Contents:

In this Issue:

With this issue we inaugurate a new and, we hope, more informative contents page. In this space each month we'll try to put the entire issue into perspective and outline the significance of the products and subjects discussed in the articles. Regular readers will also notice changes throughout the issue. Inaugurated last month, the changes involve type sizes and styles and the treatment of illustrations and inset articles.

Our cover subject this month is the HP 250 Small-Business Computer. An entry level computer for small companies and departments of larger companies getting into computers for the first time, it's designed to be easy to operate, to fit into the office environment, to adapt to the ways individual operators like to work, and especially, not to intimidate the first-time operator. But despite its friendliness and low price, it's more powerful than anything else in its price range, offering full data base management and facilities for report and forms generation.

Like any computer, the HP 250 needs to be programmed to solve business problems like inventory control and accounts receivable. HP supplies applications program packages that perform many of these functions. But because every business is different, these probably aren't the complete answer for every business. The more sophisticated user with in-house programmers can use the HP-supplied software packages to develop tailor-made programs. The first-time user with no programming capability will probably acquire the HP 250 through another manufacturer or software company who will provide ongoing software support. In fact, this is how most HP 250s have been sold so far.

Also in this issue (p. 29) is an article about a low-cost system using off-the-shelf HP products to measure how certain semiconductor-device parameters that are supposed to be constant actually change under transient conditions. Research on these effects has been done in the past using expensive special equipment. The low-cost HP system, developed at HP's central research laboratories, is being used in conjunction with previously published research data to develop various processes for making semiconductors. While it's not being offered for sale as a system, the authors feel that the system is so useful that many people involved in semiconductor process development might want to put one together for their own use.

-R. P. Dolan

A Human-Engineered Small-Business Computer

This entry level, low-cost system offers the first-time user a self-contained computing facility with full data base management capability.

by A. Peter Hamilton

A SMALL COMPUTER with full data base management capability, the HP 250 Business Computer is designed as a completely self-contained computing facility for small companies and divisions of large firms. The system offers the first-time user an approachable computer that is friendly, simple, and interactive, as well as powerful, reliable, and expandable.

Because the primary business applications are data intensive in contrast to the computational intensity of scientific applications, the HP 250's data base management software system is considered a significant contribution in a low-cost, entry-level system. Data base access is further enhanced through better data entry and reporting capabilities. These facilities, coupled with a powerful inquiry facility, provide easy access to timely information for better control of the business.

The HP 250 (Fig. 1) is designed to provide big-computer-system capabilities while minimizing training time and maximizing access to the computer's power. The system addresses the needs of the user in numerous ways. Perhaps most visible are the human factors considerations, namely the desk-style work station, the easy startup by the turn of a key, the typewriter-like keyboard, and the adjustable CRT that tilts and slides to fit the variations in individual needs. The other important but not so visible system capabilities are the 1.2M-byte flexible discs, and the expandability of the system to include a 12.8M-byte Winchester-type fixed disc, an HP 7906 20M-byte cartridge disc, a 180-character-per-second printer, and a multi-user terminal configuration. User memory is expandable from 32K bytes to 64K bytes. The HP 250 also provides asynchronous data communications capability to other HP computers.

HP 250 applications software available at this time is focused on the small first-time user, who will most likely acquire the HP 250 from an original equipment manufacturer (OEM) or a software firm. However, larger companies and experienced end users may acquire the system directly and supply their own applications software in addition to
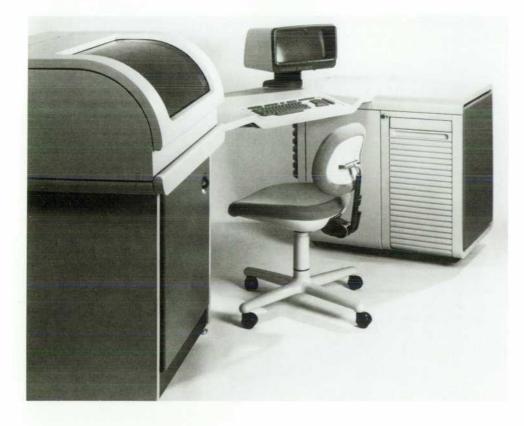

Fig. 1. *The HP 250 is a low-cost entry level computer system for small businesses. It is shown here with the optional printer noise-reduction enclosure.*

Human-Engineering the Small-Business Computer

by Barry Mathis

Traditionally, the user interface of computer products has been designed by and oriented towards the programmer. As a result, these devices have reflected the programmer's computer expertise, making computers approachable by only a small group of highly skilled initiates. For those outside this inner circle, computers have seemed forbidding, their keyboards complex, their languages obscure, their displays baffling. Unfriendly! What has been missing is human engineering for the unitiated operator.

Because it is a business computer, the HP 250 presented some interesting human factors challenges. First, its operator is likely to be a clerical person, not a computer professional or a programmer. Familiar only with a typewriter, such an operator may be intimidated by machine complexity. Confusion, frustration and fatigue result from poor operator-machine mating. Second, the tasks performed on a low-cost computer are primarily business accounting procedures. An order processing or billing clerk might spend an entire work day at the computer's console using references, documents, and procedures unique to the job and the operator's style. Third, this type of computer's home is the business office. Unlike the special-purpose computer room, this is a people environment demanding quiet operation, efficient use of space, and furniture-like aesthetics. All of these considerations influenced the HP 250 product design.

Design Approach

Initially, we talked with people about how they liked to work. People entering customer orders at a terminal told us that, along with the keyboard and the display, the source document is an important part of their work process. The need for an ample work surface, well placed and within arm's reach, was emphasized. Computer and terminal keyboards were described as complex, confusing, even intimidating. Many office machines are uncomfortable, lacking proper leg room and ample work area. They are too noisy and often unattractive. Office workers are apt to feel that they are the victims of their equipment, not its masters.

This information changed some of our basic assumptions about computer design. Table-top component systems seemed woefully inadequate. To evaluate design decisions, a series of full-sized models was developed. These crude cardboard constructions helped determine electronics enclosure volumes, cooling techniques, mechanical layouts, and service access. More important, they gave us valuable ergonomic data on heights, angles, work surface arrangements, and overall size. A number of these constructions led us to a configuration we were convinced was the optimum one.

Next we proceeded to a simulator test. This involved a plywood model, along with a terminal for its display and keyboard, and a software set to make it operational. A simulated order processing routine provided the basis for our test. A variety of people (large, small, male, female, experienced and not) tested the design. After arranging the movable components—reference catalogs, order forms, source documents—to his or her liking, each worked for several hours, gaining enough familiarity with the proposed system to be critical.

Problems arose. No two people liked the display in the same place. We had assumed this to be a fixed element. Glare often made the display difficult to read. Some wanted it directly in front of the keyboard instead of to the right where we had fixed it. Our proposed work surface was too constricted for comfort, and the terminal keyboard was felt to be confusing.

Applying these inputs, we began to explore how a CRT assembly might be made movable, a real technical problem. Printer movement was also studied. The keyboard layout was overhauled for increased simplicity, clarity, and "typewriterness." One-quarter scale modeling was employed to facilitate easy form variation. A variety of work surface shapes were considered to find a solution balancing ample area against floor space requirements.

Aesthetics also were considered, in the hope of creating an ergonomically correct design that would be welcomed into the office. Manufacturing process considerations began in earnest, and service concerns were evaluated and weighed. Finally, the parallel electronics effort was merged with the physical product design, a highly finished wood model was made for a final check of critical ergonomic and visual factors, and tooling and fabrication detailing began.

The Right Tool for the Job

The HP 250 has been human engineered to fit the operation and be very approachable. Some of its significant design features follow:

Work Station. Since operators arrange their work areas to suit the task and their personal preferences, a total work station concept has been developed. The placement of all the functional elements (e.g. work surfaces, keyboard, display, discs, and printer) is based on human engineering research. Everything is within arm's reach. Source documents, reference materials, and so on may be arranged to suit the individual. The L-shaped table top is contoured for comfort and easy cleaning. The unit fits handily into a corner and occupies very little floor space.

Movable Display. Perhaps the HP 250's most singular commitment to human engineering is its adjustable CRT display screen. It slides from side to side, swivels about a vertical axis, and tilts. This range of adjustment accommodates all preferences for screen position. For example, people wearing bifocals can tilt the screen just slightly downward to facilitate their viewing. Where there are lighting variations, slight angular adjustments can reduce distracting glare. Contrast can be varied to individual preference with an easy-to-reach knob.

Soft Keys. Eight user-defined function keys are located along the bottom edge of the display screen. A changeable function message displayed above each soft key describes its present function. This allows ease of learning and convenience of operation, and reduces the likelihood of keystroke errors.

Keyboard. Positioned at a proper height and angle, the HP 250's data entry keyboard is generously spaced yet simple and uncluttered. It closely resembles a typewriter keyboard plus a standard ten-key adding machine pad. Typing keys are contoured and the key shape helps the operator home in on the keys by feel. The keyboard is free of complex nomenclature and is available in several foreign languages.

Aesthetics. The HP 250 is designed to fit into the office. It has clean, contemporary styling. Accent panels, available in several colors, help coordinate it with the office decor. The neutral light gray is the optimum color for a working table top.

Quiet Operation. Acoustical pollution is an increasing problem in offices. The HP 250 does its part to keep this to a minimum.

Multiple-slow-speed-fan cooling and subsystem partitioning allow quiet yet efficient operation. An optional noise enclosure for the printers reduces printer sound output to a respectable 51 dBA.

Serviceability. The HP 250 provides for fast, easy service. A roll-out card cage allows access to most key components from the operator's seat. Ease of service helps customers minimize down time.

Process Engineering. An array of manufacturing technologies and materials was used in this computer. Compression molding provides a tough, durable fiberglass reinforced table top that is mar-resistant and easily cleaned. The electronic enclosure is a welded steel fabrication. RIM—reaction injection molding—provides a printer noise enclosure that has superior sound absorbing properties very much like wood. Injection molding and vacuum forming are used extensively. Two-color injection molding forms the keys.

Additional features include a drawer for supplies, quick reference instruction cards, and a tambour door to cover the flexible disc drives. System turn-on is accomplished with a single key—simple and secure. A flexible cable manager allows freedom in positioning the printer. Businesses that use numerous forms for billing, payroll, and so on are accommodated by a rolling printer stand designed for paper handling ease. The system itself is on rollers for easy installation and movement.

Barry Mathis

Barry Mathis received his BS degree in industrial design from Auburn University. An HP employee since 1967, Barry was responsible for the industrial design for eight electronic instruments, the 3800A Distance Meter, the 9815A and 9805A Desktop Computers, and the 9896A and HP 250 Business Computers. He is also named inventor on several design patents related to the 3800A, the 9805A, and the 9815A. Barry is married, has two daughters, and lives in Loveland, Colorado. His diverse interests include art, films, theater, "culinary adventures at home and about," travel, music, photography, prize fights, and "strange mushrooms."

HP's. Applications for the HP 250 in larger companies tend to be departmental in scope and span a broad range of functional tasks. The HP 250 supports this type of need with powerful application tools in the IMAGE/250 data base manager, FORMS/250 data entry facility, REPORT WRITER/250 report production facility and QUERY/250 inquiry capability. These tools simplify the development of specific applications for departments of larger companies.

Applications software for the HP 250 supports the order management and material control requirements of companies with $1 million to $20 million in sales. The applications have been designed to provide significant flexibility in tailoring the software to the user's specific needs at installation time. Parameters and capacities are adjusted without custom programming. This tailoring facility allows individual business options, such as customer and product discount categories, single or multiple warehouse locations, open-item or balance-forward invoicing, and several other choices. These capabilities, commonly referred to in the industry as "parameterized applications," provide tailored applications without custom programming in many cases.

The HP 250 is an entry level computer system in the HP continuum of computer products, but it is a powerful data base computer system with the performance and features expected of much larger computers. For a small first-time business computer user or a larger company with departmental computing requirements, the HP 250 provides capabilities at a cost not thought possible, and without an EDP staff to support it.

A. Peter Hamilton

An HP employee since 1965, Pete Hamilton was responsible for the application tools and applications of the HP 250 Business Computer. He was product marketing manager for the 9885M/S Flexible Disc Drive, the 9871A Printer, and the 9896A Business System, and is currently marketing manager for the HP 250. Born in Portsmouth, New Hampshire, Pete attended Wentworth Institute of Technology, the University of Idaho, and Valley State College in Van Nuys, California. Prior to joining HP, he did government aerospace electrical power feasibility studies for a nuclear engineering firm. A resident of Loveland, Colorado, Pete is married, has two sons (ages 13 and 16), and enjoys skiing and traveling in his spare time.

ABRIDGED SPECIFICATIONS
HP 250 Small-Business Computer

SYSTEM CONSOLE
Microprocessor CPU
32K or 64K-byte user memory
User-definable softkeys
Adjustable display screen
Office-typewriter-like keyboard (set at proper typing height)
System self-test (at turn-on)
Single on/off (key) switch
PRINTERS
HP 2631A: 180 cps, dot matrix
HP 9871A: 30 cps, impact
MASS STORAGE
FLEXIBLE DISC DRIVE
SINGLE: 1.2M bytes
DUAL: 2.4M bytes
TRIPLE: 3.6M bytes
FIXED/REMOVABLE CARTRIDGE DISC: 19.6M bytes

SYSTEM SOFTWARE
HP Business BASIC
IMAGE/250 Data Base Management
FORMS/250
REPORT WRITER/250
QUERY/250
STANDARD SYSTEM
HP 250 system console (product number 45251A: microprocessor, display screen (with softkeys), 128K-byte system memory, 32K-byte user memory, peripheral interface channel, 1.2M-byte flexible disc drive and BASIC interpreter system software.
A second flexible disc drive (Opt. 010).
180 cps dot matrix printer (Product Number 2631A, Opt. 250).
ASYNCHRONOUS SERIAL INTERFACE (Product Number 45120A, System Option 120)
TERMINAL/PRINTER INTERFACE
Five ports at 110 to 9600 baud rate. On-interrupt programming concept via system software module. HP 264X, HP 2631 and HP 2635 direct/modem connection.
20 mA current loop connection.

HP 250/HP 3000 LINK
Log-on terminal capability. Data file and program transfer. Initiate HP 3000 jobs from HP 250.
RS232C PERIPHERAL INTERFACE
Connect TTY compatible device. Parity selection, null specification.
PERFORMANCE SPECIFICATIONS
INSTRUCTIONS PER SECOND (TYPICAL): 380 000
MEMORY CYCLE TIME: 833 nanoseconds
TRANSFER RATE (MAXIMUM) - GENERAL INTERFACE CHANNEL: 1 megabyte/s
TRANSFER RATE - FLEXIBLE DISC: 62.5K bytes/second
AVERAGE ACCESS TIME - FLEXIBLE DISC: 172 milliseconds
PRICE IN U.S.A.: HP 250 Standard System, $24,500.
MANUFACTURING DIVISION: FORT COLLINS DIVISION
3400 East Harmony Road
Fort Collins, Colorado 80521 U.S.A.

