM and the File Manager, both are also designed to work well under keyboard control. It is often annoying to a fast typist to have to reach away from the keyboard, and it is also easier to take advantage of the HP 150's type-ahead feature by using the keyboard. The **TAB** key and cursor control keys are used to move a selecting pointer around on the screen; the **Select** key on the keyboard performs the same function as touching the item the arrow is currently pointing at. When a filename or subdirectory name is to be selected, the name can also simply be typed in. The function labels on the screen correspond to eight function keys at the top of the keyboard whose meanings are exactly the same as touching the screen labels. Many users find that they switch back and forth between the various selection mechanisms depending on what they are doing.

### Touchscreen Applications

Because it uses an industry-standard operating system, the HP 150 can run a variety of applications software programs, including the most widely used spreadsheet, word processing, and data base management packages. Most of the software for the HP 150 is offered by companies other than HP and may or may not use the HP 150's touchscreen.

HP-developed software packages for the HP 150 are designed to maximize the benefits of the touchscreen. These packages include MemoMaker, Personal Card File, Series 100/Graphics, and Financial Calculator. VisiCalc® for the HP 150, which was developed by HP under an agreement

VisiCalc® is a U.S. registered trademark of VisiCorp.

with VisiCorp, also makes extensive use of the touchscreen.

## MemoMaker

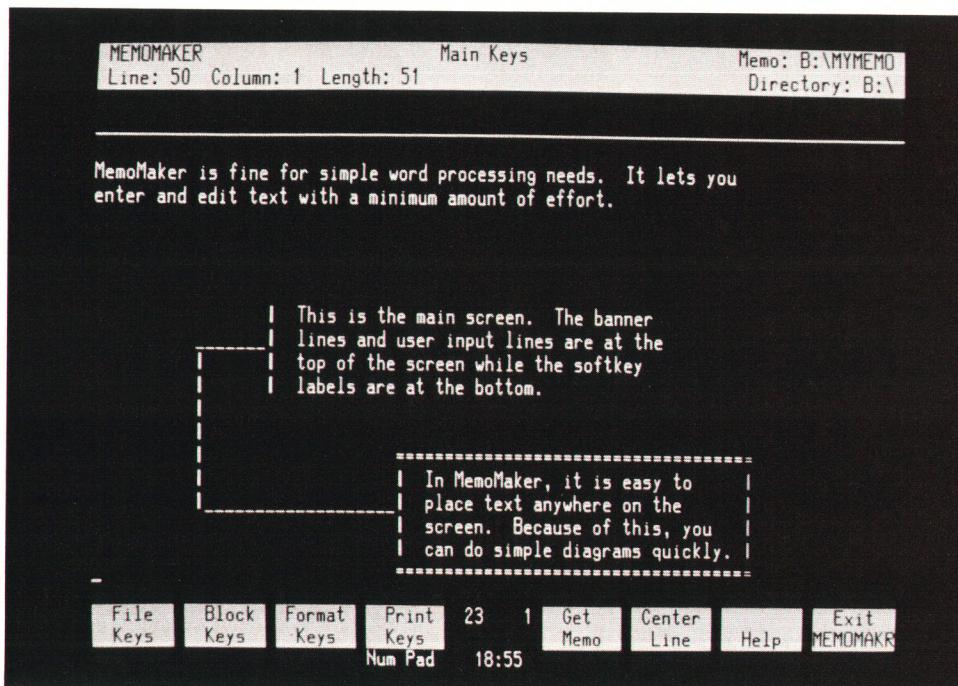
MemoMaker is a simple text editor with a few word processing features. It is designed for the user who needs to do simple text processing, but neither needs a complicated word processor nor desires to spend much time learning how to use one.

To keep MemoMaker simple, the number of features and options has been kept down. The user gets instant feedback in a "what you see is what you get" mode. Actions the user would naturally attempt work in a reasonable manner. For instance, the user is free to move the cursor anywhere on the screen and type. If the cursor is beyond the end of a line, the line is automatically extended with blanks to the cursor position. Moving the cursor into an area on the screen where there is currently no text is a very natural thing for the user to do, but many other word processors do not allow the user to do this.

In MemoMaker, the user creates text by typing, in either overwrite or insertion mode. When text is being inserted, existing text following the insertion is moved or wrapped to the next line so that no text is lost. Text that has already been created can be deleted, copied, or moved. By using the File Manager, or by explicitly specifying filenames, the user can save text or retrieve previously created text. It is easy to reformat blocks of text into nicely aligned paragraphs, underline words, or make text boldfaced. To show what text looks like, it can be printed on one of several printers with user-selected margins, tab stops, and various other options. A help facility is available at all times to explain the various commands.

### The MemoMaker Screen

The screen presented by MemoMaker consists of three



**Fig. 2.** On the MemoMaker screen, the display area is flanked by the banner and prompt lines (top) and the softkey labels and system information (bottom).

main sections (Fig. 2). At the top are four lines of banner and prompt information. At the bottom are the softkey labels and system information. In between, occupying twenty lines of the screen, is the main display area.

The main display area acts as a window on a scroll of text in the user's workspace. The text currently underneath the window is displayed on the screen as it would look when it is printed. Using the **Roll Up**, **Roll Down**, **Next** and **Prev** keys on the keyboard, the user can move the window up and down over the entire workspace. The **Home Up** and **Home Down** keys provide a quick way to display the first and last sections of the workspace.

The top line of the screen is the banner line for MemoMaker. This line tells the user that the program name is MemoMaker, what softkey group is displayed, and the name of the file that is in the workspace. This line is also where error messages appear. The next line provides information about the cursor's location within the workspace, how long the workspace is, and which directory is the default for file access. The next two lines are where prompts for input appear and where the user types in responses such as filenames. This top section is separated from the main display area by a solid line so the user won't confuse the messages and typing in this area with the workspace.

Just below the main screen are the softkey labels. These labels correspond to eight function keys on the keyboard. They also describe the action that will be performed if the corresponding key is pushed. However, the user will often find that it is more convenient to touch the label area on the screen, which is equivalent to hitting the corresponding softkey on the keyboard. Just below the softkey labels is a system information line with various indicators. The ones of particular interest to the MemoMaker user are the **Ins Char** and **CAPS** indicators. These tell if the program is in insertion mode (the default is overwrite mode) and whether the caps lock is on or not.

### MemoMaker Functions

In general, all actions that change the workspace are done by moving the cursor to the position on the screen that corresponds to a desired location in the workspace, and then typing or touching a softkey label. For instance, to enter new text into the workspace, the user moves the cursor to the desired place in the workspace, and then just types. The typing is automatically displayed on the screen and entered into the workspace at the corresponding location. Similarly, to delete a line of text, the user first moves the cursor to the desired line and then presses the **Delete Line** key on the keyboard.

This scheme of doing things at the cursor position is straightforward and needs almost no explanation. To move the cursor, the user can use the arrow keys on the keyboard or simply touch the screen at the desired position. The resolution of the touchscreen is not fine enough to distinguish individual characters, but it is good enough to get the cursor quickly to the general area on the screen. It is easy to move the cursor to the beginning of a line using touch, and this is frequently done.

The edit keys on the keyboard—**Insert**, **Delete**, and **Clear**—are used to perform the actions listed on the keys, although in some cases these are slightly different from the action of these keys on a terminal.

### Softkeys

There are five major groupings of softkeys. These are the main keys, file keys, block keys, format keys, and print keys. The main keys are presented when the user first enters MemoMaker. From this set, each of the four other levels of softkeys can be accessed. From each of the four other levels, the user can return to the main keys by hitting function key 8 (**MEMOMAKR Main**). The softkey tree is shallow to make it easier for the user to find a function. It has been observed that in products with extensive softkey hierar-

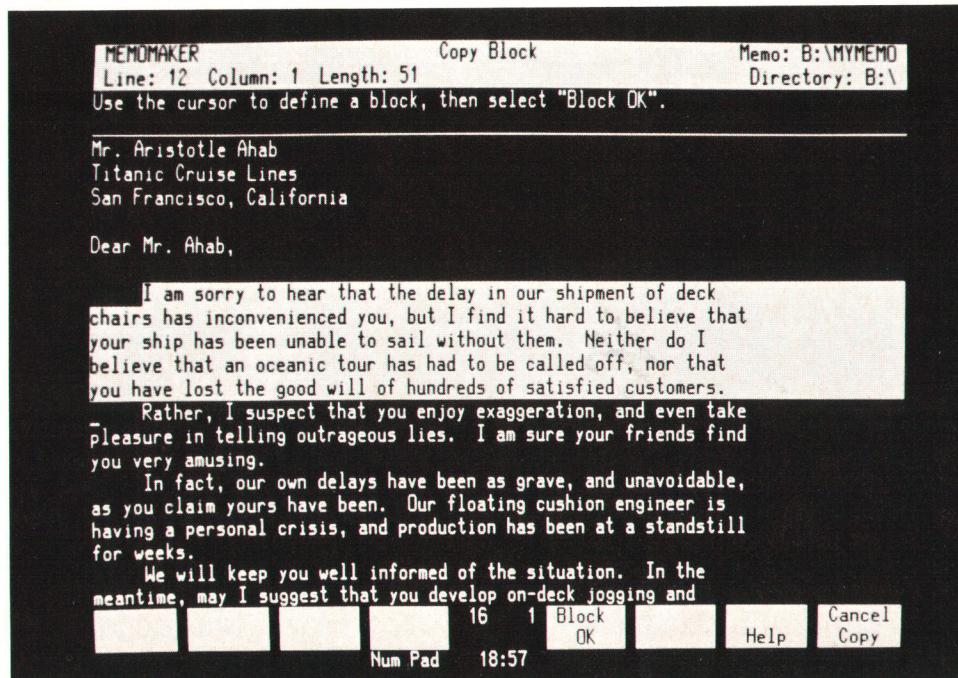


Fig. 3. As the MemoMaker user defines a block of text to be moved or modified, the text is highlighted in half-bright inverse video.

chies, it is easy for the user to become confused.

The file softkeys level contains functions for accessing the disc. These include the ability to save the workspace into a file or retrieve a previously stored file into the workspace. Also, there is a mechanism provided so that user-defined settings, such as tab stops, margins, and printer options, can be saved in a file and later retrieved. For instance, there are default tab stops every five columns, but the user can specify that there will be tab stops every four columns and save this information in a startup file. Then every time the user enters MemoMaker, the initial tab stops will be every four columns.

The format softkeys level contains functions to set margins and tab stops. When this key level is displayed, a ruler line in the prompt area at the top of the screen shows the current tab stop positions and margins.

The print softkeys level contains functions that allow the user to specify how the workspace is to be printed. There are options to double-space the printout, use a continuous feed (the default is to print a page at a time), insert page break commands into the text, or select one of several different printer devices. There is also a way to skip through the workspace to see where page breaks will occur without actually doing any printing (in a sort of preview mode).

The block softkeys level contains the traditional editing functions for copying and moving text. The user can also reformat text into aligned paragraphs or choose to underline sections or make them boldface. These commands all work on a block of text at a time. To define a block, the user moves the cursor to the beginning of the block, and then hits a softkey or touches the softkey label on the screen to activate the command (e.g., Copy Block). The user is then prompted to use the cursor to define a block. As the cursor is moved, all text between the start of the block and the cursor position is highlighted in half-bright inverse video (Fig. 3). This highlighted area is the text that will form the

block. Once satisfied with the definition of a block, the user hits the Block OK key, which completes the action. At any time before this, the user can cancel the block command using the Cancel key. This technique of defining blocks is powerful and very easy for the user to understand. Of course, the initial movement of the cursor to the start of the block and the final motion of the cursor to the end of the block can be done with touch, so a paragraph can be moved around in the workspace with just a few touches of the screen.

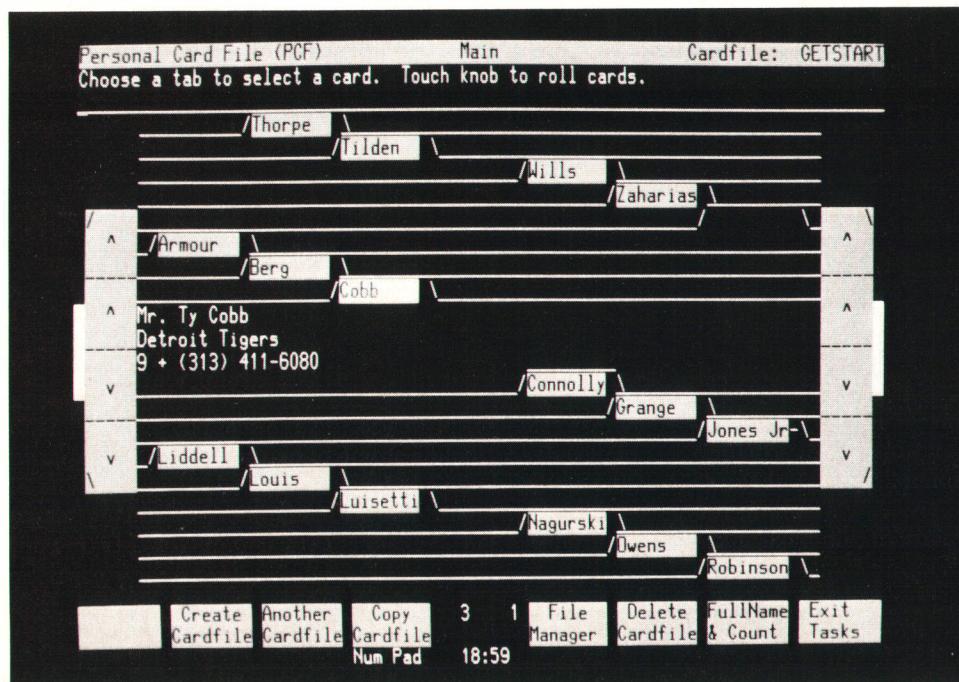
On all of the softkey levels, there is consistent key placement for similar functions. For example, function key 8 takes the user to a logically higher level. In the lower levels (e.g., file softkeys) this key causes a return to the main softkeys. From the main softkeys it causes an exit from MemoMaker. Similarly, the on-line help facility is accessed by function key 7 from all softkey groups.

#### Ease Of Use

To make MemoMaker easy to use, the capabilities included have been trimmed to prevent the user from being overwhelmed by options. An attempt has been made to be consistent and reasonable in the actions of commands, and to give the user feedback about what is happening. For instance, a Cancel key label tells what will be cancelled (Cancel Copy, Cancel Cut Out, etc.). Also, if an action could make sense, it is usually allowed, even when it is not the normal way. For example, to define a block, the user can start at the end of the block and move the cursor to the beginning instead of the reverse.

In some ways the design of MemoMaker was quite challenging. In achieving just the right balance between functionality and simplicity, some very hard decisions had to be made to keep MemoMaker a small, easy to learn and use product.

For example, there is no search and replace capability.



**Fig. 4.** The main screen of Personal Card File shows the image of a Rolodex card file with knobs, cards, and tabs that a user can touch.

MemoMaker doesn't allow the user to change the number of lines printed per page. It doesn't support columnar text manipulation. It doesn't have very extensive formatting ability (for instance, there is no subscripting). It doesn't do multiple windows. This has kept the number of softkey commands small.

## Personal Card File

Personal Card File (PCF) combines the convenience of a traditional desktop card file with the convenience of an automatic telephone dialing device and the power of a computerized information management system. There is nothing new about card filing, autodialing, or information management, but this combination of the three amounts to a new level of desktop management.

Almost everyone is familiar with the Rolodex rotary card files frequently found on business peoples' desks. To use a Rolodex card file, a person turns the knobs on the sides to roll the cards until the card with the desired information appears. This card can then be pulled out of the file, changed, and put back in the proper place. PCF has the ability to store and use information in the same way as a desktop Rolodex card file. Fig. 4 shows the main screen, on which PCF displays the image of a Rolodex card file with knobs, cards, and tabs that a user can touch. Touch a knob image and Personal Card File will smoothly roll the cards until the knob is released. Since all of the information on a card is not visible from the main screen, the user can touch a card to pull it out of the card file and see all the information.

Fig. 5 shows a card that has been pulled from a card file. A card contains descriptive text and data fields (the inverse video areas on the screen). The descriptive text is part of the card form, and is the same on all cards in the card file. The inverse video fields are filled with the information

typed onto the card. The card can then be modified and automatically reinserted in the card file.

### Custom Card Format

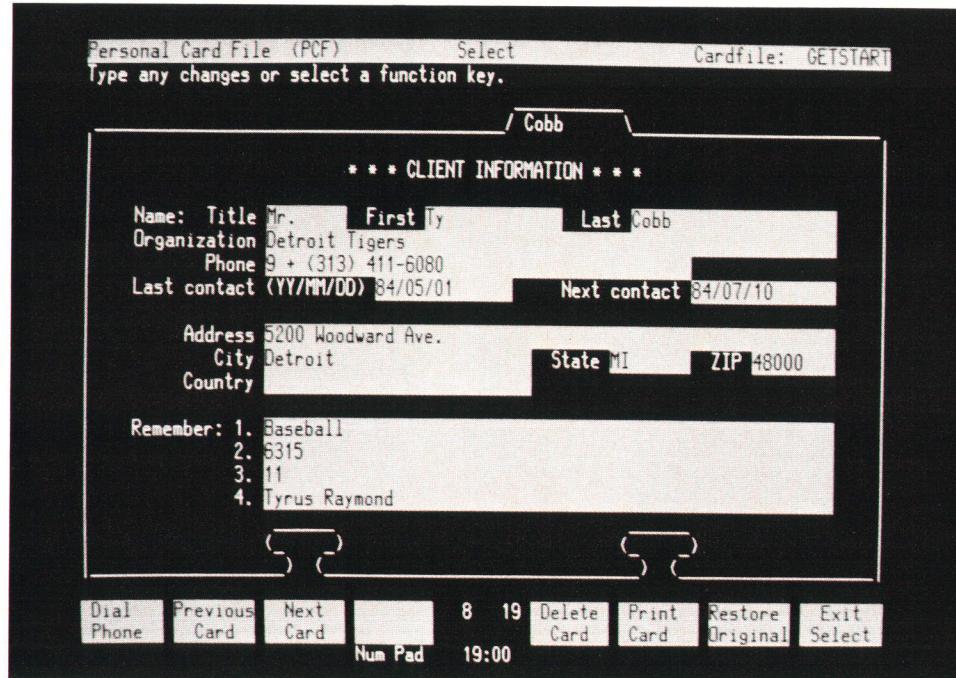
Each card file has a single card form associated with it, and every card in the card file uses this format. Fields and text can be placed anywhere on a card. This flexibility suggests uses far beyond the standard address file. Many sample card forms are included in the PCF product, and can be used as is, or adapted to fit the user's needs. During card design, a key field and an autodial field may be selected. Any field on the card can be selected as the key (for example, a catalog for a library may have the author, subject, or title field as the key) and will then be used to sort the cards in the card file. The information in this field also appears on the card tabs for easy identification of each card on the main screen. The autodial field is optional, and will contain the phone number to be dialed when a Dial Phone request is made.

Adding a card is as simple as filling in the fields and touching a softkey. While cards are being added, the card form is displayed. PCF will automatically insert the new card into its correct position in the card file.

Updating is as simple as adding a card. After selecting the card from the card file, the user types in any changes. The card will automatically be reinserted into the correct place in the card file.

### Searching a Card File

Often, only a subset of the entire card file is needed (for example, "all the Smiths in California"). Given a set of criteria, PCF will find all the matching cards. The card form is displayed for the user to indicate which fields to test and what values to look for. Wildcard searches and comparisons of numeric data make this a powerful feature. The result is a temporary card file (which can be saved as a permanent card file) that is a subset, or view, of the



**Fig. 5.** A card pulled from the Personal Card File.

original card file. Changes made to this subset will be reflected in the original card file.

Cards can be printed in several ways. Any card selected is displayed and can quickly be printed almost exactly as it is seen on the screen. When the entire card file is being printed, a card form is displayed and specific fields can be selected for printing. This is ideal for setting up mailing labels. The cards can be printed to a printer or to a file for later use. Any of these tasks can be done with a few touches to the screen.

Card files can easily be copied or appended to other card files. Information can also be transferred to and from PCF in a format suitable for use with other software, such as MailMerge®, CONDOR, and BASIC.

#### Autodial

A Hayes Smartmodem or other compatible modem allows PCF to dial phone numbers. PCF can dial the phone number in the autodial field of any card, which is specified at creation of the card file. For example, a card file can contain names and phone numbers of business contacts. To call one of them, all the user need do is press a button, pick up the phone, and talk.

The touchscreen reinforces the Rolodex card file concept of PCF. However, everything that can be done with touch can easily be done from the keyboard.

To make touching the screen as easy to use as possible, some indication of what has just been selected is given. For example, when selecting a card, the user can move a finger from card to card, and each time a new card is touched, the tab on that card will be highlighted. The last card touched remains highlighted. The **TAB** keys will also move the highlighted tab from card to card, and the highlighted card may be pulled out using the **Select** key on the keyboard.

MailMerge is a U.S. trademark of MicroPro International Corporation.  
"CONDOR" is a U.S. trademark of Condor Computer Corporation.

## Series 100/Graphics

Series 100/Graphics is a versatile charting application whose design pays special attention to the inexperienced user by greatly simplifying the process of graphics preparation. It enhances productivity by minimizing what the user has to know and remember, by offering a user interface that supports an intuitive understanding of what to do, and by providing a capability for data input from other popular software packages.

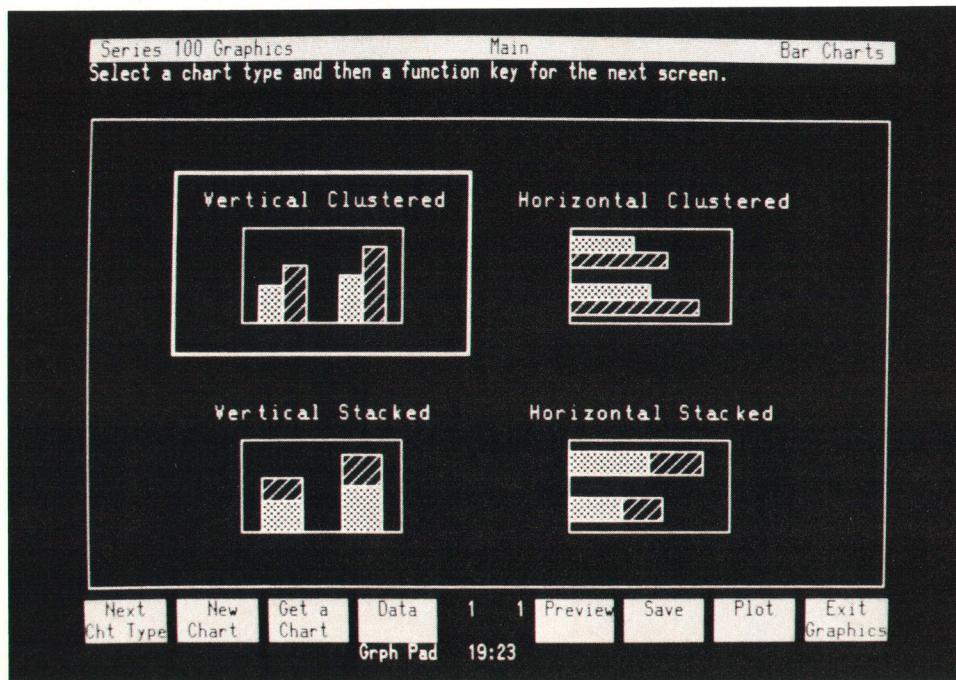
The product has been designed around the charting needs of the neophyte and the relatively infrequent user of graphics. Such a person is viewed as wanting to make a variety of high-quality presentation materials in each of the generic business chart types. Of equal importance, the chart designer is seen as valuing intuitive operation and ease of use over the complexity that so often arises with extended product functionality.