Cost-Effective Electronics for the Small-Business Computer

by Gerald L. Meyer and V. DeLloy Forbes

DESIGN CRITERIA for the HP 250 took into account the interests and needs of buyers and users of small business computers. The result was that major emphasis was placed on cost-effective performance, reliability, and safety.

Fig. 1 is a block diagram of the HP 250 electronics. The heart of the system is a 16-bit NMOS processor manufactured by the HP NMOS facility at Loveland, Colorado.[1,2] This is a hybrid chip set which is also used in other HP products, such as the desktop System 45.[3] It was chosen because of its proven reliability and effective processing power.

Because the 16-bit address field of the processor can directly access only 128K bytes of memory, while the actual memory requirement could be several times that, a method of extending the memory address space had to be used. The technique used is an expansion of the method used on the System 45.[3] Based upon dedicated processor register contents and the present state of instruction cycles, the memory block switch shown in Fig. 1 generates a four-bit digital word that selects one of 16 possible 64K-byte blocks of system memory for a memory cycle.

The processor board contains two 64K-bit ROMs, also fabricated with HP's NMOS process. These ROMs contain the code for turn-on self-test and system diagnostics and the code for loading the operating system from system mass memory (flexible or fixed disc) into system read/write memory. Loading occurs automatically at system turn-on if the self test is successfully passed.

The system read/write memory is composed of a number of 64K-byte memory blocks, depending on the requirements of the user. The maximum number is limited by the number of available slots for memory boards in the card cage. The memory ICs are commercially available 16K-bit dynamic NMOS RAMs that have been burned-in and screened to HP specifications. For error detection, the memory includes a parity bit with each byte.

The processor uses the input/output bus to interface with the keyboard logic. In addition to the typewriter-like keyboard on the console, there are eight soft keys on the CRT bezel under the display. The functions of these keys are defined by the user or by programs and are displayed on the CRT just above the keys.

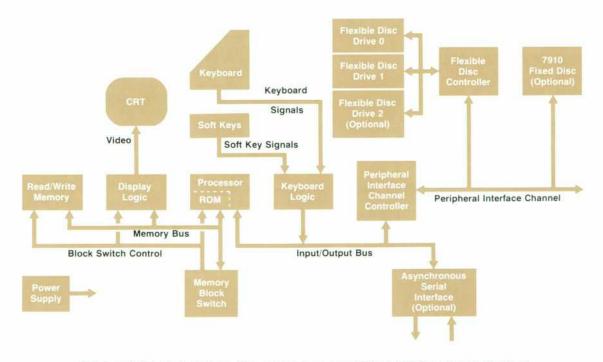A 1920-word segment of memory space is allocated to a

Fig. 1. *HP 250 block diagram. The processor is an HP 16-bit NMOS processor of proven reliability and processing power. The memory block switch is part of a memory address extension scheme for managing large memories.*

special static RAM block located on the display logic board and dedicated to refreshing the CRT display. The raster scan CRT displays a standard 80-column-by-24-line organization for a total of 1920 characters. There is a one-to-one correspondence between a location on the CRT display and a 14-bit word in the refresh RAM. Nine of these bits select one of 512 possible characters. The other five bits describe enhancements or modifications to the selected character. The possible enhancements are cursor (a blinking underline that identifies where the next character will be typed), inverse video (black on white), blinking, underline, or half-bright. The characters available include the full 128-character ASCII set (upper-case Roman, lower-case Roman, numerals, and control characters), 32 special HP 250 CRT editing characters, and an 88-character line drawing set. An optional plug-in ROM expands the set to include the characters needed for German, French, British, Danish, Norwegian, Swedish, Finnish, Spanish, Italian, and Katakana (for Japan).

Display Logic

A block diagram of the display logic board is shown in Fig. 2. It was designed to require no processor attention for a static display. Besides the benefit of not slowing the system down when the display is not changing, this architecture permits the HP 250 to have an effective self-diagnostic capability, which will be described later in this article. Another advantage of this design is that the contents of the refresh memory can be modified rapidly without causing unpleasant pictures on the CRT.

The character fonts are defined by a 9×15 dot matrix. As a result, it takes 15 CRT scan lines to paint one row of characters. The dot rate is 23.69 MHz and therefore the character rate is $23.69 \div 9 = 2.63$ MHz, or 380 ns per character. The refresh memory is separated into two sections. One section contains the character information corresponding to the even locations on the CRT and the other section contains the character information corresponding to the odd locations on the CRT. When the display is being refreshed these two RAMs are accessed alternately at sequential addresses at a 380-ns rate. Each of these RAM blocks consists of four 4×1K-bit NMOS ICs. Separating the RAMs in this manner permits rapid update of the RAM contents. When a new
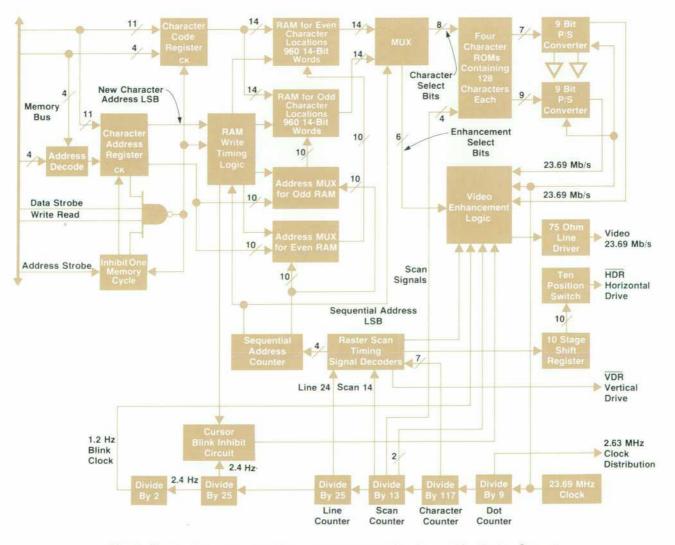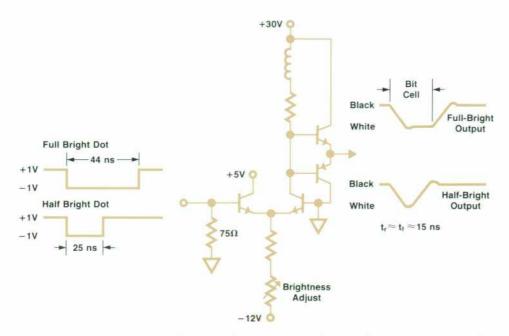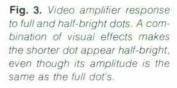

Fig. 2. *Display logic board requires no processor attention for a static display. Separate random-access display memories (RAMs), one for even character locations and one for odd, make it possible to update the display without interfering with display refresh.*

Fig. 3. Video amplifier response to full and half-bright dots. A combination of visual effects makes the shorter dot appear half-bright, even though its amplitude is the same as the full dot's.

character is to be written by the CRT, the CPU writes the new character and its address into the two registers located on the display logic board. The RAM write timing logic senses that a new CRT character is in the input registers and determines the correct time to write this word into the refresh RAM. It does this by comparing the least significant (odd or even) address bits in the input address register with the least significant bit of the sequential address scan counter. When it senses that they are the same, the write will be done on the following access, since at that time the opposite RAM will be accessed for refresh purposes. The effect is that the character is cleanly and rapidly changed without interfering with the refresh function. It takes at most three character times to change a character (about 1.2 μs), which is faster than the CPU will write back-to-back character changes. During horizontal and vertical retrace there is no holdoff of the write into RAM. The entire CRT can be rewritten in three milliseconds.

All enhancements are implemented with logic. This is true even of half-bright. Fig. 3 shows the input and output of the video amplifier with full and half-bright dot inputs. The half-bright dots are approximately half as wide as a dot cell, but of the same amplitude. Because of the combined effects of the rise time of the video amplifier, the resolution of the CRT tube, and the CRT-to-observer distance, the dot appears less bright rather than half as wide. This technique avoids the cost and problems associated with driving two six-foot video lines (the distance from the display logic board to the CRT) as would be required by a bilevel current video amplifier design.

Since the soft keys on the CRT bezel are very close to the CRT traces that call out their functions, it was considered desirable that the horizontal position of the CRT display be very stable. The block diagram, Fig. 4, illustrates how the flyback pulse time is stabilized with respect to the horizontal sync input from the display logic board. The flyback pulse circuit input-to-output delay will vary somewhat with the environment. The flyback pulse is limited and compared in phase with the horizontal sync pulse. This

phase comparator produces an output voltage proportional to phase, which then is compared with the reference voltage. The error is amplified and goes to a voltage-variable delay, which delays the input horizontal sync by just the right amount to place the flyback pulse at the proper time to maintain a stable display.

Flexible Discs and Other Peripherals

The peripheral bus controller serves the function of interfacing between the internal processor I/O bus and the external peripheral interface channel (PIC). The PIC has many functional similarities to the IEEE-488 bus (HP-IB) but is not strictly hardware or software compatible with it. The primary work of the peripheral bus controller is handled by a silicon-on-sapphire (SOS) LSI circuit built by HP's SOS IC facility in Cupertino, California.[4] Peripherals that communicate with the HP 250 system processor via the PIC are the external printers and magnetic disc memories (of which various options are available), and the flexible disc memory subsystem built into the HP 250 mainframe.
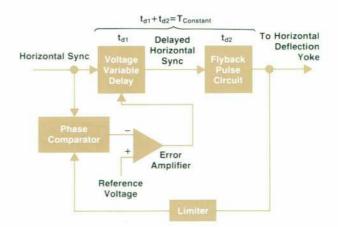

Fig. 4. Feedback loop stabilizes the horizontal sync pulse so the soft key labels on the CRT remain stable over the keys.

The HP 250 requires a mass memory subsystem capable of hosting the operating system and system utilities, a number of discrete relocating option modules (DROMs), a library of BASIC applications programs, the data base management system, a user's data base with enough capacity for a wide range of applications, and user programs, while at the same time providing data backup and operating system, data, and program portability. The various mass memory options available on the HP 250 satisfy these requirements very nicely.

For a minimum-cost system, an added requirement of the mass memory subsystem is low cost. A flexible disc ("floppy disc") memory was especially designed for the HP 250 to meet this requirement. The flexible disc drives are relatively inexpensive and small enough so that up to three drives can be mounted in the HP 250 mainframe. The media (flexible discs) are inexpensive and durable, and can be stored efficiently in a typical office environment. To provide the capacity needed by the HP 250, however, it was concluded that each drive should have a minimum capacity of one megabyte of on-line data storage, two to four times that of typical flexible disc drives.

The flexible disc memory subsystem of the HP 250 consists of one to three drives, a controller printed circuit board mounted in the HP 250 roll-out card cage, and cables that connect the controller board to the drives in a multidrop configuration (see Fig. 1). The controller communicates with the HP 250 system CPU (central processing unit) via the peripheral interface channel (PIC), as do the various

other external system peripherals.

The heart of the controller is the MCC or MC² (micro-CPU-chip), a 16-bit control-optimized CMOS/silicon-on-sapphire microprocessor.[4] The peripheral interface channel is serviced by another CMOS/SOS integrated circuit, the processor-to-HP-IB interface (PHI),[5] which communicates with the MC² processor via the processor bus (see Fig. 5). Most of the functions of the controller are carried out by the MC², executing ROM-based code. However, because of the relatively high data rate, some of the high-speed serial data functions, such as data serialization/deserialization, write data encoding and precompensation, read clock/data separation and decoding, and CRC code generation and error detection were implemented in hardware external to the processor.

The flexible disc drive has two read/write heads and is capable of writing and reading on both sides of the flexible disc. The heads are mounted in a taut-band linear actuator mechanism driven by a two-phase step motor that steps the heads to any of the 77 tracks. When a flexible disc is inserted in the drive and the door closed, a clamp mechanism clamps the flexible disc to the drive spindle, which rotates the disc within its protective jacket at a constant 360 r/min. An ac motor drives the spindle by a step-down belt-pulley arrangement. The drive also has detectors for sensing whether the drive door is closed, whether a flexible disc is write-protected, and whether the heads are at track 0, and for sensing the index holes in single- and double-sided discs. An electromagnetic solenoid is used for loading the
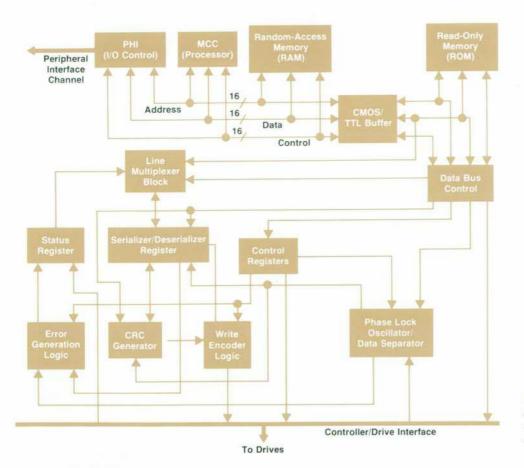

Fig. 5. Flexible disc controller uses two HP silicon-on-sapphire chips, the MCC processor and the PHI HP-IB interface chip.

	HP Format- Double Density	IBM Format- Single Density
• **Capacity**		
- Unformatted	1600 Kilobytes	400 Kilobytes
- Formatted	1182 Kilobytes	256 Kilobytes
- No. of Sides	2	1
- Tracks/Side	77	77
- Sectors/Track	30	26
- Bytes/Sector	256	128
• **Maximum Recording Density**	6560 Flux Reversals/Inch	6560 Flux Reversals/Inch
• **Rotational Speed**	360 r/min	360 r/min
• **Maximum Data Rate**	500 Kilobits/s (62.5 Kilobytes/s)	250 Kilobits/s (31.25 Kilobytes/s)
• **Access Time**		
- Track	3 ms	3 ms
- Average (Including Settling)	91 ms	91 ms
- Settling Time	15 ms	15 ms
- Head Load Time	35 ms	35 ms

Fig. 6. *HP 250 flexible disc drive specifications.*

Fig. 7. *Flexible disc drives divide each bit cell into a clock window and a data window. Lower curves show probability distributions for the locations of bit transitions within these windows. Errors occur if any transition falls outside its intended window.*

heads onto the disc, and another is used to lock the drive door to prevent removal of the disc when the heads are loaded. The door lock on each drive can also be controlled by the host system, which can issue door lock or unlock commands to a given drive via the controller. Fig. 6 presents some HP 250 flexible disc drive specifications.

Attached to each drive is a drive electronics board that contains the head step-motor drivers, solenoid and front-panel indicator drivers, drive- and head-select decoders, write and read amplifier circuits, and controller bus interface receivers and drivers.