With these points as major guidelines, the software has been engineered to maximize the rate at which the user is able to perform productive work. The overall product is organized into four independent modules: text charts and three numeric charts—bar, line, and pie. Each is run by the user as a separate application, but all share the same essential functionality within a consistent user interface. Data and drawing specifications are fully interchangeable between the numeric chart types.

The major responsibility for knowing what to do lies with the software, rather than with the user. All decision points have preselected settings. As the chart designer chooses, these default choices can be overridden.

#### Product Contributions

The touchscreen capability of the HP 150 provides a nearly effortless selection mechanism with which to direct the software or make choices among charting options. All basic directions can be given by simply touching the screen



**Fig. 6.** Series 100/Graphics bar chart menu. By touching an icon, the user chooses between stacked and clustered charts and horizontal and vertical orientations. A frame indicates the user's choice. In this case, the choice is Vertical Clustered.

(Fig. 6). Only the data to be employed in drawing the chart and the text to be contained in the titles, legends, and footnotes need be entered through the keyboard.

The on-screen graphics feature of the HP 150 enables the charting application to be engineered for friendly interaction with the user. Many of the touchable areas of the menus are themselves graphical representations of the attributes to be applied to a chart. For example, line style and bar shading are determined by touching a sample pattern on the screen.

The current chart under preparation is displayed graphically on the screen just as it will appear when plotted (Fig. 7). Many components of this on-screen preview of the chart can be touch-selected to enable each to be designated as the item to which subsequent editing changes will be made or new attributes applied. To illustrate, an axis can be touched, then rescaled. A bar chart legend can be touched, then the corresponding bars redrawn in a new fill pattern. The segment label of a pie chart can be touched, then the adjacent slice redrawn "exploded" out from the rest of the pie diagram.

Each of the four modules has an associated sample chart, which serves as both a product demonstration and a learn-by-example model. This sample graph can be operated on by the novice as an easy way to learn the functionality and chart editing techniques of the software. When a chart module is used for the first time, the sample chart is automatically brought up as the current data. With as few as two touches to the screen, the example chart can be previewed, then plotted to paper, making this a comfortable way to gain familiarity with the product.

### Graphics Features

In addition to careful engineering of the user interface, Series 100/Graphics is also rich in functions. The inclusion of a text chart module as part of business graphics is

noteworthy. This charting module offers eight different fonts and eight text sizes. Effective, visually appealing word charts can be created, using the ability to intermix two fonts and two colors on any line of the plotted chart.\*

The numeric charts operate on a matrix of numbers of up to 64 rows by 5 columns. Data can be entered at the Data menu or brought into the application from a VisiCalc® model, a CONDOR data base, or a word processor text file. The application can be instructed to plot only a subset of the current data.

Chart files are quickly stored and retrieved. Data from one chart and plotting specifications from another can be matched. Thus, Series 100/Graphics not only works well with a one-time graphics task, but also is equally useful for the recurring preparation of periodic charts.

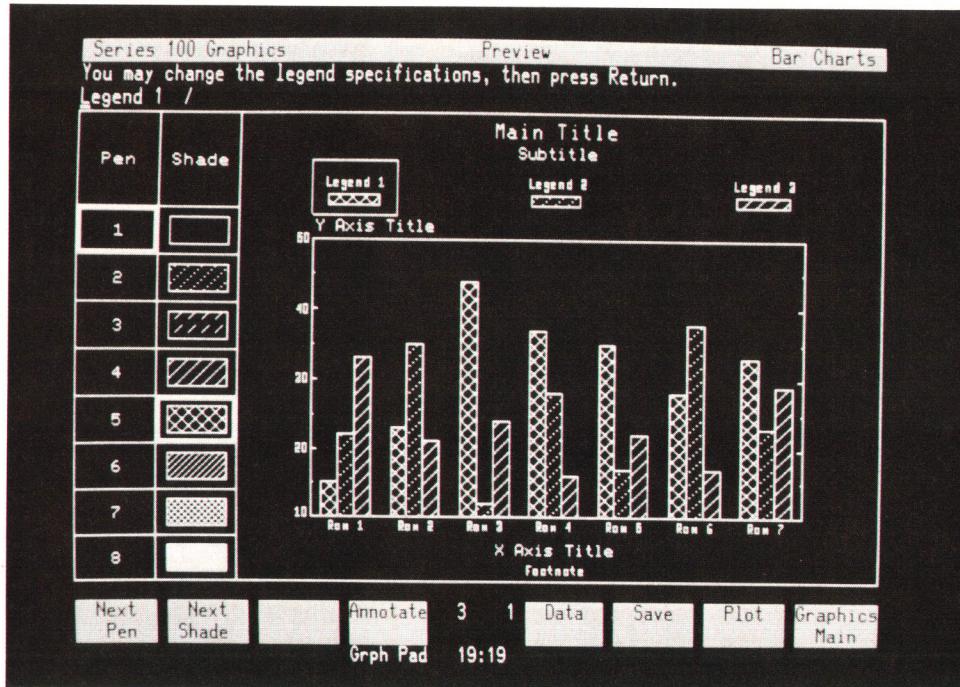
It is possible to annotate all charts with lines or text. Simple symbols such as arrows and boxes are readily constructed from lines. Both forms of annotation are positioned on the chart with the aid of the graphics cursor control keys.

For hard-copy output, Series 100/Graphics allows the user to select exactly what is to be plotted—any combination of the data, titles, legends, axes, or grid lines. It is possible to position a chart oriented either vertically or horizontally on any portion of four different sizes of plotting media. It can be rendered on paper or on transparency film for overhead projection. A chart can also be displayed on the full screen of the computer. This image can be photographed, sent to the HP 150's internal thermal printer, or directed to a compatible printing device.

## Financial Calculator

Although computers usually have significantly more computing power than calculators, most people prefer to use a calculator to perform a few nonrepetitive computa-

\*The different colors don't appear on the HP 150's monochromatic display, of course.



**Fig. 7.** A chart being prepared is displayed on the screen just as it will be plotted. A frame surrounds the screen element that is currently active (Legend 1). Touching another title, axis, or legend makes that item subject to modification. A prompt on the second line guides the user.

tions. Computers need programs to perform even simple arithmetic calculations, and most existing programs require users to learn an awkward syntax to enter computations. These programs have never gained much popularity.

Financial Calculator is a program that exploits the touchscreen and the graphics features of the HP 150 Computer to mimic the HP-12C Handheld Calculator. The user sees a familiar calculator on the computer screen and touches its keys to operate it, just as one would operate a conventional calculator. Consequently, Financial Calculator is as easy to use, and as easy to learn to use, as an ordinary calculator. Some calculator programs for microcomputers have presented similar images on the computer screen, but have not provided a natural means for manipulating the calculator keys.

The HP-12C Calculator has many features that have contributed to its success. In addition to the usual arithmetic operations, the HP-12C has one of the most complete sets of financial and statistical functions of any business calculator. Included are present and future value, internal rate of return, amortization, depreciation, and standard deviation. The calculator is programmable and can accept up to 99 program lines. Continuous memory preserves data and programs in the HP-12C while it is turned off.

Financial Calculator is an extremely faithful simulation of the HP-12C. The user sees on the HP 150 screen an HP-12C image that has the same keyboard layout and a similar numeric display (Fig. 8). When the user touches the keys displayed on the screen, Financial Calculator responds just as the HP-12C would. An HP 150 disc file saves programs and data between sessions, just as continuous memory does on the HP-12C.

Most important, however, Financial Calculator exactly duplicates the functions of the HP-12C. The program includes a precise copy of the HP-12C software and an emulator to interpret it. This combination ensures not only

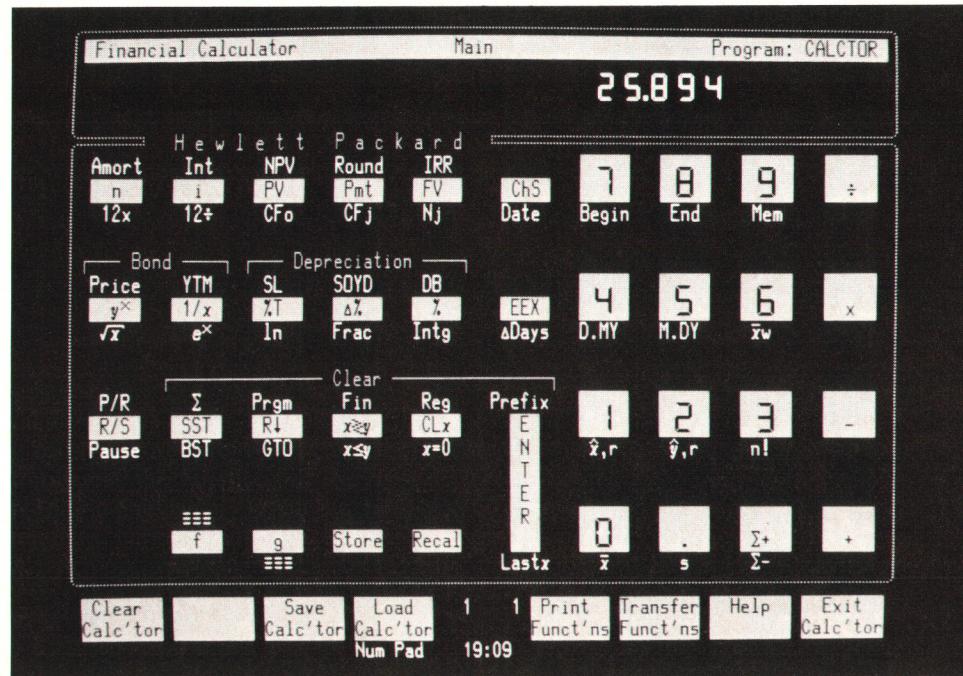
that the computer application and the calculator will perform the same functions, but also that they will produce exactly the same results with exactly the same precision. Customers who purchase Financial Calculator even receive the same manual that is supplied with the HP-12C. This design approach reduced the investigation, coding, and testing time needed during the development of the product and benefits users who are already familiar with Hewlett-Packard calculators.

HP 150 owners are not likely to discard their HP-12C calculators! After all, an HP 150 computer will not fit in a briefcase or operate on an airplane. There are reasons, however, why people who own HP-12Cs will want to buy Financial Calculator. The HP-12C can only store one program and one set of data in its continuous memory. Financial Calculator can store and quickly reload any number of programs and sets of data from HP 150 discs. In addition, Financial Calculator has a provision for passing numbers it has calculated to other HP 150 applications such as spreadsheets, data bases, and graphics programs. Numbers can also be output to printers connected to the HP 150.

Financial Calculator combines the power of a microcomputer with the simplicity of a calculator. The HP 150's touchscreen and graphics hardware make its operation natural and easy to learn. The HP 150 provides printing and data storage not usually found on calculators. Financial Calculator complements other HP 150 applications, contributing to a powerful and coordinated software system.

## VisiCalc®

VisiCalc® for the HP 150 was developed under an agreement with VisiCorp by a joint project team from HP's Personal Office Computer Division and HP's Office Systems Cupertino Operation. The goal of the project was to produce



**Fig. 8.** The Financial Calculator is a faithful simulation of the keyboard of the HP-12C Handheld Calculator.

a product that would run on both the HP 150 using touch and on a block mode terminal connected to the HP 3000. The joint development effort had the goal of making these products as compatible as possible. The second goal of the project was to provide HP personal computer and HP 3000 users an extended powerful spreadsheet capability compatible with VisiCalc®.

VisiCalc® traditionally has been driven by a powerful, yet somewhat cryptic command set. In designing the interface for the HP 150 version, the team was faced with the requirement to keep backward compatibility for current users and at the same time make the product easier to use for occasional nontechnical users. To accomplish this, it was decided to keep the current command interface as one of two different ways to interact with the product. The second method, a softkey interface, was added. These softkeys are screen-touchable and provide access to all features available using the traditional and extended commands.

On the screen, VisiCalc® 150 cells are one row high and from three to 77 characters wide. The limited height might have made it difficult for inexperienced users to touch the desired cell on the first try, so experiments were conducted to help define how applications should use touch so that users can be effective immediately. When the screen is touched, the application provides feedback to show the user which cell will be chosen when the touch is released. The user's actions are:

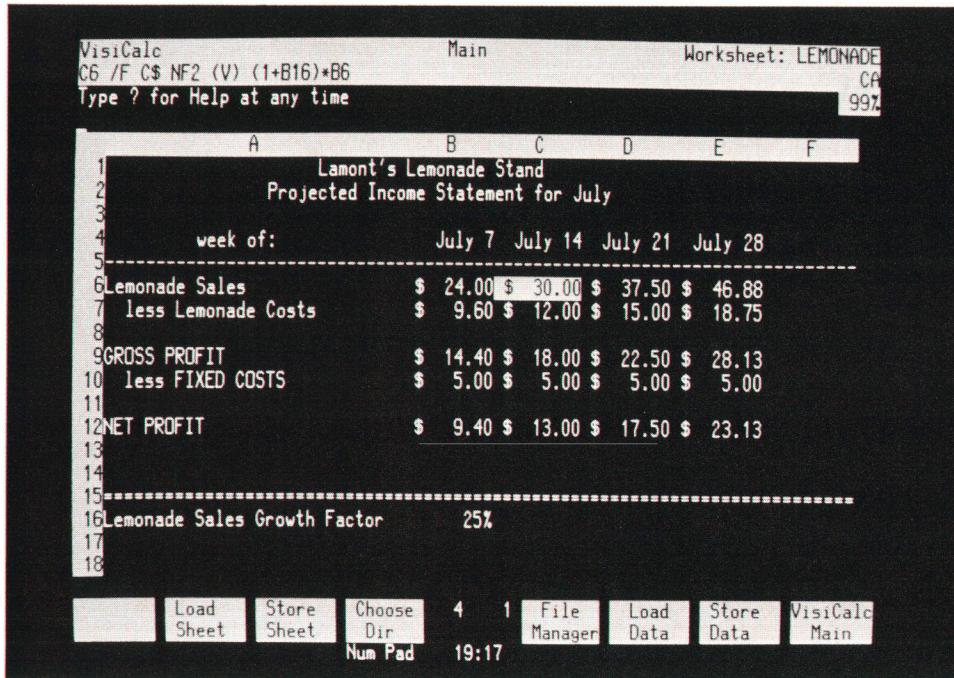
- Touch near desired choice
- Move finger on the screen to adjust the choice if necessary
- Release finger to accept the choice.

Touch is integrated into the program in a natural way. The original VisiCalc® contains the concept of a cell reference. Anywhere a user can enter a cell reference in standard VisiCalc®, VisiCalc® 150 will accept a screen touch. The

user can move anywhere on the worksheet by touching the desired cell. It is much quicker, of course, to touch a cell than to specify the cell in a goto command.

#### Acknowledgments

The authors wish to thank Steve Weiss, Kyle Hurlbut, Sherry Ramsey, and Patrick Apfel for their contributions to the development of Personal Card File, and John Taylor, Bob Farnham, and Barbara Packard for their contributions to MemoMaker. The Series 100/Graphics product team included project managers Jim Sutton, John Alburger, and Sharon Mead, design engineers Kim Becker, Lori Farnworth, Mike Loveridge, Ric Mears, Tom Neitzel, Steve Pacheco, Bruce Rafnel, Mike Van Riper, and Kevin Wentzel, technical writers Elaine Hander and Annette Crawford, quality engineer John Miles, localization engineer Henry Gage, and marketing engineers David Obershaw and Linda Lazor. Sherry Ramsey was the project manager for Financial Calculator, Patrick Apfel was the product manager, and Bob Burns was a project engineer. VisiCalc® 150 is the result of efforts from people from two divisions and several functional areas. The product team included development engineers John Funke, Scott Hanson, Jack Morgenstern, Doug Smith, and Kathy Yokota, technical writer Trish Egger, quality engineer John Miles, localization engineer Enrique Castillo, and product managers Linda Lazor and Mike Volman.



**Fig. 9.** VisiCalc® 150 incorporates touch to speed spreadsheet calculations.

# Hardware Design of the HP 150 Personal Computer

by John E. Watkins, Patricia A. Brown, George Szeman, and Susan E. Carrie

**T**HE HP 150 IS DESIGNED to be a personal computer primarily for business users. This requires hardware that provides the wide set of features needed in a business machine. The HP 150 is also a part of HP's Information Productivity Network (IPN), so it is capable of acting as a terminal in an HP 1000 or HP 3000 Computer System. To provide hardware flexibility tailored to specific needs, the HP 150's hardware is upgradable by users. The hardware plays a major role in providing a personal computer that is easy to use and maintain. Cost sensitivity, of course, was also a factor influencing the hardware design.

## Hardware Features

Fig. 1 lists the HP 150 hardware features tailored to the needs of the business computer user and the requirements of an IPN node. These feature sets have a great deal of overlap and are not exclusive. Fig. 2 is the HP 150 hardware block diagram.

MS-DOS, the operating system, requires an 8086-type microprocessor. An 8088 running at 8 MHz is used in the HP 150. Because of its 8-bit external bus, some board area and cost savings can be realized with an 8088-based system, with a decrease in performance on CPU-bound operations, compared to a comparable 8086 design. In most applications, however, a personal computer system is performance-limited because of I/O or disc speed and not because of CPU speed.

To support a variety of flexible and fixed-media disc drives, the HP-IB (IEEE 488) was chosen for the disc interface. The HP-IB also provides a means of communication with peripherals such as printers and plotters.

The basic HP 150 system comes with 256K bytes of user memory. A portion of the system data storage area, the operating system, and the application software are in this user memory area. This large amount of memory is more than adequate for running most applications, but the hardware allows expansion of user memory up to 640K bytes of RAM to allow large spreadsheet screens, large compiled programs, and "RAM discs" (see article, page 6).

A real-time clock/calendar maintains the month, date, day, and time. The system displays the time on the status line of the screen and the operating system uses the clock

as a reference to time stamp files when saving them on disc. Application programs that serve as a desk calendar, recording appointments and reminders, can also use the clock features. The clock has battery backup, so the user is not prompted for the time and date each time the system is turned on.

## Terminal Features

Terminal features of the HP 150 were selected to allow it to fit into the IPN product strategy, that is, to support existing HP 3000 software packages with a display terminal structure like that of the HP 2623A Terminal. This terminal architecture has separate alphanumeric and graphics display subsystems.

The HP 150 communicates with a host computer through serial data communications. Two independent datacom channels allow the HP 150 to be connected to a serial printer or plotter on one port and to another computer on the other port. One port, a general-purpose datacom port, supports either RS-232-C or RS-422, while the other port is an RS-232-C port generally used for peripheral support, although it too can connect to other computers or modems. Both ports communicate at up to 19,200 baud.

## Expansion Capability

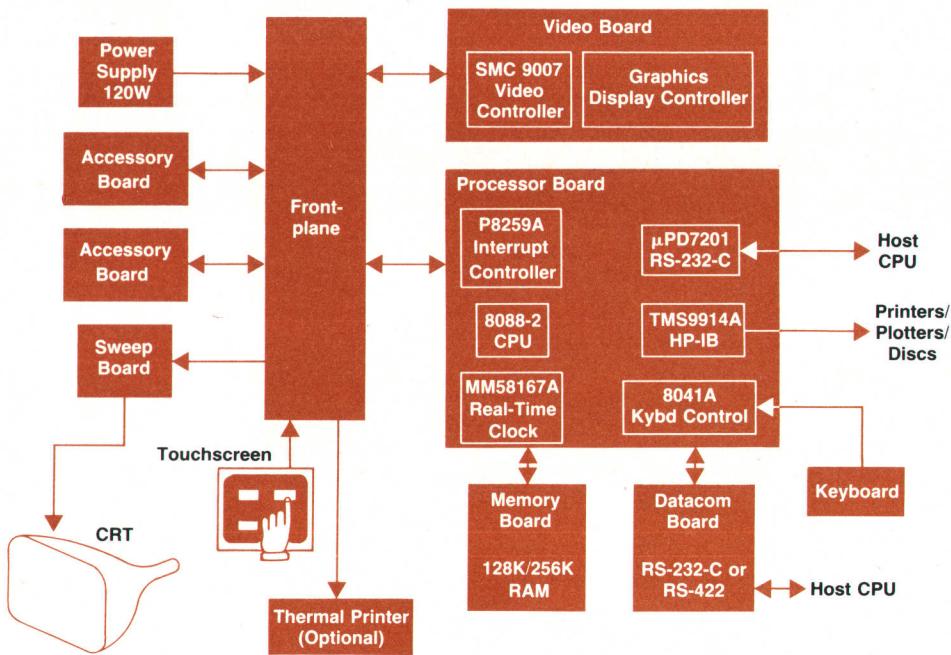
Although the standard feature set of the HP 150 is substantial, it is clear that it will not satisfy the needs of all users—hardware expansion capability is necessary. In providing this capability several goals were identified:

- Allow accessory cards to be integrated within the package, i.e., no extension boxes required. This must be done within the packaging constraints of the small footprint.
- Provide a physical and electrical interface to the accessories that will satisfy the requirements of a large variety of devices. Such devices include modems, memory expansion, language localization, and boards with ROM resident code.
- Make installation of the accessories simple and cost-effective.

After weighing several options, we decided to provide two internal accessory slots. The slots reside below the processor card and communicate with the processor over

Business Computer Features	Terminal Features
8088 microprocessor Touchscreen Large memory (256K to 640K) HP-IB (IEEE 488) Two accessory slots Real-time clock	80-column-by-24-row alpha display Two softkey rows, one status line Five alpha character enhancements Eight distinct alpha character sets Multiple-page alpha display 512-by-390-pixel bit-mapped graphics RS-232-C and RS-422 datacom support

Fig. 1. HP 150 hardware feature set.



**Fig. 2.** HP 150 hardware block diagram. The CPU is an 8088 microprocessor running at 8 MHz.

the frontplane. These I/O cards are each allocated a 64K block of memory within the processor's address space. These address blocks are decoded on the memory board, which generates slot select signals that are sent to each slot. This relieves the accessory cards of the burden of decoding and removes restrictions on the placement of boards. The firmware also provides a search-and-identify algorithm, which allows an accessory to execute ROM resident code. Finally, the slots are designed to allow the user to install the accessories. This is done by removing a backplate and sliding the option in from the rear. This reduces manufacturing and inventory costs by eliminating the need to load options at the factory. It also allows users to update their systems in a cost-efficient manner after they have purchased them. The internal printer is also designed to be installed by the user.