Flexible Disc Capacity

One of the most difficult challenges in the design of the HP 250 flexible disc subsystem was achieving the required data storage capacity. To store more than one megabyte of formatted data on a single standard flexible disc required first that both sides of the media be recorded. Second, since present-day flexible disc and head technologies limit recording densities to a practical limit of about 8,000-9,000 flux changes per inch, an efficient double-density code yielding a data bit density equal to the recording density was necessary. Since an efficient double-density code, "modified modified frequency modulation" or MMFM, had previously been used in the HP 9885A Flexible Disc Memory, it was decided to use not only the same encoding scheme, but also the same sector and data formats, so that discs recorded on the 9885A would be readable in an HP 250 drive and vice versa. Using double-density recording on all 77 tracks of both sides of the disc, the total usable flexible disc data capacity is a respectable 1.18 megabytes.

In addition to the HP double-density recording format, the HP 250 flexible disc drives are capable of reading and writing the industry-standard IBM single-density format (single-sided only), which provides a usable data-capacity of 256 kilobytes per disc. Software drivers are being developed so the IBM format can be provided on the HP 250 as a means of transferring data to or from non-HP systems equipped with flexible disc memories capable of reading and writing the more common IBM single-density format.

In the MMFM recording format, a data one is represented by a transition of the polarity of magnetization on the surface of the disc while a zero is represented by the lack of a flux transition in any given bit location (bit cell) on the disc. In MMFM, the read clock is synchronized with the detected transitions of the data stream being read. Since the data stream might be completely or partially devoid of ones and therefore of data transitions, it becomes necessary to insert clock transitions in data strings containing two or more adjacent zeros. To differentiate between data transitions and clock transitions during a read, each bit cell is divided into a clock window and a data window (see Fig. 7a).

A phase-locked oscillator (PLO) is used to generate the read clock signal, which is synchronized with the read data stream. The PLO is designed to track slow variations in the frequency of the incoming data stream, such as those that might be caused by mechanical wow and flutter, ac line frequency and voltage fluctuations, or other low-frequency phenomena, but to reject rapid variations such as those

HP 250 Input/Output System

by Dennis L. Peery

The input/output system for the HP 250 is divided into local or primary system I/O and remote or secondary system I/O.

The local I/O system is a subset of the HP-IB.* It operates the system discs and printers in the local system in specific configurations and is not intended as a general HP-IB controller. The hardware consists of one printed circuit card. The software drivers are part of the system software and are loaded when the system is turned on.

Providing the power of a general-purpose computer in a turnkey computer system such as the HP 250 presents special problems to the designer of the I/O system. To keep the system small and uncomplicated, the scheduling algorithms that manage the I/O traffic over the system bus have to be straightforward and concise. Also, the system has to be smart enough to power-up and start running with a minimum of user assistance.

So an experienced operator wouldn't be required, some other means of describing the I/O configuration had to be invented to accomplish the operator's task. This was done by requiring that devices on the system bus be capable of telling the system what they are and where they are. This self-identify feature allows the system to poll devices on the system bus at power-up and thus gather all the information it needs about the I/O configuration. This feature lets the user add devices or change their bus addresses without having to know any special commands to get them to work.

The convenience of automatic configuration of the I/O system comes at a cost of some flexibility, because all devices on the system bus are required to have the identify feature. This flexibility is regained on the HP 250 by providing for nonstandard devices on the system bus. A nonstandard device is a device not supported by a system driver, or a device without the identify feature. These types can be supported by a special driver configured as a DROM (discrete relocating option module, an optional system software module).

After the system has powered-up, all standard devices have been identified, and all nonstandard devices have been linked with their DROM drivers, the I/O system traffic controller takes over. It is the traffic controller's job to schedule all requested operations on the system bus.

To optimize use of the system bus, I/O operations are scheduled according to bus availability. This means that I/O operations that do not conflict may be interleaved if the bus becomes available. For example, output to a printer may take place between the seek and read operations on a disc.

The I/O traffic controller also takes responsibility for guaranteeing that a user's I/O operation on a device is completed before allowing another user to have access to the device. Conflicting requests for the same device are handled by queueing the requests on a first-come-first-served basis. For simplicity, there is only one queue, and all tasks on the HP 250 operate at the same priority.

Remote I/O System

There is a need in a small business system to provide a means for remote I/O devices, such as multiple work stations for data input and data base inquiry, and printers for output. In addition, a departmental computer often needs a communications link to other computers for batch data transfers. The asynchronous serial interface (ASI) and supporting terminal input/output DROM (TIO), an optional hardware and software extension of

*The HP Interface Bus (HP-IB) is HP's implementation of IEEE Standard 488 and identical ANSI standard MC1.1.

the HP 250 I/O capability, provides these remote I/O functions.

Asynchronous Serial Interface

The asynchronous serial interface (ASI) provides an intelligent interface that controls the flow of data between the HP 250 and any one of the following types of devices: Remote/250 Console (HP 2649D), terminal, printer, or HP 3000 Computer. Besides handling the transfer of data, this interface monitors the data link to make sure that the channel is operational.

The asynchronous serial interface provides five ports. Each port is independently configurable. It can be set to run at any one of ten speeds between 110 and 9600 bits/second, to handle even, odd, or no parity, to process characters with one or two stop bits, and to provide the necessary RS-232C control signals to talk to remote devices connected either directly or through full-duplex modems. To allow transmissions over long distances (up to 1 km at 9600 bits/s) on direct connections, a 20-mA current loop option compatible with the HP 2645A Terminal comes with the standard ASI hardware.

An 8-bit microprocessor, along with 2K bytes of firmware resident in the RAM (random-access memory) on the ASI board, acts as a traffic manager for the data to and from each port. How the data is handled depends on the type of remote device the port is configured to support. A memory location for each port, initialized by the main system before starting the ASI microprocessor, sets the input/output protocols that the firmware must follow in handling the data. This byte contains information such as immediate input or deferred input, terminal or computer, ENQ-ACK or no ENQ-ACK, and echo or no echo.

In connections to auto-answer modems, the ASI firmware continually monitors the condition of the data channel. As long as the connection is made, data is allowed to flow out to the remote device. If, however, either of the RS-232C control lines indicating that the connection is good goes off, the ASI firmware automatically disconnects the phone and notifies the main system that a disconnection has occurred. As soon as the connection is reestablished, the ASI firmware informs the system that the connection is again usable.

The ASI hardware has two additional features that enhance its performance and flexibility. The first feature simplifies the transfer of data between the ASI hardware and the main system. A cycle-steal technique allows the main processor to directly access the RAM on the ASI board without informing the ASI microprocessor. This technique frees the ASI microprocessor to monitor the ports continually for input even while data transfers to and from the main system are taking place. This feature also eliminates the need for a complicated handshake between processors.

The second feature enhances the ASI's flexibility. There is no ROM (read-only memory) on the ASI board; the memory is all RAM (random-access memory), half of which can be memory-protected. All the ASI firmware, including self test, is downloaded from the main system before the ASI microprocessor is started. Once the ASI microprocessor is started, the RAM containing the ASI firmware is automatically protected by preventing writes to that portion of the memory. This feature makes it possible to modify the ASI firmware without costly ROM turnarounds and hardware retrofits.

Software Support for ASI

An optional extension to the HP 250 operating system, the TIO

(terminal input/output) DROM, provides a convenient interface between the BASIC-language applications programs and the asynchronous serial interface (ASI). Using the BASIC-language extensions implemented by TIO, applications programs can support most of the features of the HP 264X-Series Terminals. In addition, programs may communicate with the HP 3000 Computer System through that system's asynchronous terminal controller. An HP software product, a program called LK 3000, provides terminal emulation and file transfer utilities that temporarily turn the HP 250 into a very smart log-on terminal.

Remote I/O operations may execute concurrently with other remote I/O operations and with processing. This characteristic allows the creation of applications programs supporting multiple users of remote terminals. The remote I/O scheme appears to the applications program as a multilevel priority interrupt system. The interrupts, which are emulated by the operating system software, may occur at the completion of processing of each BASIC-language statement in the applications program. Thus, the applications programmer does not have to consider the possibility of interrupts within a BASIC-language statement.

Consider, for example, the remote terminal keyboard input operation. The complete operation consists of the following steps (assuming an appropriate prompt has been sent to the terminal display medium):

1. Enable the remote terminal keyboard. In the full-duplex operations supported by the ASI, this means that subsequent keystrokes will be stored in the ASI input buffer and echoed back to the terminal.
2. Wait (or perform other processing) while the user enters the input data. The ASI performs the minor editing operations of backspace and cancel line.
3. The user terminates the input operation by striking the carriage return key. The ASI disables the keyboard (subsequent keystrokes are ignored) and notifies the operating system by means of a hardware interrupt.
4. The system notifies the applications program by means of an emulated interrupt that the input line is available.
5. The applications program accepts and processes the input.

TIO implements a single BASIC statement, ON INPUT, that initiates this process. The statement not only enables the remote terminal's keyboard (step 1 above), but specifies the applica-tions program's interrupt routine (information used by the system in step 4). The program may then continue processing or may execute a WAIT statement, which suspends processing until the next interrupt. Once the input is available (step 5), the program invokes the intrinsic string function AREAD$, which returns the contents of the ASI input buffer.

Remote device output is accomplished by the same BASIC statements used for local device output, such as the PRINTER IS, PRINT, and PRINT USING statements. Only the logical device identifier changes. Whereas local HP-IB printers are associated with identifiers in the range zero through seven, remote device identifiers are in the range 11 through 20. An important characteristic of remote output, in fact, is that applications programs require no changes other than specifying the correct identifier when the print output device is changed from a local printer to a remote RS-232C printer.

The TIO DROM also implements the ON OUTPUT statement to allow scheduling of concurrent output operations to several remote terminals or printers. The ON BREAK statement enables the applications program to detect breaks in the data link, and the ON CONNECT and ON DISCONNECT statements allow detection of the physical condition of the link in the case of RS-232C direct connections, and the processing of incoming data phone calls through an auto-answer modem.

There is a utility program, an extension of CONFIG (see article, page 14), for configuring the remote devices to be serviced and supported from the HP 250 via ASI. Ports can be assigned to the type of device (remote console, terminal, printer, or computer) with associated memory requirements, protocol, and identifier specifications where appropriate. This assigned configuration is updated to the system disc and is loaded with the operating system software at power-on. It needs modification only when the topology of the system is to be changed.
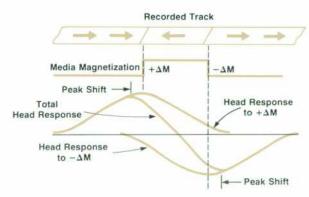
caused by noise or recording nonlinearities. Thus the PLO acts as a flywheel to smooth out the higher-frequency phase variations and provide a relatively stable synchronized read clock signal. The PLO output consists of two phases, one representing the clock window and the other representing the data window, and these read-clock phases are used to clock the data-separator circuit, which sorts the incoming data stream into ones (transitions occurring in the data window) and zeros (either no transitions or transitions occurring in the clock window).

Ideally, the incoming transitions occur in the centers of their respective windows. In reality, phase noise caused by the imperfect head-disc interface, electrical and magnetic noise, and the nonlinearities of magnetic recording disperses the transitions from the window centers and causes them to be distributed according to Gaussian or normal probability distributions centered within the clock and data windows, as shown in Fig. 7b.

There is a finite probability that data bits can be shifted enough to fall within an adjacent clock window, or that clock bits can be shifted into a data window. Either will result in data errors. The data error rate is very nearly equivalent to the sum of these probabilities. Since it makes little difference whether an error was caused by a dislocated clock bit or a dislocated data bit, we can envision a composite window encompassing a single probability distribution (Fig. 7c). The data error probability is equivalent to the area under the probability distribution outside of this composite window.

Some factors that cause transitions to be dispersed from the centers of their windows are predictable, so their effects can be minimized. One such factor is bit spreading, which occurs when the opposite-polarity magnetic fields of adjacent transitions in the media subtract from each other. If two transitions are closely spaced on the disc but are more widely separated from other transitions, then, as illustrated in Fig. 8a, the fields of the two transitions subtract from each other on the neighboring sides so the resultant field (as detected by the head during read) has peaks that have a slightly wider separation than do the corresponding magnetic transitions in the media. Since this phenomenon is a function of the bit pattern written and is predictable, we can precompensate for this peak shift phenomenon by detecting those patterns that would result in bit spreading

before writing, and arbitrarily writing those adjacent transitions closer together (Fig. 8b). Then, when the pattern is read from the disc, the detected peaks of the transition fields occur with the correct spacing. The HP 250 controller effectively precompensates all bit patterns that would otherwise result in significant bit-spreading effects when read.

One of the primary objectives in the design of the flexible disc subsystem was to achieve high data reliability, and a large share of the design effort went toward optimizing system parameters to achieve that goal. One data reliability design goal was to achieve a soft (correctable) error rate of less than one error per 10^9 bits read. To measure error rates of this magnitude with acceptable resolution requires that great amounts of data ($>>10^{10}$ bits) be read from the flexible disc. Since this is extremely time-consuming, an accelerated error rate test was devised. During a read, programmable delay lines artificially advance or delay the bits (transitions) going to the data separator with respect to the PLO-generated clock and data windows. By this means the transition probability distributions are shifted from the window centers by an amount equal to the programmed delay, and the probability of errors (and hence the error rate) is much higher. Thus, relative quantitative error rate measurements can be made in a matter of minutes rather than days, and the relative effects of various system parameters upon error rates can be easily determined so those parameters can be optimized for minimum error rates. This technique was used extensively during the development of the flexible disc memory for the HP 250.
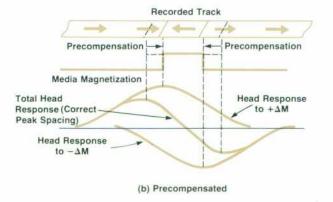

Fig. 8. *Bit spreading causes some closely spaced transitions to appear farther apart when read. Precompensation minimizes this predictable effect in the flexible disc system.*

One of the features designed into the flexible disc subsystem to insure high data reliability is the read-verify capability. A special margin-detect circuit in the data separator, enabled whenever the read-verify function is activated by the host system, detects whether any transitions occur within 100 nanoseconds of any window boundary. This indicates a margin-error condition and implies that the data, although presently readable, may not be recoverable should the error margin deteriorate further because of media/drive interchange, environmental variations, or wear of the flexible disc media. This usually indicates that the disc is approaching wearout, or has otherwise been damaged or contaminated, and that replacement of the disc is desirable.

Another feature is track-sparing, in which disc tracks containing defects are detected and marked defective ("spared"), and are thus prevented from being used for data storage. When a new disc is formatted, four worst-case data patterns are written and read-tested using the read-verify function. Should a margin error occur on any track, then that track is spared. If more than four tracks of a disc are spared during formatting, the system will reject the disc and abort the formatting routine. This assures that only good flexible discs will be accepted by the system, insuring the highest possible data reliability.

Reliability and Service

The HP 250 does not require a special air-conditioned or humidity-controlled environment. In addition, it is not a must to have a specially filtered power line or antistatic carpeting. The HP 250 will operate in 10°C to 40°C temperatures with 30% to 80% relative humidity. It has been tested with 15,000-volt static discharge pulses and is designed to meet worldwide safety requirements.