#### Processor I/O Operations

The 8088 microprocessor accesses the HP 150 I/O system through memory and I/O mapping of the various devices. The keyboard and the touchscreen are controlled by an Intel 8041A single-chip microcomputer, which acts as a slave processor to the 8088. The 8041A contains integral ROM and RAM, a time/event counter, and two 8-bit I/O ports. A 7201 dual-channel UART (universal asynchronous receiver/transmitter) controls the data communications ports, and HP-IB peripherals are accessed under the control of a TMS9914A controller chip. Communication with the optional integral printer is made via a simple interface on the frontplane board that handshakes bytes of data to be printed. Each of these devices is accessed through the 8088's I/O address space.

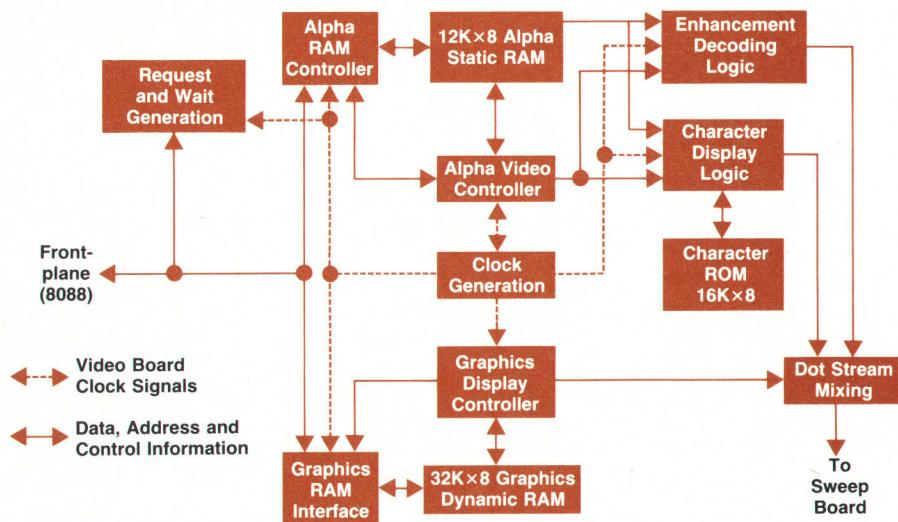
The video subsystem and the accessory slots are accessed through the memory address space of the 8088. Quick transfers of blocks of data to an accessory device or to alpha or graphics memory are facilitated by this organization. To simplify implementation, the system status lights (LEDs)

mounted on the mezzanine memory board are also memory mapped.

#### Video Board

Fig 3 shows the major components of the video board and the flow of data and control information between them. The following is a brief description each of these components.

- **Processor Wait and Request Generation.** Decodes 8088 access to the alpha RAM, video controller register, or graphics RAM and produces signals for the RAM controllers. Also inserts necessary processor wait states.
- **Graphics RAM Interface.** Interfaces the 8088 and the graphic display controller chip (GDC-3091) to the graphics RAM.
- **Graphics RAM.** A  $32K \times 8$  block of dynamic RAM is used for storing graphics information. Because the graphics display is a bit-map display, the information is stored in that format. A portion of the RAM is also used by the alpha video firmware for storing variables.
- **Graphics Display Controller.** A custom gate array graphics controller chip retrieves data from the graphics RAM and forms the graphics dot stream. The chip also provides timing signals for the graphics RAM.
- **Alpha RAM Controller.** Interfaces the 8088 processor to the alpha RAM and to the SMC 9007 video controller. Also interfaces the SMC 9007 to the alpha RAM.
- **Alpha RAM.** A  $12K \times 8$  block of static RAM stores the alpha information to be displayed.
- **Alpha Video Controller.** The SMC 9007 VLSI video controller retrieves data from the alpha RAM and provides the signals needed to generate the alpha dot stream. The chip also generates raster signals to the sweep board.
- **Character ROM.** A  $16K \times 8$  ROM contains the template for each alpha character. The ROM contains eight character sets: Roman, Roman extension, bold, bold extension, italic, italic extension, line drawing, and math. Each set



**Fig. 3.** Major components of the video board.

contains 128 characters.

- Character Display Logic. This hardware uses information fetched by the SMC 9007 from the alpha RAM. It gets the template for this from the character ROM, and with other signals from the SMC 9007, the alpha character dot stream is generated. This stream does not include any enhancement information.
- Enhancement Decoding Logic. This hardware decodes the enhancement information fetched from RAM by the SMC 9007. It also relies on enhancement signals provided by the SMC 9007 to produce control signals used by the mixing hardware.
- Dot Stream Mixing. This hardware mixes the alpha and graphics dot streams and selects enhancements as directed by the control signals from the enhancement decoding logic. The final dot streams are full bright and half bright, which are sent to the sweep circuit.
- Clock Generation. Produces various clock signals used throughout the video board.

### Compact Electronic Design

Providing the system feature set presented several challenges, but chief among these was the small footprint package. The package provides room for three layers of boards that communicate over a frontplane through pin and socket connectors (see Fig. 4). There is also room for one layer of boards between the top two main board layers. Boards in this space are called mezzanine boards and communicate with the boards immediately below them through dedicated connectors. The bottom board space is allocated to the accessory cards. The upper board houses the video circuitry, and the middle board carries the processor and other LSI chips. The standard system ROM and RAM reside on a mezzanine memory card. A second mezzanine card supports the RS-232-C/RS-422 communications port. These cards interface directly with the processor cards.

To reduce the IC count on the video card, a PLA (programmable logic array) and a TTL gate array are used. The gate array implements most of the circuitry of the graphics controller section, including control of the RAM. Compared to discrete circuitry, the gate array consumes one fifth the space, one fourth the power, and one half the cost. The

PLA is used to perform the alpha enhancement decoding, and reduces the board area used by a factor of 2.5.

Board space was at a premium during the design of the RAM controller for the mezzanine memory board. The RAM and the system ROMs are both housed on this board. Because of the small space and the irregular shape of the board, a commercial LSI controller with its support circuitry would not easily fit. A discrete controller was implemented; it puts the 8088 in a hold state (where it stops accessing external devices) while four consecutive rows in each dynamic RAM get a refresh cycle. Every 56 microseconds, another four consecutive rows get refreshed. This controller implementation reduces processor bandwidth under worst-case situations by only 4.5% and costs less than an LSI controller implementation. A frontplane signal sent to the accessory slot connectors allows any installed memory expansion board to synchronize its refresh with that of the standard RAM. This eliminates the performance degradation that would occur if the expansion memory were to do its refresh asynchronously with the other board; it also reduces the refresh circuitry required by the accessory.

Performance was also an issue of concern in the video subsystem. To prevent long processor holdoffs, no row buffers are used in the alpha section (double row buffers with DMA could not be considered because of space considerations). Unfortunately, this created several tight timing margins, introducing the need for techniques not usually used with a VLSI implementation. These techniques include skewing the clocks with buffers to provide more time to decode signals or to increase hold-time margins. Hold times for the data sent from the RAM to a register are increased by chaining the output enable signal of the RAM with the clock signal of the register receiving the data. Some control signal pulses are stretched by clocking the signal into a flip-flop and then using the clear input to remove the signal (see Fig. 5).

### Product Design

For the HP 150 to have a friendly, nonoverpowering appearance, it had to be compact. This created space efficiency requirements, which had a major impact on the hardware design.

## Software Graphics in the HP 150

The HP 150 has a software vector generation graphics system. This differs from the hardware graphics systems in most previous HP products. When hardware generation is used, the graphics subsystem is supplied with information describing a vector and the hardware then calculates each dot of the vector and writes it into RAM. For example, the processor may give the graphics system an (x,y) coordinate pair, a slope, and a vector length. In the HP 150, the processor and the system firmware do the entire job of calculating each dot of a vector and writing it into RAM.

Because there was concern about the performance of the HP 150 compared to other HP graphics products, studies were done with three products:

- HP 150 Personal Computer with software vector generation
- HP 2628A Terminal with half hardware, half software vector generation
- HP 2623A Terminal with hardware vector generation.

The HP 2628A has registers in hardware that allow the processor to write to the graphics system to cause an increment in vector calculations. The hardware actually writes the dot into RAM.

The first study was a calculation of dot drawing speed. In these calculations, it was assumed that each dot of the vector must be computed and written into RAM separately. In hardware graphics systems, this is generally true all the time. In the HP 150, this may not be the case. For example, horizontal vectors can be written into RAM eight bits at a time on the HP 150 (these vectors were not considered in this study). The results are shown below:

HP 150	24 $\mu$ s/dot (typical)
HP 2628A	5.75 $\mu$ s/dot
HP 2623A	5 $\mu$ s/dot.

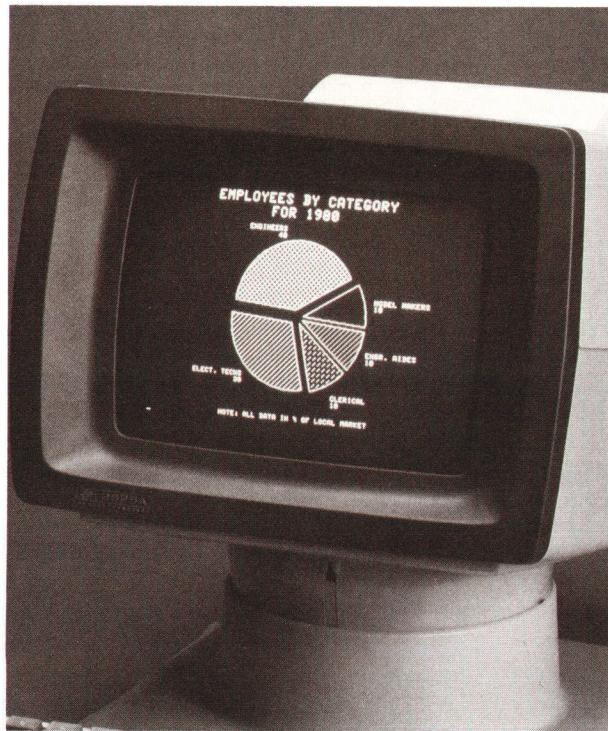


Fig. 1. Pie chart used for graphics speed comparison.

These numbers indicate an incredible difference in expected drawing speed on these products. To find out what kind of difference the customer would perceive, there was a second study. This study put full screens of graphics on these products via datacom. Information was sent via datacom because all three products can receive information that way.

This second study compared two different screens drawn on each of these products at two different baud rates. The two screens drawn are shown in Figs. 1 and 2. The pie chart is typical of business applications and the picture of a gear has a variety of vector types. A demonstration pod was used to send the information. This pod attaches to the RS-232-C port and can send information at any given baud rate. The handshaking was XON/XOFF. Here are the results:

Screen	Gear Drawing	Pie Chart	
Baud Rate	9600	19,200	9600
HP 150	8.4 s	8.1 s	7.2 s
HP 2628A	8.1 s	6.2 s	7.1 s
HP 2623A	7.9 s	*	7.4 s

\*The 2623A does not operate at 19,200 baud.

The conclusion of the second study was that, under these constraints, the difference in time for drawing a screen on these products is, for all practical purposes, negligible. The extra cost and space associated with hardware vector generation do not pay off in extra perceived performance in general business applications.