Easy and rapid service was considered a must in the design of the HP 250. Each time the unit is turned on, it sequences through a diagnostic self test to ensure proper operation of the important subassemblies in the computer. An indication on the CRT tells what is presently being tested. If the test is successfully passed, this is noted on the CRT and testing of the next subassembly is begun. The processor in the flexible disc controller independently runs a self test of the flexible disc subsystem at power turn-on, and returns the result of this test to the main processor when requested during the main system self test. If all tests are completed, the operating system is loaded into read/write memory from the system mass memory. This procedure is completed in less time than it takes the CRT to warm up and is normally invisible to the user.

If a subassembly does not pass the test, one of two things can happen. One is that the program recognizes this failure and returns the word "fail" to the CRT next to the subassembly name. The program then halts and the display is held on the screen. This then gives the customer or service person rapid information to identify the defective subassembly. In some cases the program may get lost and never return from the test of a defective subassembly. This also gives good information since the name of the subassembly under test is on the screen.

The service person also has other self-test diagnostic capability available. By modifying a rotary switch setting

on the CPU board, the service person can run a more extensive memory test, keyboard test, or CRT test to further verify performance.

Acknowledgments

The HP 250 benefitted from an enthusiastic and dedicated electronic hardware design team. Major design contributors were Dennis Peery, John Forman, John Leonard, Jay Seaver, Bob Emmerich, Bob Chalstrom, Craig Miller, Sterling Mortensen, and Gordon Nuttall. Guiding this development were the project manager, Clair Nelson, and section leader, Russ Sparks.

References

1. W.D. Eads and D.S. Maitland, "High-Performance NMOS LSI Processor," Hewlett-Packard Journal, June 1976.
2. J.E. DeWeese and T.R. Ligon, "A NMOS Process for High-Performance LSI Circuits," Hewlett-Packard Journal, November 1977.
3. J.C. Keith, A.K. Vogen, and L.T. Schulte, "System 45 Hardware Design," Hewlett-Packard Journal, April 1978.
4. B.E. Forbes, "Silicon-on-Sapphire Technology Produces High-Speed Single-Chip Processor," Hewlett-Packard Journal, April 1977.
5. J.W. Figueroa, "PHI, the HP-IB Interface Chip," Hewlett-Packard Journal, July 1978.

V. DeLloy Forbes

DeLloy Forbes joined HP's Santa Clara Division in 1969, doing analog and digital circuit design, then transferred to Loveland Instrument Division, where he designed the dc input amplifier, attenuator, and ohmmeter for the 3455A DVM. He recently moved to Fort Collins Division, where he was project leader for the HP 250 Business Computer input/output system, floppy disc subsystem, and keyboard. Born in Idaho Falls, Idaho, DeLloy received his BSEE degree in 1969 from Brigham Young University and his MSEE degree in 1971 from Stanford University. He is named inventor on a patent for the 3455A autocalibration algorithm. DeLloy lives in Loveland, Colorado, with his wife and their three children, and spends much of his spare time backpacking, skiing, taking photographs, playing piano, and directing a church youth group.

Gerald L. Meyer

Raised in Custer, South Dakota, Gerry Meyer is a 1965 BSEE graduate of the South Dakota School of Mines and Technology and a 1966 MSEE graduate of Purdue University. Gerry was a project manager for the HP 250 Business Computer electronics and designed the display logic for the HP 250. Prior to joining HP in 1976, Gerry worked as a design engineer on government aerospace communications electronics. He is named inventor on a patent for a fast phase-locked loop synchronization technique. A resident of Loveland, Colorado, Gerry is married, has two children, and enjoys camping, hiking, and skiing in the Rocky Mountains. He also works with local schools for the concerns of retarded children.

HP 250 BASIC: A Friendly, Interactive, Powerful System Language

by Dennis L. Peery

THE HP 250 IS A BASIC-LANGUAGE machine. System commands and BASIC-language statements are syntaxed and executed on the same level. Most statements may be executed either from the keyboard or from a program.

The HP 250's execution of user programs is interpretive, in the sense that the form of the program in read-write memory—the internal form—is not directly executable machine code, and the program can be listed essentially as the user typed it in. However, the internal form of a user's program is not at all similar to that of most interpreters, which retain the character form of the user's program and check operator precedences and variable names each time a statement is executed. Instead, the program is stored as pointers to the proper execution code for each language operation (such as LET, *, or READ), as pointers to a symbol table for variables and constants, and as pointers to a scratchpad location for temporary results. Execution time is reduced drastically by operating on the internal form rather than on the character form of a user's program.

HP 250 BASIC has the standard features of HP Business BASIC: multi-character variable names and line labels (up to fifteen characters), numeric and string arrays (up to six dimensions), independent subprograms and multiline functions, and formatted output via the PRINT USING statements. These features have already been discussed in these

pages in the article on the HP System 45 (April 1978).[1] This article will discuss some of the special features of the HP 250.

CRT/Keyboard

The HP 250 CRT is entirely interactive. Data or program lines from any line on the screen may be entered into the system. When the ENTER key is pressed, the entire current line (the line in which the cursor resides) is passed into the system.

The CRT logical length is dynamic. As much memory space is allocated for use by the CRT as is currently available. In a 64K-byte user memory system, for example, there might be more than 1000 lines (20 characters per line) available for CRT display. A minimum size of 1000 bytes is reserved for use by the CRT dynamic buffer.

Four video enhancements are easily accessible from the keyboard: inverse video, half bright, blinking, and underline. Using the CONTROL key and one of four special function keys, enhancements can be independently toggled on or off at any time.

Video enhancements can be included in strings without special escape sequences. The statement

<p style="text-align:center">1000 PRINT "<u>Underlined String</u>."</p>

can be listed, modified, and stored exactly as it appears. Enhancements are automatically converted to the proper escape sequences for printers and terminals.

The HP 250 has 27 special characters that correspond to all of the display control keys on the keyboard. These keys may be accessed in a display functions mode that allows the keys to be displayed and not executed. Like video enhancements, the special display control characters can be stored in strings like ordinary characters. A common statement is:

<p style="text-align:center">1000 DISP "⌧ ⌐ ";</p>

This statement clears the CRT and leaves the cursor in the upper left hand corner.

Internally, the 250 uses an eight-bit coding scheme for the character set. However, eight bits are not enough to support the display control characters, line drawing characters, and European characters along with the normal ASCII character set. Therefore, some strings contain characters indicating a switch to a different character set. These extra control characters are troublesome for European applications. Therefore, a European character set can be configured as the default alternate character set in the operating system so that European text has no extra control characters.

User-Definable Keys

Along the bottom of the CRT are eight unlabeled keys called soft keys. These eight keys, along with eight more unlabeled keys in the top left section of the keyboard, are the HP 250 special function, or user-definable (UDF) keys. There are separate definitions in shift and unshift modes for the eight keys on the keyboard, so altogether there are 24 UDF keys. The UDF keys can be used for ON KEY program branching or as typing aids.

When the soft keys are defined (UDF keys 1-8), labels appear on the CRT immediately above the defined keys.

Fig. 1. *Eight keys under the HP 250 CRT display can be defined differently in different applications programs to allow the operator to select various functions. The functions appear on the CRT above the keys. Also shown here is an example of a form in which data entry and display are programmed using standard* INPUT, ENTER, *and* DISP *statements.*

When soft key definition labels exist, the bottom three lines of the CRT are reserved for these labels, reducing the effective length of the display screen to 21 lines.

ON KEY

One of the must prominent features of HP 250 BASIC is the implementation of the ON KEY statement. Like the HP System 45, the HP 250 can initiate program branches (GOTO, GOSUB, CALL) via the UDF keys. An HP 250 extension to the ON KEY statement is programmatic labeling of the soft keys. Since the keys are dynamically defined, they can be used to guide a user through a set of soft key menus to select the desired program operation. An example of a program to use the soft keys is shown below and the resulting display is shown at the bottom of the CRT in Fig. 1.

```
100 ON KEY#1: "EDIT STOCKROOM" GOTO Edit__stockroom
110 ON KEY#2: "EDIT ACCOUNT'G" GOTO Edit__accountg
                        .
                        .
                        .
170 ON KEY#8: "ENGRG MENU" GOTO End
```

When line 100 is executed, it defines the label of soft key 1 and its program branch action. Similarly, lines 110 through 170 define keys 2 through 8. In this case, if the user presses key 1, the current input state is terminated and the routine Edit__stockroom is invoked.

Typing Aids

UDF keys can also be used as typing aids to reduce a complex series of up to 160 keystrokes to one key. A set of typing aid definitions can be stored and loaded from mass storage as a file, so that keys do not have to be continually redefined. Typing aid definitions may be listed on the CRT or a printer.

A useful UDF key feature is the ability to store action keystroke sequences and delay their execution until the UDF key is pressed. Action keys include the ENTER and

EXECUTE keys, the cursor control keys, and others. With a combination of action keys and alphanumeric characters, UDF key definitions can resemble small programs written in the language of keystrokes.

A simple example of delayed execution of action keys is a key that clears the CRT and lists the current program. The key definition is:

$$\mathbf{r_{R}} \; \mathbf{C_{L}} \; LIST \; \mathbf{X_{E}}$$

The action keys SHIFT HOME ($\mathbf{r_R}$) and SHIFT CLEAR ($\mathbf{C_L}$) are stored before the character LIST. This causes the entire CRT display to be cleared and the sequence of characters LIST to appear on the first line of the CRT. The action key EXECUTE ($\mathbf{X_E}$) causes the current line to be executed. Thus, the screen can be cleared and the program listed simply by pushing one key.

Mass Memory

The HP 250 supports a variety of devices and file types. Three types of files can be created directly by the user. These are program files (PROG), data files (DATA), and key files (KEYS). Binary program files (BPRG) can also be created (by combining existing BPRG files), but the user cannot write binary programs. File types used by the IMAGE/250 data base manager are backup files (BKUP), root files (ROOT), and data set files (DSET). Form files (FORM) can be created and purged only by running special utility programs. DROM and SYST files are reserved for system use and cannot be created by the user. DROM files are special system files that contain DROMs, or discrete relocating option modules. These consist of optional system software such as the IMAGE/250 statements. DROM files are accessed *only* when the system is loaded. Files created by a system other than the HP 250 are catalogued as OTHR and are not directly usable without special system software.

The only type of file that can be used for general-purpose storage by the user is the DATA file. This general-purpose file can have a logical record size that is independent of the physical record size, and can be read or written serially or directly. The user can address the file word by word, if necessary. DATA files can contain any combination of numeric and string data. The system stores programs in text format as strings in a DATA file.

Files may be protected to allow restricted access to DATA files and prevent accidental purging. PROG files may be also be RUN-ONLY. A RUN-ONLY program is simply a PROG file stored with a special protect code. The special RUN-ONLY protect code, which cannot be generated by the user directly, means that RUN-ONLY program files cannot be purged or copied without running a special utility program. A RUN-ONLY program also has special characteristics when it is run by the user (see "System Utilities" below).

Printed Output

The HP 250 printer select statements PRINTER IS, SYSTEM PRINTER IS, and PRINT ALL IS are used to direct all non-mass-storage output. The select code values accepted by the printer select statements are 0 to 20. Optionally, each printer select statement can specify a print file instead of a print-only device.

The main operating system supports printers connected on select codes 0-7. Select code 8 is the CRT and select code 9 is a "bit bucket" or null device (output is ignored by the system). Output can be sent to a mass memory file instead of a printer if the user specifies a file name instead of a select code. Using the COPY statement, hard copy of stored output is easily obtained.

The PRINTER IS statement applies only to output from the PRINT and PRINT USING statements. The SYSTEM PRINTER IS statement applies to system output such as that generated by CATALOG, TRACE, program stepping, and LIST. The PRINT ALL IS statement applies to all CRT interaction. The PRINT ALL printer will produce a hard copy of all input and output that takes place on the CRT. The PRINT ALL IS statement can be used to produce an audit trail.

The ease of use of video enhancements and alternate character sets forces the system software to deal with sending many types of control characters to printers. Nonstandard printer enhancements are masked off (inverse video, blinking, etc.). The underline enhancement is treated differently depending on the type of printer. For all system supported printers, underlining is automatically done by system software. Alternate characters sets are also treated differently depending on the type of printer. If the printer is not standard and a DROM driver is not included with the system to support it, all characters from alternate sets are replaced with blanks.

Program Development/Debugging

The HP 250 has several features that aid in program development and debugging.

The TRACE features, accessed via the TRACE, TRACE VARIABLES, TRACE PAUSE, and TRACE ALL statements, are identical to those on the HP System 45.[1] Messages given by the TRACE features are directed to the device designated as the system printer. This printer may be the CRT, a disc file, or a printer, and is specified by the SYSTEM PRINTER IS statement.

Single-step capability is also built into the HP 250. The HALT key doubles as the STEP key when a program is halted. Each time the HALT/STEP key is pressed, one program line is executed and the next program line is output to the system printer.

Routing the TRACE and single-step output to the system printer allows the programmer to debug programs without disturbing either the CRT, which may contain a form, or the standard printer, which may be printing a report.

The interpretive, on-line programming environment of the HP 250 helps make program development a pleasure. Changes may be made to the program in memory in the middle of execution, and the program can be continued from where it was suspended.

When the system detects an error (and no ON ERROR condition was specified), it suspends the program at the current line. The programmer has the option of making a correction to the program and continuing the program at the line that caused the error, starting the program over at the beginning, or just ignoring the error and continuing execution of the program. Program malfunctions may also be detected and corrected in this way.

To correct the program error, a programmer might take advantage of the following features:

1. The HALT key may be pressed to suspend execution of the program.
2. Program lines are directly accessible through the LIST or FETCH commands, and can be displayed on the CRT for examination.
3. Changes may be made in a program line displayed on the CRT, and the line may be reentered into the program by pressing the ENTER key.
4. The programmer may branch back to the statement that caused the problem or to some previous statement and reexecute that part of the program by single-stepping through the questionable lines.
5. If the changes correct the problem, the program can be RE-STOREd immediately. RE-STOREing the program can be done without affecting the state of the program in memory.

After taking any or none of the above actions, the programmer can continue from any location until another problem is found.

System Utilities

The HP 250 operating system is supported and enhanced with BASIC-language utilities supplied by HP. These utilities are RUN-ONLY programs and may also include binary programs. A RUN-ONLY program cannot be listed or modified in any way by the user. One of the utilities, ROUTIL, allows the user to produce RUN-ONLY programs from ordinary programs. The procedure is not reversible. Thus, software can be totally protected from tampering by making it RUN-ONLY.

Some important utilities provided by HP are CONFIG (system configuration), DUPL (disc duplication), INIT (disc initialization), EDITOR (a SCHEMA text editor), and two FORMS utilities: CFORM and MFORM. These utilities and many others are essential parts of the HP250 software system.

System Loader

The HP 250 operating system is loaded into memory from a SYSTEM file every time power is turned on. A ROM-based routine loads the system from a flexible disc. Once the system is loaded, the system disc is no longer needed, since the operating system is completely memory-resident.

Before loading the system, the loader determines whether enough memory is available for the system and the user. There must be one full 64K-byte block and at least two 32K-byte upper half-blocks of memory present. Based on the results of the self-test memory check, the loader decides which blocks of memory will be used for the system. In many cases, dynamic loading allows the system to be loaded and run despite memory failures. If the system cannot be loaded because of a lack of memory, the loader can load a diagnostic program that requires only 32K bytes of memory.

A directory search of available flexible discs locates the system file. The directory search allows this file to reside in mass memory as an ordinary file. Thus, dedicated tracks are not required for the system file and the mass memory medium containing the system file may reside in any mass memory drive.

After loading, the loader passes pertinent information (such as available user memory, software block numbers, and the mass memory device containing the system file) to the operating system and transfers control to the operating system. The time for the system load is about ten seconds from a flexible disc.

System Power-Up and Configuration Utility

After the system self-test and loading procedure has been completed, the system power-up procedure is initiated. At this point, the power-up code is present in memory along with most of the resident operating system. After the power-up code is executed, it is overwritten by a final section of the resident operating system that is not needed during the power-up procedure.

The power-up procedure involves the following:
1. Checking the current system configuration against the configuration specified in the system file.
2. Initializing system and user memory areas.
3. Identifying and initializing I/O devices.
4. Loading the configured DROMs and keyboard tables into system memory.

A utility program called CONFIG allows the user to specify a number of system attributes. This utility provides a means of DROM and foreign language selection, user memory allocation, I/O device configuration, and auto-start command selection.

Attributes of devices connected to the HP 250 via the asynchronous data communications ports are also specified through this program. Any communications via these ports will assume the device attributes given by the configuration tables at power-up time.

During power-up, the information specified via CONFIG is resident in memory along with the power-up code. This information is checked against the actual attributes of the system and any discrepancies are reported on the main console CRT at the end of the power-up sequence.

After the system configuration is checked and the system and user memory areas are initialized, the optional system software modules (DROMs) are loaded into system memory. The user may specify via the CONFIG program which DROMs are to be loaded. Examples of DROMs available with the HP 250 are IMAGE, SORT, and FORMS. DROMs are individually loaded to allow the HP 250 to be customized for a particular application.

Unlike the resident language system, which must be loaded into fixed addresses, DROMs are relocatable modules and may reside starting at any address. Each DROM, however, must reside in one contiguous block of memory. DROMs are stored in individual disc files (of type DROM) on the same medium as the system file.

Also loaded during power-up are the keyboard tables. These may be standard U.S. ASCII, the line-drawing set, or any of the foreign keyboards available with the HP 250. All available keyboard tables are part of the system file so that one operating system contains all of the information needed to support any foreign language.

After the entire power-up procedure is completed, the power-up code and its accompanying data areas are overwritten with the final section of the resident language system and the system is brought to life.

A feature added especially for business applications is the auto-start capability. At the conclusion of power-up,

any single command, specified via the CONFIG program, may be executed. Typically, this command will be a RUN command that will start an applications program. This capability means that all the user must do to bring up the system and start the applications package is turn the key on the HP 250. Everything else is automatic.

FORMS/250

FORMS/250 is a data entry and display package designed to simplify applications programs that require data entry and display, provide a convenient and fool-proof interface to the user, and provide a degree of independence of the CRT format from the applications programs. The form package consists of a FORM DROM, which contains operating system and language extensions, and two BASIC utility programs for the creation and modification of forms.

A form is just a CRT image combined with some information that gives the form special properties. The CRT image consists of any number of protected lines containing text input fields, or output fields. The input and output fields are the only areas in the protected lines that are accessible to the user or program. The distinction between input and output fields is made to allow forms to be used conveniently for data display as well as data entry.

The auxiliary information consists of three order tables, one each to define the input order, output order, and tab key order. Since the fields are referenced by field number in an applications program, these tables isolate the program from the physical location of the fields on the display, and from the order in which the user accesses them via the TAB key.

The form can exist on the disc as a special FORM-type file, accessible only through the FORM DROM or the utility programs. The CRT image that makes up half of the form file is literally that—a direct copy of a section of the HP 250 display buffer. The advantage of storing the form in internal CRT format is that it is fast. The file is placed into the CRT display buffer by direct memory access, eliminating word-by-word processing when the form is fetched from the disc.

The GET FORM statement, which exists in the FORM DROM, is used by a BASIC program to activate and display a form on the CRT. Before the form image is loaded into the display buffer, the field ordering tables are placed in a dynamic buffer in user memory. Any access to fields on the form is routed through a form handler routine. This routine exists in the FORM DROM, and becomes the manager of the ordering tables.

The usual approach to adding features to BASIC for input and output is to add new statements with new keywords. While FORMS/250 has a few new keywords for the activation and termination of forms, data entry from a form and display to a form is done with the standard INPUT, ENTER, and DISP statements. The attempt was made to make input and output via fields on a form a logical and consistent extension of standard input and output via normal CRT lines. For example, as shown in Fig. 1, the statements

```
100 GET FORM "FORM2"
200 DISP A$,B$,...
300 LDISP "Edit the data as desired."
400 Input: INPUT
500 IF TFNUM <>8 THEN Input
600 ENTER A$,B$,...
```

display the form on the screen, fill the form with data to be edited, tell the user what to do, allow the user to edit any data in the form, and finally, enter data into the program variables.

The CFORM and MFORM programs are special RUN-ONLY BASIC utility programs for creating and modifying form files. They both contain convenient facilities for editing text, input and output fields, video enhancements, and field ordering. MFORM is actually a subset of CFORM, allowing only modifications that will not disturb the operation of the form with existing programs that may use it. For example, text can change, fields can be moved, the tab key order can be changed, and so on, but fields cannot be added, deleted, or changed in size, since this would affect how the form operates with a program. This distinction gives the programmer complete control over form development with CFORM, while also giving the less sophisticated user a no-risk way to change the cosmetics of the forms with MFORM.

REPORT WRITER/250

One of the major areas of concern while developing the HP 250 operating system was the problem of producing reports. Since the HP 250 was designed to be a business computer, it was estimated that well over half the application programs to be written would produce some kind of a report.

To address this problem it was decided to provide a unified set of BASIC-language commands in a DROM format known as REPORT WRITER/250. REPORT WRITER/250 is designed to alleviate many of the programming burdens of report production by providing special facilities for page and line counting, break recognition and branching, form page size determination, report formatting, and printer independence. Among its capabilities are the following:

Break recognition. A relatively sophisticated BREAK WHEN statement was developed to handle the problem of break recognition. A TRIGGER BREAK statement is also provided to handle cases where the break condition is too complex to be described in the BREAK WHEN statement.

Hierarchical structure. Since most reports have a series of subtotals, a feature called "levels" is provided so that a programmer can describe the hierarchy of the breaks and subtotals involved in the report.

Self-triggered blocks of codes. Along with breaks and hierarchy came the necessity to allow the programmer to insert as many lines of BASIC program code as needed when report conditions are triggered. This output-producing code is triggered automatically by REPORT WRITER/250 once it recognizes the break.

Line and page counting and printer independence. REPORT WRITER/250 provides comprehensive line and page counting capabilities. All the programmer need do is document the page size and REPORT WRITER/250 does all the line and page counting. Thus reports are independent of the vertical page size. These counting routines are printer independent: any REPORT WRITER/250 program will operate with any acceptable HP printer.

Other features. REPORT WRITER/250 automatically pro-

vides the programmer with several other features, such as restart, summary reporting capabilities, exception reporting capabilities, automatic totaling and averaging, and moveable left margin.

Consistent, self-documenting code. REPORT WRITER/250 provides a secondary although significant benefit in that the programmer must define the report in a straightforward, consistent manner. This makes all reports self-documenting and easier to maintain.

Acknowledgments

I would like to extend appreciation and recognize the following people for their contributions to the design and implementation of the products described here and for their contributions to this article: Jeff Lindberg, Pat Ryan and Tim Eckerman for the language system, Don Stavely and Art Boyne for FORMS, and Al Kandell for REPORT WRITER.

References

1. W.D. Eads and J.M. Walden, "A Highly Integrated Desktop Computer System," Hewlett-Packard Journal, April 1978.

Dennis L. Peery
Raised in Orem, Utah, Dennis Peery received his BSEE degree in 1971 from Colorado State University. Dennis joined HP in 1965 as a production test technician, then worked as an integrated circuit design engineer, and recently became project manager for the HP 250 Business Computer data communications and multi-user extensions. Dennis, his wife, and their four children (ages 15, 13, 8, and 6) live in Loveland, Colorado. A real sports enthusiast, Dennis enjoys riding motorcycles and playing basketball and racquetball. He has also been a member of the National Ski Patrol and the U.S. Volleyball Association.

Low-Cost Data Base Management

by Michael V. Hetrick

A JULY 1974 HEWLETT-PACKARD JOURNAL article entitled "Powerful Data Base Management Systems for Small Computers" introduced IMAGE, a sophisticated data management system for the HP 3000 Computer.[1] The same title could be used here, except that the much smaller computer now offering a version of IMAGE is HP's small-business computer, the HP 250.

IMAGE/250 is a collection of utility programs and BASIC programming statements that may be used to define, manipulate, and maintain a data base. In many ways, IMAGE/250 is identical to IMAGE/3000, although certain specifications were reduced and some operational modes were eliminated (see Table 1). IMAGE/250 is not strictly a subset of IMAGE on HP's larger computers. However, a design goal was to offer a common method of data base management so that the IMAGE/250 techniques apply to other computers offering IMAGE.

This article briefly compares IMAGE/250 and IMAGE/3000, and then concentrates on the enhancements added to IMAGE/250 to improve its performance on a flexible-disc-based computer.

IMAGE/250-IMAGE/3000 Comparison

Data base operations are organized into three major categories: definition, manipulation, and maintenance. The methods used to perform these operations on the HP 250 are similar to those on the HP 3000, as indicated in Table 2. The computers' data base definition languages, in which inter-

Table 1. IMAGE Feature Comparison

	IMAGE/250	IMAGE/3000
Data items per data base	255	255
Data sets per data base	50	99
Characters per item name	15	16
Characters per set name	15	16
Data items per data set	127	127
Maximum data entry size (master)	1018 bytes	4094 bytes
Maximum data entry size (detail)	1022 bytes	4094 bytes
Detail data sets per master data set	8	16
Entries per chain	65534	65535
Entries per data set	65334	8.3M bytes
Security passwords	31	63
Security level	set	set and item
Compound data items	yes	yes
Sorted chains	no	yes
Data transfer	entry	entry or item
Multi-volume data base	yes	no
DBOPEN modes	3	8
DBGET modes	4	8
DBLOCK:		
Data base	yes	yes
Record	no	yes
DBERROR	no	yes
DBEXPLAIN	no	yes
Data base restructuring	yes	yes

relationships between data sets are established, are practically identical. The data base structure, already familiar to IMAGE/3000 users, is a network organization in which indexes, known as master data sets, allow rapid access to data stored in detail data sets.

Table 2. IMAGE/250 Data Base Management Commands and Utilities

DATA BASE MANIPULATION:	DATA BASE DEFINITION:
DBOPEN	EDITOR
DBCLOSE	SCHEMA
DBLOCK	
DBUNLOCK	DATA BASE MAINTENANCE:
DBINFO	
DBPUT	DBCREATE
DBDELETE	DBERASE
DBUPDATE	DBPURGE
DBGET	DBSTORE
DBFIND	DBRESTORE
	DBLOAD
DBASE IS*	DBUNLOAD
IN DATA SET*	
WORKFILE IS*	DATA BASE INQUIRY:
FIND*	
SORT BY*	QUERY

*No HP 3000 equivalent

IMAGE/250's data base manipulation statements—those used to add, delete, modify, and retrieve data entries—are similar to the IMAGE/3000 intrinsics. On the HP 3000, the manipulation language is a set of procedures that are callable from several host languages. On the HP 250, it is a set of BASIC statements. Therefore, the CALL prefix was dropped from the statement syntax, but the parameters remain the same.

IMAGE/250 and IMAGE/3000 maintenance operations are functionally similar, although most IMAGE/250 operations are performed through BASIC statements. The BASIC language capability has many practical uses on a small business computer. For example, data base backup operations involving the DBSTORE and DBRESTORE statements can be built into a BASIC program. The program can direct the operator to insert and remove mass storage volumes, prompt for information, and generally lead the user through the entire procedure. Programs such as this provide a friendly and interactive environment for completing complex tasks. Creating, purging, and erasing data bases may also be done under program control.

IMAGE/250 Design Objectives

The IMAGE/250 design philosophy was influenced by the HP 250 architecture and the hardware design goals. This resulted in two major design objectives:
1. Implement a simplified version of IMAGE/3000 with as many essential features as possible while providing enhancements for optimal performance in a business data processing environment.
2. Extend the allowable data base size beyond the on-line mass memory storage capacity. Since the standard mass memory configuration of two flexible discs offers a total storage capacity of 2.4M bytes, this was an important

objective.

In achieving the first objective, concessions were made by reducing data entry sizes, data set capacities, and other specifications. The number of ways to open and access a data base was also reduced. Three of the eight DBOPEN modes were implemented; this is sufficient to allow exclusive or shared data base access. In implementing four of the eight DBGET modes, the essential access methods (serial, direct, chained, and calculated) are supported.

The ability to create sorted chains while adding records was dropped. This is a disc-intensive activity that seriously degrades throughput, especially on the low-speed flexible discs. Instead, innovative search and sort routines that function efficiently in the HP 250 memory configuration were devised (see "Data Base Sorting" below).

Two additional simplifications were to provide password protection only at the data set level, not the data item level, and to transfer data to and from the data base only in the full record mode. Data set password protection, in which a user with read or write access to a data set has read or write access to all information in the set, is not as versatile as IMAGE/3000's item security scheme. However, in the HP 250 operating environment, where relatively few users are involved, security on the set level is adequate.

Supporting only the full record mode of adding, updating, and retrieving entries simplified the implementation of the corresponding manipulation statements: DBPUT, DBUPDATE, and DBGET. In these statements, information is transferred to and from the data base via a single string variable known as the buffer. The buffer contains data *exactly* as it is stored in the data base. It may contain any combination of alphanumeric characters and HP internally formatted numeric types (integer, short precision, or real precision). For example, to add a data set entry consisting of a ten-character name, an integer employee number, and a real precision salary, the following buffer contents are required for DBPUT:

A	n	
d	e	
r	s	name (padded with blanks)
o	n	

25038				integer number (2 bytes)
256				8-byte representation of $52,000 salary:
5	2	0	0	2 bytes of sign and exponent and 6 bytes of
0	0	0	0	binary-coded decimal digits (2 digits per
0	0	0	0	byte)

Retrieving an entry from this set through DBGET results in a similarly formatted buffer, that is, the first ten bytes represent a name, the next two bytes an integer, and the last eight bytes a real precision number in an internal BCD (binary coded decimal) format. It is the program's responsibility to make sense out of the buffer contents by converting appropriate substrings to values in properly dimensioned numeric or string variables.