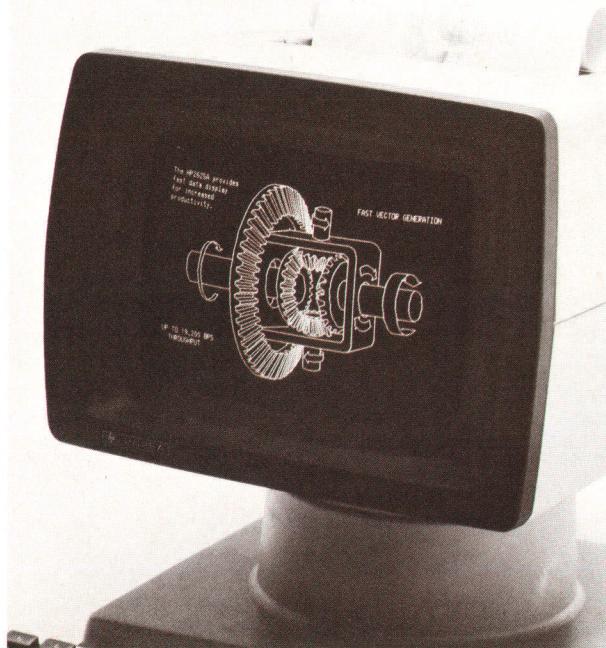
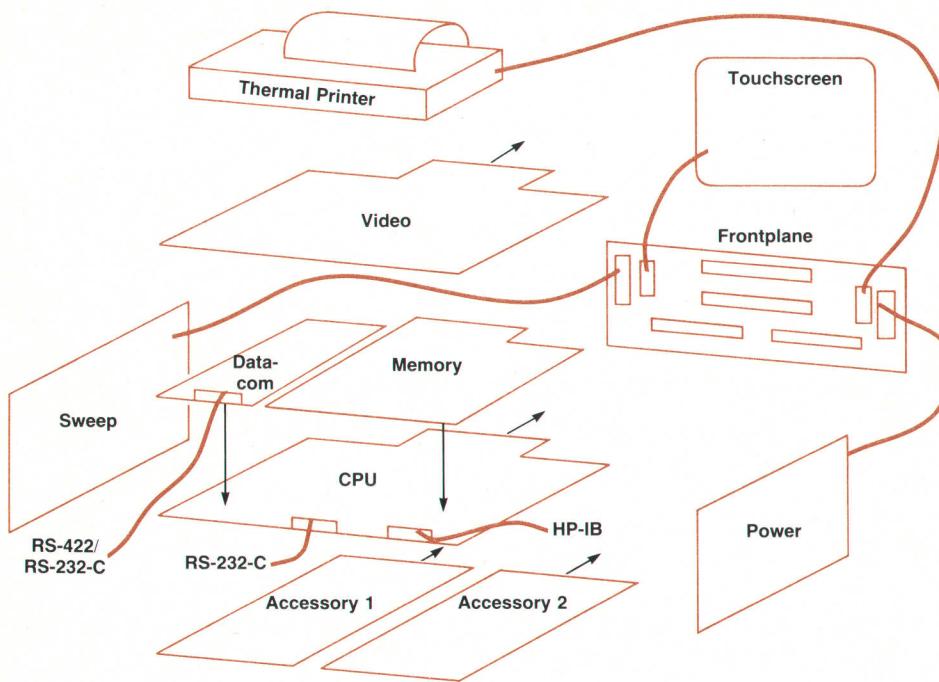


Fig. 2. Gear drawing used for graphics speed comparison.



**Fig. 4.** The HP 150's small footprint required a compact board arrangement and interconnect scheme.

One effect of the space constraints is the relatively small nine-inch CRT display. Much effort was put into creating a high-quality display. To produce sharp, legible characters, additional attention was given to the video board, sweep board, and CRT. On the video board, half-dot shift is used to generate smooth diagonal lines, while dot stretch helps thicken dots to produce brighter characters. The sweep board was modified to further increase character contrast. As recommended by HP ergonomic studies, the HP 150 has a dark-bodied green CRT for good contrast and minimal eye strain. Much work also went into meeting European ergonomic standards for the display.

### Manufacturing Considerations

Design for manufacturing is important to the success of any product and was crucial for the HP 150. Much time was spent understanding the manufacturing process, and design changes were made to fit into that process. For example, components were selected and positioned to allow maximum use of autoinsertion. Printed circuit board traces were moved away from pins to prevent solder shorts during wave soldering.

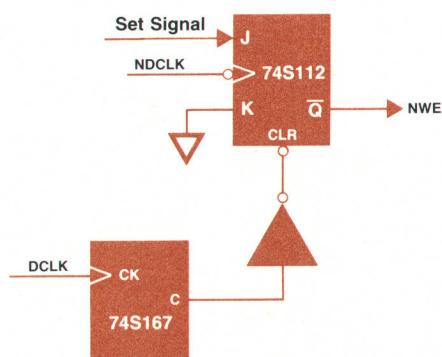
Much effort was put into the self-test software to assist the production line in testing units. Part of the self-test allows remote data collection and monitoring. This remote monitoring ability was used extensively during the development cycle. Hundreds of units were placed in a strife chamber, which subjected the units to hot and cold temperature extremes. An HP-85 Computer outside the chamber communicated with each unit in the chamber to log any failures that occurred while the unit was running its self-test. This strife testing is still used on an auditing basis to ensure the reliability of the units being assembled.

Ergonomics was also a major objective in the design of the user interface. To give users a comfortable feel, the keyboard has a low profile, adjustable tilt, and sculptured

keys. This device is an impressive achievement of mechanical design (see article, page 34). Keys are well laid out and have just the right amount of stiffness and tactile feedback.

The touchscreen provides users with a simple interface to applications. Instead of spending time memorizing computer commands, users should be productive with their PCs immediately. With the help of the touchscreen, commands can be entered just by touching highlighted areas on the screen, allowing applications to be menu-driven and simple to learn and use.

To fit well into home and office environments, the product has to comply with various safety and regulatory standards both in the U.S. and abroad. Because the PC is aimed at the home market as well, an extra level of safety is built into it. The package is designed to prevent possible hazards to users in the home environment. Several safety agencies, including UL (U.S.A.), CSA (Canada) and FEI (Finland), have inspected and approved the design. ESD (electrostatic



**Fig. 5.** To meet tight timing requirements, some control signal pulses are stretched by clocking the signal into a flip-flop and using the clear input to remove the signal.

discharge) limits have also been observed; the unit can withstand up to 15,000 volts of static discharge without loss of data or damage to components. Much effort was spent on making the unit comply with FCC level B (U.S.A.) and VDE level B (Europe) electromagnetic interference regulatory limits.

### Acknowledgments

The authors would like to thank Howard Lee for his contribution to the hardware design efforts as project manager. Thanks also to Robert Heyde, Eric Smith, Bob Ko, and Rich Wales for their contributions in hardware design, quality assurance, and meeting product regulations.

## Personal Computer Printer Is User Installable

by Joseph D. Barbera

**T**HE HP 2674A PRINTER (Fig. 1) is designed specifically for use as an option for the HP 150 Personal Computer. It features quiet printing, autoloading of roll paper, user installability, and a zero footprint.

The printer is a wedge-shaped device (Fig. 2) that fits into the top of the HP 150 cabinet and straddles the yoke of the HP 150's cathode-ray tube. One cable connects the printer to the computer.

Thermal printing was chosen for its inherent simplicity and quietness. The printhead, which is already used in another HP printer, was modified for 12V service, and an additional interstitial wear layer was added to provide life in excess of 100 million characters.

### Mechanical Design

One of the printer's unusual features is its soft, round platen. This provides a printing surface as well as the paper motion device. The low durometer rating of the platen was picked so that the printhead depresses it, making it look like a flat platen. This is one of the keystones of the design, allowing for the compactness and simplicity of the printer.

A sheet-metal chassis is used instead of plastic because of the shorter tooling lead time. A low number of moving parts and a low total parts count lead to high reliability and low assembly costs.

The customer receives the printer in a separate box and simply drops it into the computer and depresses a pair of quick-latches. The single cable has a 50-pin connector that includes 12V, 5V, data I/O, and ground connections. Paper is inserted until detected by a sensor, and then the printer completes the loading sequence automatically.

### User Troubleshooting

To make the printer completely user installable, it was necessary to provide some means for the user to isolate a problem to either the printer or the HP 150. Two light emitting diodes and a pushbutton switch are provided. One LED monitors the power supply voltages and will not be lighted if the cable is not properly connected or if there is a power supply problem. The other LED monitors the

data strobe input to the printer and blinks when data is being successfully transferred. The pushbutton switch initiates a local confidence test.

### Electronics and Firmware

The printer has microprocessor-controlled electronics to



**Fig. 1.** The 2674A Printer is a thermal printer that fits into the top of the HP 150 Personal Computer cabinet.

minimize the processing load on the HP 150 processor. The HP 150 communicates with the internal printer via an eight-bit TTL data bus. Data is sent in the form of eight-bit ASCII characters or graphics dot data. Commands to invoke special features such as compressed print or graphics are sent as ASCII escape sequences. These are the same escape sequences used by all HP printers. The feature set of the 2674A Printer is based on HP-PCL (Printer Control Language), which ensures compatibility across HP's printer product line.

Data from the HP 150 is buffered in a 2K-byte RAM on the printer control board. The microprocessor then determines which direction it should move the carriage to minimize the time required to print the next line, and accelerates the head carriage up to print speed. The head carriage and paper advance motors are 7.5-degree step motors, and both are directly controlled by the microprocessor.

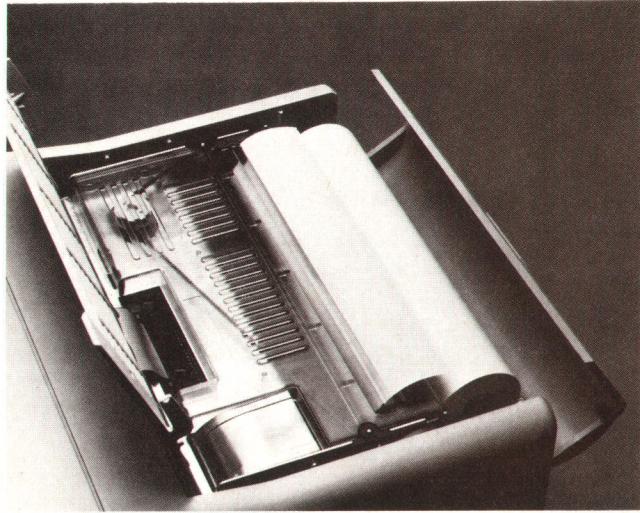
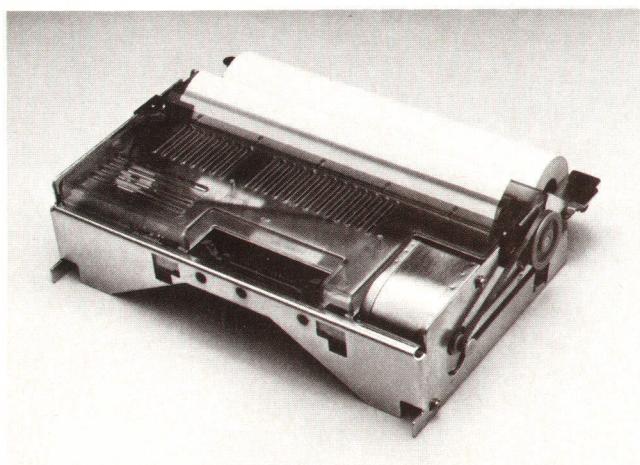
The microprocessor then gets the dot data to be printed. If it is in graphics mode, it looks up the dot data in the RAM buffer and converts it from the row format sent from the computer to the column format required by the print-

head. If it is printing character data, it looks up the dot pattern data in the ROM on the printer control board. When the printhead has reached the correct speed, the microprocessor starts sending the dot data to the printhead. To print at 120 characters per second, a new column of dots must be sent to the head every 463 microseconds. At the same time, the microprocessor continues advancing the head carriage motor. If a dot to be printed in a row was off during the previous column, it is left on for 430 microseconds. However, if a dot in a row was on in the previous column it will already be warm so it is only necessary to turn it on for 150 microseconds. This is known as second-dot compensation, and it is performed by the microprocessor while printing.

After a line has been printed, the head carriage is decelerated to a stop and the paper advance motor is activated. The microprocessor uses a reflective optosensor to check for holes in the paper that indicate top-of-form in perforated paper or the end of the paper. If a top-of-form hole is found, the paper is advanced to the top of the next page automatically after the last line on a page is printed. If a paper-out condition is detected, the paper remaining in the printer is ejected and the microprocessor waits for more paper to be inserted. When more paper is detected, the paper advance motor starts rotating so that the paper is caught by the platen and automatically loaded. Normal operation is then resumed.

Close contact was maintained with the HP 150 firmware development group to ensure that the printer interface would produce a "what you see is what you get" result for screen dumps. The HP 2674A is able to print any character set available on the HP 150, and it can print a dump of the graphics display in about 40 seconds.

The thermal printhead is delicate and easily damaged by abnormal conditions, so the control electronics has special circuitry to monitor the voltage on the printhead. If a head driver is left on too long, the drivers are disabled and the microprocessor is reset. This circuit also protects the printhead when the power is turned on and off and when the microprocessor is not working.



**Fig. 2.** The 2674A Printer is a wedge-shaped device that straddles the yoke of the HP 150's CRT.

#### Acknowledgments

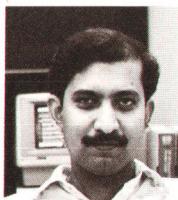
A tip of the hat to Elwood Barlow and Jim Traut of the HP Portable Computer Division for their cooperation and patience while we integrated their printhead into our product. The hardware portion of the 2674A was designed by John Widder and Robin Murray. Mechanical engineers Bill Huseby and Omar Khalifa designed the chassis, carriage, and mechanism. Industrial design was by resident wizard Bill Dalebout of the Vancouver Division, working closely with Lee Winick of the Personal Office Computer Division. The firmware that makes the printer operate was designed and implemented by Brooke Winter with early help from Claude Nichols. They maintained close contact with Zvika Bronstein of the HP 150 firmware group. Randy Goodner suggested the self-test and the diagnostic LEDs. Project manager Wally Thrash, who took over for Brodie Kent and also contributed to the electronic design, gets credit for making this project a success in a comparatively short time.

# Authors

August 1984

## 3 — Touchscreen Personal Computer

### Srinivas Sukumar



Srinivas Sukumar received his BE degree in electrical engineering from the University of Bombay (Victoria Jubilee Technical Institute) in 1973 and his MS degree in electrical engineering from Washington State University in 1975. After two years of software design for medical information systems, he joined HP in 1977, contributed to the firmware design of the 2626A Display Station, and served as a project manager for the 2622A and 2623A Terminals and as section manager for the HP 150 Computer. Born in Bombay, Srinivas is married and lives in Saratoga, California. He enjoys racquetball and tennis.

## 6 — Operating System and Firmware

### Laurie E. Pollero Wood



Laurie Wood received her BSCS degree from California State Polytechnic University in 1979 and joined HP the same year. She developed the firmware for the HP 125 Computer and contributed to the development of the operating system and utilities for the HP 150 Computer. She is now project manager for the HP 150 operating system and firmware. A native of Escondido, California, Laurie lives in Sunnyvale, California with her husband, an HP engineer. In her spare time she plays piano and enjoys a large variety of outdoor activities.

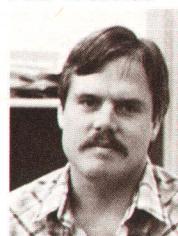
### Charles H. Whelan



Software engineer Chuck Whelan came to HP in 1973 with nine years' experience in the design of software for real-time control systems, distributed systems, and computer operating systems. He's developed software for three HP Divisions, most recently the operating system and firmware for the HP 150 Computer from HP's Personal Office Computer Division. He's the author of *Winning Deal*, a set of card games currently available for the HP 150. Chuck received his BA degree in mathematics from Oregon State University in 1964. He's married, has four children, and lives in San Jose, California.

## 11 — HP 150 Touchscreen

### Peter R. Stratton



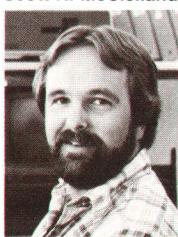
Peter Stratton joined HP in 1980 as a development engineer for terminals and workstations. For the HP 150 Computer, he was responsible for designing and implementing high-speed video, touchscreen, and console firmware. A patent resulted from his touchscreen work. Coauthor of three papers on eclipsing binary star systems, Peter is married, has one child, lives in Rocklin, California, and enjoys hiking, skiing, racquetball, and playing classical guitar. His home town is Eugene, Oregon and he received his BS degree in computer science in 1980 from Oregon State University.

### Thomas E. Kilbourn



Tom Kilbourn was born in Centralia, Missouri and attended the University of Missouri, graduating in 1979 with a BSEE degree. With HP since 1979, he's developed software design tools, done hardware design for the HP 150 touchscreen, and served as a production engineer for 264X Terminals. Currently attending Stanford University for his MSEE degree, Tom is married, has a son, lives in Sunnyvale, California, and enjoys motorcycles, reading, his family, and outdoor activities.

### Scott R. McClelland



Scott McClelland wrote the touchscreen scanning code for the HP 150 Computer. Three patent applications resulted from that work—Scott is co-inventor on two and sole inventor on one of those applications. While a student at Santa Clara University, he wrote graphics software for aircraft and space shuttle simulations for NASA. He joined HP after receiving his BSEE degree in 1981. Scott was born in Redwood City, California and now lives in Loomis, California. He likes to ski and race motorcycles.

## 15 — Applications Software

### Peter S. Showman



Pete Showman was project manager for the Personal Applications Manager for the HP 150 Computer. An HP development engineer since 1967, he has done analog and digital circuit design, developed microwave measurement software, and managed the development of several graphics hardware systems. He's a member of the IEEE and the ACM, and has served on ANSI's X3H3 committee

on graphics devices and metafiles since its formation in 1979. Pete is a 1965 BEE graduate of Cornell University and received MS and Engineer degrees in electrical engineering from the Massachusetts Institute of Technology in 1967. Born in Madison, Wisconsin, he is married, has two sons, and lives in Cupertino, California. He coaches youth soccer teams, heads school fund-raising projects, plays Irish and square dance music on the piano and violin, and enjoys "recreational" programming, skiing, and working on his TR4 sports car.

### Karl W. Pettis



Karl Pettis received a BS degree in mathematics in 1975 and an MS in computer science in 1977, both from Michigan State University, and an MS in computer science in 1978 from Yale University. He's also done further studies in computer science at the University of Arizona. With HP since 1981, he's worked on an HP 3000 to HP 150 Pascal crosscompiler and developed the MemoMaker software package for the HP 150 Computer and other business software. He's a member of the ACM and author of an IEEE Transactions paper on dimensionality estimation. Karl was born in Gainesville, Florida and is now a resident of Santa Clara, California. His interests include games, square dancing, comic books, and Gilbert and Sullivan productions.

### Karlie J. Arkin



Karlie Arkin received her BS degree in computer science from the University of Iowa in 1980 and worked there as a custom programmer and consultant before joining HP in 1981 as a programmer/analyst. She developed the Personal Card File package for the HP 150 Computer, generating a patent application for part of that work, and is now a development engineer with HP's Personal Software Division. On her own time, Karlie is involved in the Big Brother/Big Sister program and enjoys boardsailing, scuba diving, and photography. A native of Bettendorf, Iowa, she now lives in Santa Clara, California.

### John Price



John Price received his BA degree in psychology from Stanford University in 1966 and his MBA degree, also from Stanford, in 1973. With HP since 1973, he has worked on enhancements to financial software products for the HP 250 Computer, on customization and installation of modules of Materials Management/3000, and on the design and implementation of Series 100/Graphics. He was born in Point Pleasant,

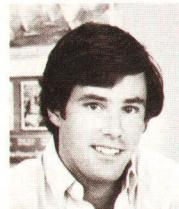
ant, West Virginia, now lives in Sunnyvale, California, and enjoys music, skiing, and reading. He's married and has two sons.

#### **W. Bruce Culbertson**



Bruce Culbertson received his BS degree in mathematics from the University of California at Davis in 1973 and his MS in mathematics from the University of California at San Diego in 1976. After teaching high school mathematics for five years, he returned to school, received his MS degree in computer and information science from Dartmouth College in 1983, and joined HP as a development engineer. He developed the Financial Calculator software package for the HP 150 Computer. A native of Oakland, California, he lives in Palo Alto, California and enjoys cross country skiing, running, hiking, and classical and ethnic music.

#### **Robert D. Shurtleff, Jr.**



Rob Shurtleff received his BA degree in computer science from the University of California at Berkeley in 1977. He joined HP in 1979 with two years' experience in software engineering and served as project leader for development of the Visi-Calc® 150 software for the HP 150 Computer. Now a project manager with HP's Personal Software Division, he also teaches data structures and Pascal at a local community college. Rob is a skier and a sailboard racer. He lives in Palo Alto, California.

#### **Jeffrey A. Spoelstra**



Jeff Spoelstra has been with HP since 1982, developing software for the HP 125 and HP 120 Computers and the Personal Card File package for the HP 150 Computer. A native of Des Moines, Iowa, he graduated from Iowa State University in 1982 with a BS degree in computer science. He is married, lives in San Jose, California, and enjoys astronomy and racquetball.

#### **25 — HP 150 Hardware Design**

##### **George Szeman**



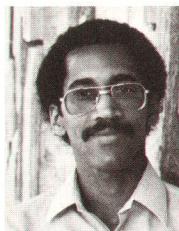
George Szeman helped design the video subsystem hardware for the HP 150 Computer. He's a 1981 BSEE graduate of the Stevens Institute of Technology and has been with HP since 1981. He's currently doing VLSI chip design. A resident of Sunnyvale, California, George is married and enjoys tennis, hiking, skiing, and working on his cars.

##### **Susan E. Carrie**



Susan Carrie joined HP in 1981 after receiving her BS degree in computer science from Michigan State University. She designed digital hardware and self-test firmware for the HP 150 Computer. A native of Detroit, she now lives in Santa Clara, California. Her interests include drawing, running, and softball.

##### **John E. Watkins**



John Watkins joined HP in 1978 after receiving his BSEE and MSEE degrees from Stanford University. He contributed to the development of the HP 300 and HP 125 Computers and helped develop the processor and memory subsystems of the HP 150 Computer. A native of Sacramento, California, he is married, has one child, and lives in Sunnyvale, California. He's active in his church and his hobbies are personal computers and electronics.

##### **Patricia A. Brown**



Trish Brown was born in St. Louis, Missouri and attended the University of Missouri at Columbia, graduating in 1979 with a BSEE degree. An HP employee since 1979, she's been a manufacturing engineer for the 262X Terminal family and a development engineer for the HP 150 graphics subsystem and memory option boards. She's a member of the IEEE and has served as an instructor in HP's Reg-

ional Occupation Program for high-school students. Her leisure activities include running, swimming, skiing, reading, and working on her house in Loomis, California, near HP's Roseville Terminals Division.