Since this is a cumbersome task using conventional BASIC string manipulation commands, a new BASIC statement, IN DATA SET, was invented. This statement defines a relationship between the fields of a data set entry and the local program variables. If an IN DATA SET statement is

active for a particular set, the data set values retrieved via DBGET are automatically transferred to the program variables declared in the IN DATA SET statement. Conversely, program variable values are automatically transferred to the data set record during DBPUT and DBUPDATE. The program below illustrates how IN DATA SET facilitates data transfer during DBGET:

```
10      Dim Name$[10], Buffer$[20]
20      INTEGER Number
30      REAL Salary
           :
        (open data base)
           :
100     IN DATA SET "PAYROLL" USE Name$, Number, Salary
1010    DBGET (Base$, "PAYROLL",4,S(*),"@",Buffer$,500)
1020    PRINT Name$,Number,Salary
```

After line 1010 is executed, Buffer$ contains the internal image of the PAYROLL data set's 500th entry. At the same time, the variables referenced in line 100 by IN DATA SET are automatically updated with the appropriate Buffer$ substrings. Thus, when line 1020 is executed, the variables Name$, Number, and Salary already contain the field values for the 500th record—in this case, Anderson, 25038, and 52000, respectively. Through IN DATA SET, the burden of buffer unpacking (and packing) is removed from the program and controlled by IMAGE.

Extending Data Base Size

The second design objective was to extend the data base size beyond the on-line mass memory capacity. This was accomplished with these restrictions:

1. Individual data sets must reside completely on a single volume; multiple-volume data sets are prohibited. Although the maximum allowed data set size is over 16M bytes, the actual size is limited by the largest available space on the mass storage medium.

2. Up to 23 volumes may be allocated for the data sets. Any combination of flexible discs and HP 7906 and HP 7910 hard discs are acceptable. However, no more than ten data set volumes may be on-line at one time (per data base).

3. For a given data base operation, the ROOT file—the file containing all the data base structural information—and all required data sets must be on-line simultaneously. For example, to add detail data set entries, the mass storage volumes containing the detail data set and all related master data sets (as well as the ROOT file) must be on-line. This is an important consideration in the design of a multi-volume data base.

Data sets are assigned to volumes when the data base structure is defined through the data base definition language. As the schema processor compiles the data base definition, the volume names become a permanent part of the ROOT file.

When a data base is accessed through a program, only the ROOT file location (volume) must be specified. Data set volume names are not required, since IMAGE relies on the ROOT file for this information. A 20-byte volume table created when a data base is opened maintains a list of all the resident volumes in a concise two-byte format. This list is updated automatically as media are removed or inserted. Before an operation on a data set can be executed, IMAGE first determines which volumes are required and then examines the volume table for their presence. The operation proceeds only if all necessary volumes are on-line.

Midprogram volume replacements are acceptable, but require special attention. To minimize disc activity, information is transferred from an internal private buffer to the appropriate data set on a "need to" basis. Although a data base modification may be complete from a program's standpoint (i.e., the DBPUT, DBUPDATE, or DBDELETE statement has been executed) it is not completed internally until the private buffer is actually copied to the disc. Numerous events trigger this action. One is the need to transfer other information into the buffer for an ensuing data base operation; another is terminating data base access through the DBCLOSE statement. A special DBCLOSE mode (mode 4) also dumps the private buffer, if necessary, without terminating data base access. A mode-4 DBCLOSE should be executed before midprogram volume exchanges to ensure that the latest data base modifications are copied. Failure to do so may result in a corrupt data base.

Multivolume data bases pose potential integrity problems if a particular maintenance operation is not performed on all required volumes. The maintenance procedures are designed to function on a volume-by-volume (or set-by-set) basis. Anomalous results are avoided if all pertinent volumes are accessed while creating, purging, erasing, storing, or restoring a data base.

Data Base Sorting

A frequently performed operation in the business data processing environment is sorting. HP business BASIC does not support any direct construct for sorting of data, so a special DROM was developed to provide this facility on the HP 250.

The sort is designed specifically to process information in an IMAGE/250 data base. It is important that sorts can be performed not only on the entries in a single data set, but also on a subset of those entries. It is also important that the sort be able to extract information that might be scattered across many data sets.

SORT/250 operations supply information that allows the programmer to access the desired portions of the data base in sorted order. This information is in the form of pointers to the appropriate data base entries.

Subsets of the data set information are selected by the FIND statement. This statement has the syntactical form:

FIND ⟨numeric expression⟩,

where the numeric expression specifies the condition each record must meet to be retained for subsequent processing. This expression may contain any operator or function allowed in standard HP BASIC. Thus, substring comparisons, string searches, and even user-defined multiline functions may be used to select the appropriate set of records.

When a FIND statement is executed, pointers corresponding to the selected records are put into a special file called the workfile. This file resembles a regular HP BASIC data file. The pointer values may be read from the file using the
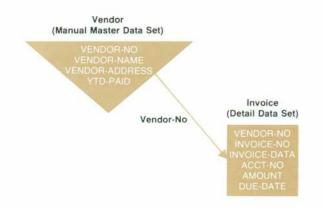
Fig. 1. *A simple accounts payable system designed to use the data base management capabilities of IMAGE/250.*

BASIC READ # statement. Direct-mode DBGETs may then be used to extract the actual data base information.

The SORT BY statement is used to sort information in the data base. Records in a data set can be sorted on any field in the set including subitems. If a FIND has previously been executed, only the records with associated pointers in the workfile that meet the selection criteria will be sorted. If a FIND has not been executed, all records will be sorted. In either case, the result of the sort is a series of data base record pointers stored in the workfile.

Sorting across Multiple Sets

SORT/250 provides facilities for dealing with structures that are more complex than simple data sets. Hierarchical structures consisting of up to ten sets can be described and manipulated. This kind of specification, known as a thread, consists of a list of the sets involved in the hierarchy along with information describing how they are interrelated.

A two-level hierarchy for a simple accounts payable system is shown in Fig. 1. It consists of a single manual master
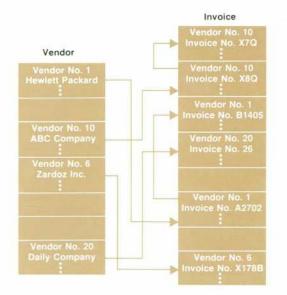

Fig. 2. *Information that might be contained in the data sets of Fig. 1. Pointers showing how the two sets are related are used by SORT/250 to retrieve information.*

data set containing the vendor information, and a detail data set that contains the invoices associated with each vendor.

Fig. 2 shows sample information that might be contained in the data sets. If the master set were sorted by vendor name, the pointers in the workfile after the sort would be:

$$3, 7, 1, 4$$

By retrieving these pointers sequentially and using them in direct-mode DBGETs, it is possible to produce a list of all vendors sorted by vendor name.

However, to produce the same report subsorted by invoice number would require access to information in both sets. If SORT/250 is given a thread structure that includes these two sets and it is told to sort by vendor name with a subsort on invoice number, it will oblige by returning pointer pairs. The first pointer will refer to a record in the master data set and the second will refer to a record in the detail data set. Performing such a sort for this sample data would yield the record pointer pairs:

Vendor pointer	Invoice pointer
3	1
3	2
7	4
1	6
1	3
4	8

The Sort Algorithm

Two primary concerns when writing a disc sort are processor time and the amount of disc I/O. SORT/250 works by first loading as many records as possible into memory, sorting them, and writing the sorted results to a scratch area (a workfile) on the disc. This process is continued until all records to be sorted have been put into one of these sorted segments. The sorted segments are then merged to produce the final sorted output. Using this method, the processor time is consumed primarily in sorting the segments, while reading records, writing segments, and merging segments are I/O intensive.

For maximum speed, it is desirable to reduce the burden on the I/O system and put as much work as possible on the processor. This is particularly true of flexible-disc-based sorts. One way to do this is to make the sorted segments large. This reduces the number of such segments, thus reducing the time spent in the merge procedure. It also means that larger sorts can be performed without a merge.

To give the internal portion of the sort as much memory as possible, the contents of the sorting user's memory are written to the workfile. At the completion of the sort the user's memory is restored to its original state. Regardless of the amount of memory possessed by a given user, only 122 sectors (approximately 31K bytes) are written to the workfile. This prevents the size of the workfile from increasing when the amount of user memory is increased beyond 32K bytes.

During the sort, the free memory is divided into areas for buffer space, pointers to sorted segments in the workfile,

pointers to records in memory, a table describing the thread and the sort keys, and the records themselves. However, since the ultimate output of the SORT BY statement is a set of pointers to data base entries, the whole record being sorted (or records for a multiset thread) need not be retained. Only the sort keys are kept, after being transformed into an alternate representation that makes them easier for the record compare routine to process. Specifically, the transformation is such that the keys can be compared as 16-bit unsigned integers. It is no longer necessary to know the type of a particular key during the compare. The compare does not need to check for special cases, or perform floating point arithmetic.

After a record is read (or group of records in the case of a multiset thread), the sort keys are extracted, put into suitable form for comparison, and passed, along with the appropriate record number(s), to the sort process. This construction, consisting of sort keys and record numbers, is called a work record. The sorted segments, which are written to the workfile preparatory to a merge, consist solely of sorted work records.

The sort process was developed for optimal use of the processor hardware. The internal sort algorithm is called "modified binary search and insert", or MBSI. Most of the processor time in a sort is spent moving memory or performing comparisons. The HP 250 processor has a limited form of multiword move instruction that allows a fixed number of words (from 1 to 16) to be moved. However, it has no multiword compare instruction. MBSI takes advantage of this by reducing the number of compares to a minimum at the expense of a greater number of moves.

For a normal binary search and insertion sort, the number of compares required is approximately:

$$N(\log_2 N - \log_2 e) + \log_2 e,$$

where N is the number of records to sort. The compare time is also directly proportional to the length of the work records being compared (in the worst case). Here N represents the number of records that can be sorted in memory, not the total number of records that are sorted after any merges are done. For SORT/250, $101 < N < 4459$, but the maximum number of records that can be sorted is 65534.

The worst case number of moves is about $(N^2 - N)/2$. To keep the move time down, the actual records are not moved. A vector of pointers to the work records specifies the sorted ordering. When a new record is added to the list, its position is established by inserting a pointer into the vector at the appropriate place.

One of the advantages of a binary search and insertion sort is that it does not require the simultaneous presence of all N records before the sort can begin. To sort record i requires that records 1 through $i-1$ be present and in sorted order. This corresponds nicely to the way the records are read off the disc. After record i has been read, a request can be made to read record $i+1$, and while waiting for the disc I/O to complete, the processor can be sorting the pointer to record i into its position in the pointer vector.

On fast discs, however, the read may be completed before the insertion is complete. Also, more than one record is often read in a single disc read request. Thus, there may be several records queued for handling by the sort process. Despite the overlapping of sorting and I/O, the N^2 factor of the sort process "shows through" in the sort times. To reduce this show-through to a minimum, the algorithm was modified somewhat. The N items in memory are broken up into K subsorts of N/K items each. The resulting K sorted groups of records are then merged to produce the same effect as sorting all N items. As a function of N and K, the number of compares remains the same as before, namely:

$$N(\log_2 N - \log_2 e) + \log_2 e,$$

which is independent of K. The worst-case number of moves becomes:

$$(N^2/K - 3N - K^2 + K + 2NK)/2.$$

The solution for the best K is relatively difficult. Solving a slightly different problem that does not use worst-case assumptions about moves required during the K-way merge yields a best value of $K = \sqrt{N}$. When substituted into the above equation this gives a number of moves corresponding to:

$$1.5N^{1.5} - 2N + 0.5\sqrt{N}.$$

Actually, K should be chosen as the integer part of ($\sqrt{N} + 1/2$), since the square root generally yields a nonintegral number. This improves the number of moves over simple binary search and insert for all N greater than 2. However, because of the form of the multiword move in the BPC, $K=16$ was chosen experimentally as near optimal.

This change, along with the overlapped I/O, virtually eliminates nonlinearities in the internal sort times. The remaining nonlinearities are caused by the final merge of the sorted segments on the disc.

If all the records to be sorted fit into memory simultaneously, the pointers can be extracted directly from the work records in memory and put into the workfile to complete the sort. This avoids a time-consuming disc merge operation. If more records must be sorted than can fit in memory simultaneously, the current contents of memory must be put into physically sorted order and written to a scratch area of the workfile. Then a new group of records can be sorted.

In certain cases, a single-pass merge is not sufficient to merge all sorted segments on the disc. When SORT/250 determines that an intermediate merge is required, it will perform one. A two-pass merge is rarely required. In any case, no segment that was the result of an intermediate merge is ever used in a subsequent intermediate merge. Such a merged segment is only read by the final merge.

Data Base Inquiry Facility

An inquiry facility is a computer program that allows data base access in an ad-hoc manner. It provides an attractive alternative to writing special-purpose programs to produce a one-time report or data modification. A good inquiry facility lets the user interact with the data base in a simple, natural fashion, and requires minimal knowledge of the data base structure and format. For example, the question, "What are the names and addresses of all customers in

Colorado?" should be answered simply and quickly.

The primitive operations that need to be performed on a data base are:

1. Select a subset of the data in the data base.
2. Print the selected subset.
3. Update the data base (additions, deletions, and modifications).

A realistic inquiry facility will, of course, offer extensions of these primitives to enhance its usefulness.

An inquiry facility is an example of a data-independent, data-driven program. It is data-independent in the sense that its functions (search for data, report data, modify data) are applicable to any data base. It is data-driven in the sense that the specific commands that are meaningful depend upon the data base (e.g., the command "list all employee names" is meaningless for an accounts receivable data base.) Custom (application) programs, on the other hand, are of the class of data-dependent programs, since they are designed to function with only one specific data base. In the design of computer systems, data-independent, data-driven programs are desirable because of their flexibility and universality. They offer the potential for a significant reduction in programming effort over custom programs, since their functions may be applied to any set of data. Some specific extensions that were implemented to allow an inquiry facility to replace simple custom programs are discussed below.

Query/250

QUERY/250 is an inquiry facility for IMAGE/250 data bases. It consists of 4500 BASIC program lines and uses the data base management functions of IMAGE/250 and SORT/250 to access the data base. The operation of QUERY/250 is based upon QUERY/3000, the IMAGE inquiry facility on the HP 3000. There are seventeen QUERY commands, which may be divided into four functional groups (Fig. 3). The selection commands are used for data modification as well as data reporting. The target users of QUERY were defined to be casual computer users[2] as well as computer programmers.

To accommodate both casual and expert users, a two-level command input structure was designed. A user who is familiar with the QUERY commands may type and execute command lines directly. For the user who is unsure of the command functions or syntax, QUERY labels the HP 250 softkeys with the QUERY commands. Pushing a softkey brings a form to the CRT containing a brief explanation of the action to be performed and a set of blanks to be filled in. As the blanks are filled in, QUERY composes the command at the bottom of the CRT. This use of the softkeys has two secondary effects. The keys are ordered to lead the user through the proper command sequence for reports and they also unobtrusively remind the user of the functions available in QUERY.