#### **30 — User Installable Printer**

##### **Joseph D. Barbera**



Joe Barbera received his BSME degree from the University of Portland in 1980 and joined HP the same year. He was responsible for mechanical design of the 2674A Printer. A native of Brooklyn, New York, he was a U.S. Army paratrooper for two years, then worked as a carpenter before going for his mechanical engineering degree. Now living in Portland, Oregon, Joe is a member of the Muscular Dystrophy Association and enjoys "any sport that risks loss of life and limb," including mountain climbing, white water canoeing, sky diving, and scuba diving.

#### **34 — Standard Keyboard**

##### **Michael R. Perkins**



Mike Perkins received a pair of BS degrees in 1981 from the University of California at San Diego, one in applied mechanics and one in bioengineering. While a student, he worked in physical oceanography at the Scripps Institute for two years. He joined HP in 1981 as a development engineer, and has contributed to the HP 2700 Color Graphics Workstation project and helped develop the 46010 Keyboard. Mike lives in Santa Clara, California and enjoys volleyball, scuba diving, backpacking, and cooking.

##### **Lorenzo Dunn**



Lorenzo Dunn hails from Cayucos, California, and has BS and MS degrees in product design from Stanford University. He joined HP in 1980, working on telephone test instruments. Since joining the Personal Office Computer Division, he has worked on keyboards, computers, and peripherals. He is single, lives in Palo Alto, California and enjoys inventing, bicycling, photography, writing, and seeing the countryside from a car with no top.

# A Standard Keyboard Family for HP Computer Products

*It's designed to meet ergonomic requirements, satisfy user preferences, be easily customized, and be produced in high volume at low cost.*

by Lorenzo Dunn and Michael R. Perkins

**T**HE HP 150 TOUCHSCREEN Personal Computer is the first HP product to use the new HP 46010 family of keyboards (Fig. 1). These keyboards will be used for a wide variety of Hewlett-Packard products, providing a uniform human interface for customers who use HP personal computers, terminals, and technical computers made by HP's Personal Office Computer Division, Roseville Terminal Division, Grenoble Division, and Fort Collins Systems Division. Variations on the keyboard layout will also be used for HP portable computers made by the Portable Computer Division.

Before the HP 150 project, it became clear that an HP standard keyboard would have to be a departure from previous HP keyboards. In the past, each HP keyboard was designed with the individual product in mind, leading to a proliferation of incompatible keyboards. Because of this fragmentation, volumes were low and costs high. Worse, users had to learn a new keyboard layout for each product they used.

As HP computer products moved from terminals, which were often dedicated to a single function such as word processing, to computers, which can change function with each new piece of application software, a general-purpose keyboard layout became a necessity. A group of representatives from many HP divisions was formed to arrive at a common keyboard layout.

## Design Considerations

Aside from the need to arrive at a standard keyboard layout, there were a number of other factors influencing the new keyboard design.

First was customer demand for keyboards with sculptured keycaps and a low profile for fast and more comfortable typing, a typewriter-like layout, and tactile feedback (the click you can feel when typing on a good keyboard).

Next were regulatory edicts that made ergonomic keyboard design a necessity as well as a competitive advantage. For example, regulations in Germany govern the height of the keyboard above the table, the contrast range of the case and keycap colors, and the detachment of the keyboard from the display unit.

The dropping cost of computer products meant that accessories like the new keyboard would have to be built with low cost as well as ergonomics in mind. The keyboard would have to be easy to build in high volumes.

Finally, the demands of selling computer products in an international market meant that the keyboard would have to be localized (key legends in the native language) for at least 17 different nations.



**Fig. 1.** The HP 46010 Keyboard is produced in seventeen local language versions for the HP 150 Personal Computer. Related keys are grouped together. Color is used to separate blocks (see Fig. 3).

## Low Profile

As mentioned earlier, good ergonomics was one of the primary goals in the design of the HP 46010 Keyboard. The height of the keyboard above the desk surface is one of the most visible ergonomic specifications. German regulations and good design practice require that the keyboard be low-profile, that is, that the home-row (**ASDF...**) keycaps be at a height of 30 mm above the desk. There are two reasons for this: first, to keep the overall height of the keyboard above the floor as close to the original desk height as possible, and second, to allow typists to reset their hands on the desktop while typing. Both of these features have been shown to improve keying performance significantly (increase speed and reduce errors).

But there are trade-offs in making a low-profile keyboard. It turns out that people are most comfortable typing on a keyboard that is angled up at about 12 degrees, and this increases the height of the home row. Another essential feature is a full-travel, tactile feedback switch. To meet the low-profile height requirement and keep the tactile feedback switch, the keyboard is designed to tilt so that it can be at its lowest height at a 3° angle to the desk and can be tilted up to a more comfortable 11.5° angle if the user desires.

The tilt leg (Fig. 2) is a single ABS plastic piece that snaps into the bottom of the keyboard. It clicks into a detent in the up or down position, and can be operated with one hand, from either end.

## Tactile Feedback

A full-travel, tactile feedback switch is the heart of a quality keyboard. A switch is considered to be full-travel when it has a stroke of 3.8 mm from the start of its travel to the end. For tactile feedback, an essential element for high-speed typing, the switch mechanism must tell the user's finger when the switch has made contact. Usually the contact point is well before the bottom of the keystroke and high-speed typists will not push the key all the way to the end of its stroke. HP keyboard designers worked closely with a carefully selected keyswitch vendor to develop a keyswitch that meets all of the tactile feedback requirements.

The HP 46010 Keyboard's keycaps are another example of ergonomic attention to details. They are step-sculptured keys. The step refers to the different heights of the different rows, while the sculpture refers to the curved surface of the whole keycap array. High-speed motion photography done in the early stages of the project and preferences expressed by customers showed the desirability of step-sculpture keyboards. The numeric pad keys are stepped, but not sculptured, because this proved to be optimum for numeric data entry.

The top of each cap is dished in a toroidal section to aid in locating the typist's finger. The **F**, **J**, and **5** keys, used to locate the user's hands on the keyboard, have a deeper dish than the other keys. The texture of the key tops was carefully chosen. It had to be rough enough to eliminate reflections, but not so rough that it would trap dirt or feel abrasive to the user's fingers.

## Layout and Graphics

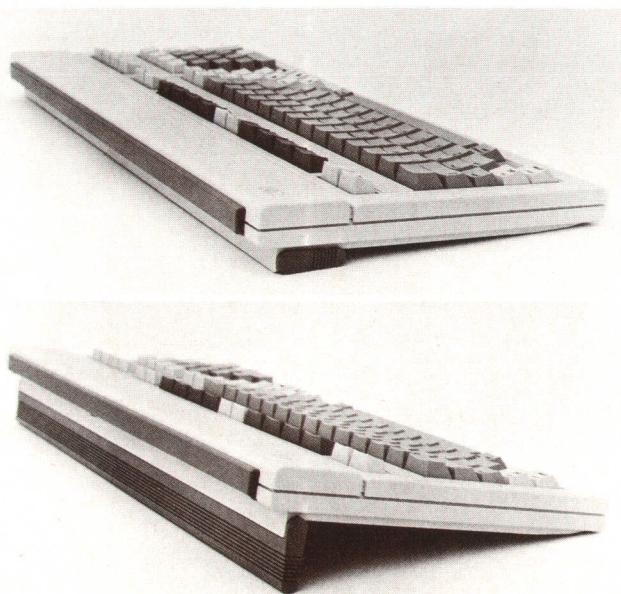
The key layout is designed to group related keys together. For example, the cursor related keys such as **Select** and **Home** are located near the cursor control keys, which are arranged in a comfortable "inverted T" layout. Above these are the editing keys. Color coding is used to separate blocks of keys, and keys that can have severe effects (like **Break** and **Stop**) are placed as far away from the main array as possible. Overall, the layout provides a high degree of functionality in a small amount of space.

The graphic design of the legends provides a clean, modern appearance along with superior readability. The use of blue legends to code the nonobvious shifted functions reduces the visual complexity of the keyboard.

Another design consideration is the need to customize the keyboard for specific applications. For example, certain software packages and hardware options redefine keys on the keyboard. International character sets, math sets, and line drawing sets are often used by redefining the keyboard. On other keyboards this has been handled by stick-on labels and other methods.

The HP 46010 Keyboard's numeric keys are designed so that an overlay can be placed over them, leaving room for a secondary legend below each key. Another overlay is available for relabeling the function keys. To solve the problem of getting to alternate character sets that are mapped onto the main array, a formed metal reference plate fits onto the top of the keyboard. It is the same width as the keyboard's main array, and it has a compressed graphic representation of the keyboard (Fig. 3).

Since the new corporate keyboard is to be used with a wide variety of products, it is designed to be visually compatible with HP's overall computer products styling program, retaining some of the visual motifs of this program, but applying them in a way appropriate to a keyboard.



**Fig. 2.** The tilt leg allows the user to choose the best combination of height and angle.



**Fig. 3.** Overlays for the numeric and function keys and a reference plate for the main array can be used to relabel keys that are redefined by specific applications.

### Product Design by CAD

The final product design of the keyboard was completed in a relatively short time, partly because of the use of HP DRAFT, a computer-aided design system. The Division that completed the layout of the keyswitch array sent their design to the Division that had final design responsibility in the form of a flexible disc. The final product design then became an interactive process of "building" the keyboard around the array on the computer screen. This allowed partitioning of part design responsibility with minimum errors and maximum speed.

The keyboard is designed to be built in high volume. Excluding the keycaps, there are only five major parts in the assembly: the top and bottom case parts, the kickstand tilt detail, the cable cap (all injection-molded ABS), and the keyswitch array.

The blank polyester keycaps are assembled onto the array and then printed, all at once, by a proprietary process. This

process enables economical production of 17 localized versions with one set of tools and considerably fewer different keycaps. It also reduces the misload rate significantly.

### Acknowledgments

This project would not have been possible without the help of countless individuals representing several divisions within HP. We would especially like to thank Tom Neal for his contributions to the industrial design. Mark Della Bona, Rick Nelson, Zong Luo, Dan Chung, Stu Center, and Dick Bergquam contributed to the product development, and Brian Leuenberger, Peter Fisher, and John Zielinski to the production processes. Lisa Quatman, Sandy Reynoso, Jack Wren, and Roger Slocum and their model shop teams devoted many long hours to assure the success of this project. Thanks also to the printed circuit department and to Larry Brown, the keyboard program manager.

Address Correction Requested  
Hewlett-Packard Company, 3000 Hanover  
Street, Palo Alto, California 94304

### HEWLETT-PACKARD JOURNAL

AUGUST 1984 Volume 35 • Number 8

#### Technical Information from the Laboratories of

Hewlett-Packard Company

Hewlett-Packard Company, 3000 Hanover Street

Palo Alto, California 94304 U.S.A.

Hewlett-Packard Central Mailing Department

Van Heuven Goedhartlaan 121

1181 KK Amstelveen, The Netherlands

Yokogawa-Hewlett-Packard Ltd., Suginami-Ku Tokyo 168 Japan

Hewlett-Packard (Canada) Ltd.

6877 Goreway Drive, Mississauga, Ontario L4V 1M8 Canada

Bulk Rate  
U.S. Postage  
Paid  
Hewlett-Packard  
Company

### CHANGE OF ADDRESS:

To change your address or delete your name from our mailing list please send us your old address label. Send changes to Hewlett-Packard Journal, 3000 Hanover Street, Palo Alto, California 94304 U.S.A. Allow 60 days.

# HP Archive

This vintage Hewlett-Packard document was  
preserved and distributed by

**[www.hparchive.com](http://www.hparchive.com)**

Please visit us on the web!

The HP Archive thanks George Pontis  
for his contribution of this material.

On-line curator: John Miles, KE5FX  
[jmiles@pop.net](mailto:jmiles@pop.net)