In the interest of a simple command set, the list commands contain no provision for user-specified formatting of reports. Reports may be formatted, however, by calling a report writer program from QUERY. The data base designer may also exercise limited control over QUERY report formats by using IMAGE/250 control numbers. A control number format specification is associated with an item in the data base schema. It allows control of the decimal point

Selection	Reporting	Modification	Utility
THREAD	SORT	ADD	DATA BASE
FIND	BREAK...TOTAL	DELETE	PASSWORD
	OUTPUT TO	REPLACE	INFO
	LIST		WORKFILE
	LINEAR LIST		DO
			RUN
			EXIT

Fig. 3. *QUERY/250 provides access to IMAGE/250 data bases without special programming. These are the QUERY command keywords.*

placement, the currency sign, commas every three digits, and the date format. Control numbers also contain a QUERY write-inhibit flag that prevents the QUERY user from modifying values of a particular item. The write-inhibit flag is extremely useful in preserving audit path integrity, since QUERY cannot automatically maintain audit paths.

The THREAD command allows a search path (defined in the schema) to be specified for a multiple-set access. The ability to perform a search across data sets significantly expands the range of problems QUERY can solve. This multiple-set access is possible because of the SORT/250 IMAGE enhancements.

Custom Program Replacement

Three functions were added specifically to make QUERY an attractive alternative to custom programs. They are: the ability to enter data from a CRT form, the ability to define and execute procedures, and the ability to load and call a BASIC subprogram.

The ADD command allows the QUERY user to input data via a CRT form that has been defined using FORMS/250. For repeated data entry this allows a cleaner and more intelligible CRT format than the set-and-item-name prompt QUERY uses when a form is not specified.

The DO command executes a previously defined sequence of QUERY commands. The HP 250 text editor is used to write the list of commands and store them in a DATA file. This function allows often-used procedures to be executed with a single command. The contents of a typical QUERY procedure are shown in Fig. 4.

The RUN command loads and calls a BASIC subprogram. This function was designed to allow the user to format output with REPORT WRITER/250 while still taking advantage of the simple, flexible QUERY data selection commands. The RUN command may also be used to execute a program that manipulates the data in the data base.

User Reaction

Initial user reaction indicates that QUERY is a useful programmer's tool. Applications cited include: verifying

```
DATA BASE ARBASE, ARDISC
PASSWORD CLERK
FIND CUSTOMER FOR STATE="CO"
SORT BY CITY, NAME
BREAK ON CITY
LIST NAME, ADDRESS, ZIP
```

Fig. 4. *A QUERY/250 procedure to list names and addresses of customers in Colorado from the accounts receivable data base.*

custom program correctness by examining the data base, checking the state of a data base, and correcting a data base that has been scrambled. QUERY may be used by casual computer users, depending upon their level of sophistication. The largest drawback of QUERY for the casual computer user is the necessity of understanding the data set and item names and the data base structure. The casual computer user expects to use QUERY mainly for ad-hoc reporting.

It appears the user-customization features added to QUERY to allow replacement of custom programs will not be often used. User comments indicate the features offered are too limited to be of much use in solving most problems. These features will find their major use in providing a quick, minimum-effort solution to simple problems that require mainly data storage and retrieval.

Acknowledgments

Many people contributed to the success of IMAGE/250. Fred Mann wrote the data base definition BASIC utilities while Harry Plate and Charles Hanes shared the data base maintenance commands and utilities. Al Kandell contributed IN DATA SET, and the remaining data base manipulation statements were written by Vince Griesemer, Harry Plate, and Fred Mann. Jerry Kaufman and Tim Gill designed QUERY and SORT/FIND, respectively. They also wrote major portions of this article, for which I am grateful.

References

1. R.E. McIntire, "Powerful Data Base Management Systems for Small Computers," Hewlett-Packard Journal, July 1974.
2. E.F. Codd, "Seven Steps to Rendezvous with the Casual Computer User," in Data Base Management, J.W. Klimbe and K.L. Koffeman, eds., North Holland Publishing Company, 1974.

Michael V. Hetrick

Mike Hetrick is project manager for the HP 250 operating system and participated in the definition and implementation of IMAGE/250. He also contributed to the hardware design of the 9815A Desktop Computer and was production engineer for the 9830A Desktop Computer. Prior to joining HP in 1973, Mike worked as an automatic control systems engineer for a midwestern electronics company. Born in Milwaukee, Wisconsin, Mike is a 1969 BSEE graduate of General Motors Institute and a 1970 MSEE graduate of the University of Colorado. The Hetricks—Mike, his wife, their daughter, and their two sons—live in Loveland, Colorado. Mike's outside interests include racquetball, basketball, softball, clarinet, and camping trips with his family.

Applications Software for the Small-Business Computer

by Scott W. Y. Wang and Loyd F. Nelson

TO SOLVE USER PROBLEMS, a computer system requires not only hardware and system software, but also applications software. But what applications should be provided? For the HP 250, a small business computer, the goal is to improve the user's knowledge of the business. Each business, of course, has its own set of characteristics and requirements, so the first step in applications development is to categorize the businesses and identify the functions that are common to enough businesses so that it makes sense to provide applications for them.

The applications developed for the HP 250 cover those functions that are critical to the life, growth, and profitability of the business. We call these life-stream applications to characterize the favorable impact that comprehensive management information can have on asset management.

The design model for the life-stream applications is graphically represented by Fig. 1. This model shows that businesses can be divided into three broad categories: the distribution, manufacturing, and service industries. The general accounting function is placed in the center of the figure to indicate its applicability to all businesses. The distribution industry's life stream is to provide timely delivery of finished goods and efficient collection of customer payments. Finished goods inventory data and accounts receivable along with the attendant applications of order entry and sales analysis can contribute significantly to the operational efficiency and profitability of this type of business. The manufacturing industry's life stream is to define and have available the material required to build finished products and to determine where the costs are in the products. Applications that provide bill-of-material maintenance and methods to track inventory can minimize parts shortages, reduce obsolete inventory, and point the way to improved allocation of manufacturing resources. The service professional's life stream (e.g., law firms, accounting, and medical groups) is effective client billing and tracking of job costs. These industries have also shown interest in text processing.

An evaluation of the functions associated with the life streams of these three industries and the development re-
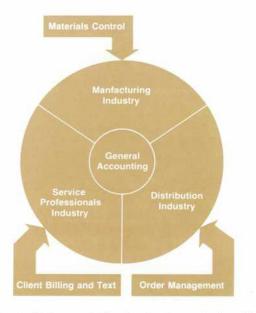
Fig. 1. *Design model for the development of applications programs for the HP 250 covers functions that are critical to the life, growth, and profitability of various businesses. Currently available software addresses the needs of the distribution and manufacturing industries.*

sources available resulted in two applications software products for the HP 250: OM/250 and MFG/250.* OM/250 is an order management system designed for the distribution function of distributors and manufacturers. The software modules in this system include accounts receivable (AR), order entry (OE), inventory control (IC), and sales analysis (SA). MFG/250 is a materials management system designed to control material and provide manufacturing bills of material and product costings for the discrete manufacturer. Fig. 2 lists the features of the OM/250 and MFG/250 applications software products.

Designing the Applications Data Base Organization

Once the general makeup of the applications packages was determined, the next step was to determine the primary features of each application and develop the data base organization. The design objective was to provide viable applications capabilities and performance using dual flexible discs while at the same time providing expandability to larger mass memories without software modification. This affected the HP 250 in two ways. First, an initial investigation into small business inventory sizes showed that it was doubtful that a viable application would fit on a dual ½-megabyte flexible disc system. The decision was made that the HP 250 would have a dual one-megabyte flexible disc system as a starting point. Second, it was decided that IMAGE/250 would have to have multivolume capability (see article, page 19). Thus IMAGE/250 was designed to determine automatically which mass memory device contains the data desired and to handle data base storing/restoring and unloading/loading under multivolume conditions.

Certain IMAGE/250 capabilities contributed greatly to

*These application software packages are designed for businesses in the U.S.A. and Canada. We expect to modify some modules for some European countries at a later date.

meeting the expandability requirement without having to make applications software changes. For instance, in moving to a larger mass memory system, the application must be able to handle larger amounts of data, and IMAGE/250 provides the DBINFO statement, which a program can use to determine the current size of a data set. Also, the number of data elements in a data set can be changed without changes to the program accessing that data set. A related technique involved designing the optional features supported by the various modules so that the data associated with these features is located in one data set. For the smaller mass memory configurations, the size of this data can be set to one, thereby allowing the feature to be available on the larger configurations but not on the smaller. The multivolume design of IMAGE/250 makes it unnecessary for all data sets in a data base to be on-line at the same time. This is extremely important for the OM/250 package, because it allows the data bases to be organized so that a viable interactive solution for order entry can be achieved in a dual flexible disc system.

Another benefit of IMAGE/250 was that the data base definition language made it mandatory to consider the data organization and data relationships before developing the applications software. Thus, as a consequence of defining the features that the applications modules would have, the data base requirements were also defined.

IMAGE/250 provides a consistent way of accessing the data in the data base from all applications software modules. However, the initial design of IMAGE/250 provided for data entry only in string form. The applications development effort showed that a more effective way of breaking string into its data elements was needed. The IN DATA SET statement was invented to fill this need (see article, page 19).

Designing the CRT/User Interface

The next problem that had to be solved was providing a consistent and easy-to-learn user interface. The guiding principles were:
1. It should be quick and easy for the user to get to any particular task.
2. The user interface should be as friendly (that is, informative and forgiving) and as devoid of computer command jargon as possible.
3. There should be standard methods and procedures for handling common operations regardless of which applications function or program is running.

To meet these objectives several decisions were made. For example, the modules provide for reports that can be printed on request, at the end of the next batch update, or on a periodic basis (daily, weekly by day of week, or monthly) after each update. HP 250 CRT soft keys were used extensively to provide easy communication with the user. Standards were established for the use of the soft keys. For example, the rightmost CRT key always provides a means for the user to leave the current applications environment. Pressing a soft key terminates the current data input; this is a powerful programming feature. The keys are used in menu trees for selecting a desired function and within functions to select the options available. Another standard that was established is a three-line message area directly above

Inventory (IC)

LIFO or FIFO or Weighted Average Costing Method
99 Product Classes
5 Discounts, 5 Markups, 5 Price Breaks Per Part
Calculated EOQ
Calculated Reorder Point
ABC Analysis
Cycle Counting
Multiple Warehouse Reporting
Multiple Company Reporting

Accounts Receivable (AR)

Balance Forward or Open Item Statements for Each Customer
Variable Aging into Five Periods
Sales Dollars Reported to General Ledger by Order Line Item
Full Truth in Lending Compliance
Over-the-Counter Cash Sales Journal
Credit Card Sales Journal
Past Due Letters
15 Cash Discounts
11 Customer Classes
Multiple Company Reporting

Order Entry (OE)

Post-Billing and Pre-Billing Operation
Picking Lists (by Warehouse or Order)
Acknowledgment Letters
Credit Limit and Post Due Checks at Entry of Order
Cash Sales/Credit Card Sales/Prepayments
Automatic Backordering with Customer Option
Automatic Backorder Filling
Stock Reservation
Automatic Price Extension
3 Tax Categories
Surcharge Capability by Line Item
2 Other User-Defined Optional Charges
Container Charge Credits

Sales Analysis (SA)

Salesperson Profitability Reporting
Customer Profitability Reporting
Product Profitability Reporting
Bar Graph Reporting
Quota Reporting
Commission Calculation by Product Class for Each Salesperson
Multiple Commission Calculation Methods
Regional Reporting
Split Sales Reporting
5-Year History on Salespersons, Customers, and Products
Multiple Company Reporting

Manufacturing (MFG/250)

Bill of Material Processor (30 Levels)
Where-Used Reporting
Product Cost Reporting (Implosions with "What-If" Costs Change Option)
Reorder Reports (5 Computation Methods)
Cycle Counting Subsystem
Stock Status Reports (Excess Inventory, Data, History Activity)
ABC Analysis (Simulated Also)
Stock Picking Lists (Including Simulation Preview)
Engineering Change Control
User-Defined Validation Tables

Fig. 2. *Features and capabilities of the OM/250 and MFG/250 applications software packages.*

the CRT soft keys for giving the user directions, receiving user responses, and displaying error, warning, or status messages.

The FORMS/250 features were used extensively to provide a consistent, easily developed and maintained method of handling user input and data display. The CURSOR statement for placing the cursor in the appropriate field regardless of its previous position on the CRT was of considerable help as the data input/output forms were being developed. Another feature that proved to be of help to the user in segmenting the display into functional areas is the HP 250 line drawing set. By using the lines to box in areas or to connect related data, the user can quickly see which data makes a functional group.

To tie the modules together into a consistent operating unit from the user's point of view, all modules are accessed through a single applications supervisor program called HP 250. It and its related programs constitute the control module. The control module acts as a supervisor; it maintains the system date and passwords and controls access to the applications software modules. It also maintains other system parameters such as the applications integration status array, hardware/software configuration status array, holiday/calendar, and so on. This information is kept in a control data file for use by all modules as required.

Designing the Memory and Mass Storage Organization

As the programs for the modules were developed, it became clear that many of the routines for handling various operations, such as the displaying of messages on the CRT, could be gathered together into a system utilities file for use by all programs. To provide rapid, selective access to these routines, the LOAD SUB capability was developed. Subprograms in the utilities file can now be selected and loaded programmatically with a significant increase in speed over the LINK statement, which was initially the only way of accessing the utilities file. Another fallout of this development is the DEL SUB capability. The applications programmer now can also programmatically manage the user memory and delete utilities as needed to make room for other applications programs.

As the number of programs increased, the problems of distributing the programs among the available mass memory devices and handling different mass memory configurations had to be solved. All files are divided into groups and each group is associated with a mass memory label, so that within limits, the groups can be migrated to different labels for different mass memory configurations. However, the tasks of spreading programs and data bases among multiple flexible discs and customizing data base capabilities are difficult to do manually, so utility programs were developed to solve these problems. For example, with OM/250, a program manages a data base on all applications program and nonprogram files, keeping track of their capacities, update date, target disc volume labels, and other information. This program is also used to generate masters for various mass storage configurations.

Another utility program is used to take advantage of the flexibility in IMAGE/250 to customize data base capacities according to user needs. The most difficult problem, however, is not setting the capacity but dynamically managing data bases and work files on multiple volumes. For example, more than ten data bases are used in the OM/250 applications software, and these are spread over eight flexible disc volumes. Changing the capacity of one data base on one volume can affect another related data base on another volume that may have limited space for expansion. In addition, the work file space requirement for FIND/SORT varies according to data set capacities. This utility program allows the software designer to set these data base and work file conditions and then computes to see whether the user capacities will fit in a given mass storage configuration. The

program can also optimize disc volume use by increasing capacities to take up the space not used according to the priority set by the user.

Designing a Flexible Data Reporting Scheme

In designing a flexible method for generating reports, the objective was to provide as much user latitude as possible. All application reports are produced by the REPORT WRITER system utility. REPORT WRITER provides basic tools such as totals, page breaks, and so on. In conjunction with the FIND/SORT capability, it provides considerable flexibility for report specification. A report setup program is used to select FIND and SORT criteria. The user can also specify which printer should be used for the report.

Insuring Data Integrity and a Quality Application Package

Throughout the development of the applications modules, an objective was to make sure that their operation was clearly evident to the user. It is very important to give the first-time user the feeling that the computer is just a tool and that people can only change the data base in a clear and concise way as opposed to the feeling that the computer somehow "gobbled up" data input and decided on its own what changes should be made to the data base. To achieve this goal all applications are transaction-oriented. Most data bases are updated in a batch mode. A transaction is generated whenever the data base needs to be updated, except for some special cases in OM/250 where data bases are updated directly. OM/250 software modules are highly integrated. Communication between modules is also carried out by transaction data files. Each module requires a daily update to the data base from these transactions. OM/250 does provide on-line capability for updating information that requires it. For example, OE knows how much is in inventory even though the inventory transactions have not been processed. This is done by checking not only the data base but also the transaction data file to come up with the latest information. A batch-updated transaction-oriented system offers advantages in data base integrity and ease of backup and recovery, since data bases are not constantly being updated. In addition, MFG/250 provides a means of going directly from the transaction building program to a transaction review/edit program with which the user can review, list, edit, or find transactions that have been entered already. An audit trail is produced whenever a data base is being modified either by direct input or by transaction. An audit trail provides a printed record of what has happened to the data base.

A daily backup and/or a backup before posting is required by the system. The backup/recovery function is monitored and executed by the control module. It involves making backup copies of all data bases and control data files. In addition, MFG/250 provides a special backup/recovery procedure that enables the user to recover without reentering transactions, using the backup transaction data base.

Quality assurance and reliability tests on the HP 250 applications software are done in three stages: designer's QA, integration test, and test sites. Designer's QA is done by the software module designers, who are expected to stress the module as much as possible. Integration test verifies the interactions among the software modules. This is crucial in the case of OM/250, where accounting information is passed between various modules. The test sites are actual user businesses. Checked there are installability, user documentation, and the user data conversion process.

Scott W. Y. Wang

An HP employee since 1972, Scott Wang received his BSEE degree in 1971 from Massachusetts Institute of Technology and his MSEE degree in 1972 from the University of Michigan. Scott designed the firmware for the HP 81 Desktop Calculator, contributed to the design of the NMOS II 16K ROM integrated circuit, and was project manager for the OM/250 and FIN/250 applications software. Born in Taipei, Taiwan, Scott left his country in 1962, visited Ethiopia for three years, and then settled in the United States. Scott is married, lives in Loveland, Colorado, and spends much of his leisure time skiing, hiking, taking photographs, and listening to music.

Loyd F. Nelson

Raised in Houston, Texas, Loyd Nelson is a 1967 BSEE and a 1969 MSEE graduate of Colorado State University. Loyd joined HP's Desktop Computer Division in 1973, where he was responsible for the system design of the 9825A Desktop Computer. He then transferred to Fort Collins Division, where he contributed to the data base system definition for the HP 250 Business Computer, designed the MFG/250 software package, and became project manager during its quality assurance phase. Before coming to HP Loyd designed computer-driven air traffic control radar displays. A resident of Loveland, Colorado, Loyd is married and the father of "three wonderful daughters." In his spare time he enjoys jogging, backpacking, and bicycling.

Capacitance and Conductance Deep-Level Transient Spectroscopy Using HP-IB Instruments and a Desktop Computer

A low-cost system of standard components is a useful engineering tool for checking new semiconductor processes and devices during their development.

by Leonard Forbes and Ulrich Kaempf

OBSERVING TRANSIENTS in the capacitance and current of the depletion region or space charge layer near PN junctions in semiconductor devices has been demonstrated to be a powerful technique for characterizing deep-level imperfection center concentrations, energy levels, thermal and optical emission rates, and thermal capture cross-sections.[1,2] The original techniques developed at the University of Illinois were subsequently automated at Bell Telephone Laboratories to analyze exponential capacitance transients in the so-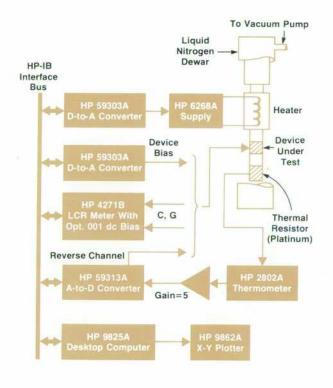called deep-level transient spectroscopy system, or DLTS system.[3] More recently, these techniques have been expanded to include the use of current transients on FET devices to measure deep-level imperfections in materials and situations where capacitance transient techniques are not feasible.[4]

This article describes the use of the HP 4271B LCR Meter,[5] which measures both capacitance and conductance, and the HP 9825A Desktop Computer for gating and control in an automated DLTS system for both conductance and capacitance transients. These components interface directly through the HP Interface Bus (HP-IB).* The 4271B is a very-high-resolution capacitance meter. It can resolve ± 0.001 pF and 0.01% capacitance changes, small enough to detect defect concentrations four orders of magnitude below the background concentration of usual shallow-level donor or acceptor impurities.

The 4271B can also measure conductance, which makes possible a conductance DLTS system. This is difficult or impossible with other DLTS systems. Conductance DLTS is particularly applicable to the determination of oxide or bulk trapping in very small FET devices, such as those employed in most modern circuits, which have junction or gate capacitances much too small for capacitance DLTS techniques.

Both capacitance and conductance DLTS techniques have been employed at HP Laboratories in the evaluation of bulk and oxide trapping in a wide range of technologies including standard MOSFET, silicon-on-sapphire, light-emitting diodes, and GaAs FETs. Despite the highly advanced state of technology, capacitance or conductance transients or changes, indicating various defects, have been found on all of these. An understanding of the nature and cause of these changes or transients is an important consideration in process design and in the reliability of integrated circuits and other devices.

DLTS System

Fig. 1 is a block diagram of the basic DLTS system. One HP 59313A Analog-to-Digital Converter is used to convert the analog temperature output from the HP 2802A Thermometer to an HP-IB format. The 59313A also has three extra A/D channels and a simple open-collector transistor in a reverse D/A channel. This reverse channel can be used to

Fig. 1. *Capacitance and Conductance deep-level transient spectroscopy (DLTS) system based on the HP 4271B LCR Meter and an HP Desktop Computer.*

*Hewlett-Packard's implementation of IEEE 488 and ANSI MC1.1.

drive an infrared emitter for minority carrier capacitance DLTS scans, or to control the gate voltage of FETs in conductance DLTS scans. The HP 4271B has an internal bias supply (Option 001) that can be controlled by the computer through the HP-IB.

The low-temperature portion of the system consists of an Air Products liquid nitrogen insert, or simple low-cost dewar. This has been modified by attaching a four-inch copper rod for thermal resistance, a heater spool, and a sample mount. Devices under test are mounted in either TO46 or 22-pin DIP packages. Temperature is measured using the platinum probe for the HP 2802A Thermometer. To increase temperature resolution, the analog output of the 2802A is run through a simple operational amplifier with a gain of five before going to the A/D converter.

Capacitance DLTS

The capacitance transient technique for the identification of imperfection centers in semiconductors[1] is based upon the observation of the thermal emission of carriers from the imperfection level in a reverse-biased PN junction. The centers can be filled either with majority carriers by first zero-biasing the junction or with both types of carriers by illumination with infrared radiation, which generates a large number of electron-hole pairs. The rate of emission of these carriers to the closest band edge is an exponential function of temperature;

$$e = A (T/300)^n \exp (-\Delta E/kT) \qquad (1)$$

where A is the emission coefficient or constant, T is the temperature in K, k is Boltzman's constant, and ΔE is the energy difference between the level and the bandgap edge. This emission of carriers results in a modulation of the space charge density in the reverse-biased junction and a transient in the high-frequency small-signal capacitance as illustrated in Fig. 2. The time constant of this change[1] is:

$$\tau = 1/(e). \qquad (2)$$

As originally applied, the DLTS technique[3] consisted of a dual-gating technique to analyze capacitance transients. Fig. 2 illustrates an exponential capacitance decay for which the capacitance value is measured at two time periods, t_1 and t_2. These times are determined by, in this case, the internal operation of the LCR meter and the WAIT statements in the computer program. The LCR meter integrates the capacitance value over the first 20 milliseconds following address and instruction to read the capacitance value. The remaining time, a worst-case 500 milliseconds, is used in encoding the data. For the exponential decay illustrated, a maximum difference ΔC in the capacitance values at times t_1 and t_2 is obtained if the time constant of this decay is

$$t_{max} = (t_2 - t_1)/\ln(t_2/t_1). \qquad (3)$$

For the other cases illustrated, where the time constant of the decay is either too slow or too fast, the difference in capacitance value between times t_1 and t_2 is smaller. The time constants of interest are exponential functions of temperature, so if the temperature is scanned, the difference will be maximized at the temperature where the time constant is t_{max}, and a peak will be produced in the output plot.

Capacitance DLTS has become a widely employed technique and excellent summaries have been written on the characteristics of implantation defects in silicon,[6] impurities in silicon,[7] hole traps in GaAs,[8] and electron traps in GaAs.[9] These review articles give plots of time constant or T^2 times time constant versus 1000/T, and can be employed to predict the temperature at which each of these impurities or defects will produce a peak in the DLTS scan.

As an illustration of this technique, Fig. 3 shows the result of a capacitance DLTS scan using the HP system on an arsenic-implant-damaged silicon N^+P diode, using zero bias as the initial condition. Two peaks are evident in this scan, one at $-92°C$ and one at $-117°C$. The time constant that produces a peak for this case is around 0.35 second. By using this time constant and the appropriate value of 1000/T, these levels can be plotted as two points on Sah's curve[6] and shown to correspond to the 0.40 and 0.48 eV levels. The levels at higher temperature in Fig. 3, and consequently deeper energy, are so numerous in this case of implantation damage that has not been annealed that they run together and cannot be resolved using the DLTS technique. A subsequent thermal anneal, however, will result in a DLTS scan with no structure on the highest gain scale, showing that
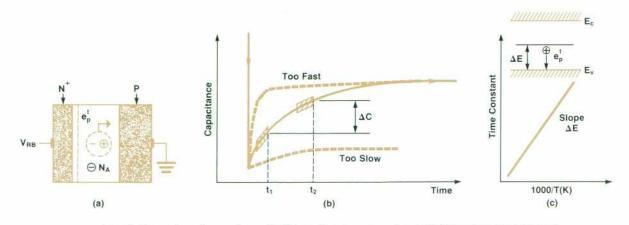

Fig. 2. *Principles of capacitance DLTS. (a) Emission of carriers in N^+P junction is stimulated by applying bias or infrared radiation. (b) Capacitance transient results when junction is reverse biased. LCR meter measures junction capacitance at two gated time periods. Maximum capacitance change ΔC occurs at a temperature (i.e., time constant) that is characteristic of the type of defect. (c) Temperature dependence of capacitance time constant.*
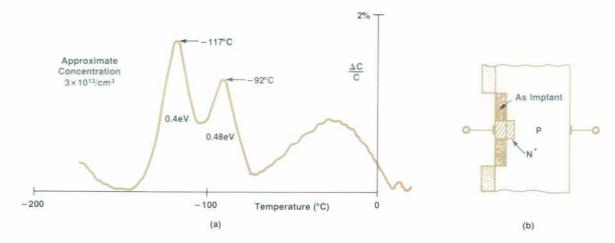
Fig. 3. *Capacitance DLTS scan on arsenic-implant-damaged diode. (a) Measured data has peaks at two temperatures. Using published data, these can be related to energy levels and particular defects. (b) Device cross-section.*

the concentration of any and all of these defects has been reduced to less than $5 \times 10^{11}/cm^3$.

Conductance DLTS

While capacitance DLTS is a powerful technique, it is limited in the case of small FET devices, which have only very small capacitances, and on high-resistivity substrates, which limit the ability to measure and resolve capacitance values. In these cases what is referred to here as conductance DLTS constitutes a technique that can be used not only for bulk trapping[4] but also for oxide or insulator trapping. The concepts involved are much the same as in the case of capacitance DLTS and will be illustrated here by the observation of electron trapping at the chromium level in the gate junction of GaAs FETs.

Fig. 4 illustrates the electron emission in a GaAs FET following application of operating gate and drain voltages. In this case modulation of the space charge in the gate junction results in an increase in conductance of the FET. The LCR meter can be used directly to measure small-signal drain conductance. The time constant associated with this conductance change is again simply the reciprocal of the emission rate and results in a conductance DLTS peak as shown in Fig. 4 at about $+32°C$. This can be shown to correspond to the very recently measured emission rate and characteristics of chromium in GaAs.[10,11]

In the conductance DLTS technique presented here a large-signal pulsed bias is applied and the emission of carriers from impurity centers in a depletion region is used for indirect modulation of the FET's conductance.[4] Since the centers being observed are in a depletion region, the technique is very similar to capacitance DLTS and not at all similar to small-signal conductance techniques.[12]

Conductance DLTS has also been found to be most useful in the characterization of oxide trapping. In this case carriers are injected into the oxide gate insulator in FETs and the subsequent transients in conductance are observed as these carriers are re-emitted by the oxide traps.

Summary

The laboratory-assembled system described here based on the HP 4271B and 9825A is an engineering tool that can be put together quickly using standard components at a nominal cost and used to check new processes and devices during their development. Defect levels are obtained using previously published data on impurities and defects in semiconductors. Capacitance DLTS can resolve defect concentrations as low as $5 \times 10^{11}/cm^3$ and conductance DLTS
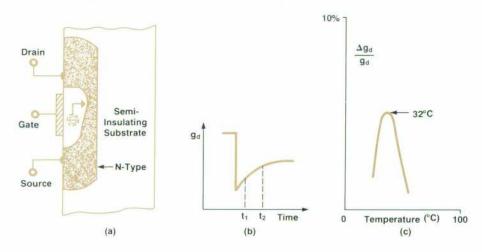

Fig. 4. *Conductance DLTS example. (a) GaAs FET cross-section, showing electron emission from chromium center. (b) Transient in measured drain conductance following application of operating gate and drain voltages. (c) DLTS scan.*

Ulrich Kaempf

Ulrich Kaempf is a project engineer with HP Laboratories, designing automated test and processing equipment for integrated circuit research. A native of Switzerland, he received his diploma in electrical engineering from the State School of Engineering at Burgdorf in 1954. In 1969, following ten years as engineering section leader and European technical manager for a major semiconductor firm, he joined HP as a product manager for automatic test systems. He assumed his present position in 1974. Ulrich is married, has three teenage children, and lives in Los Altos, California. He is active in the American Youth Soccer Organization and in competitive long-distance running. In 1978 he set regional Amateur Athletic Union records in 15-km and marathon races.

Leonard Forbes

Len Forbes is an associate professor of electrical engineering at the University of California at Davis and a consultant to HP. A native of the province of Alberta, Canada, he received his BSc degree in engineering physics in 1962 from the University of Alberta at Edmonton and his MS and PhD degrees in electrical engineering in 1963 and 1970 from the University of Illinois at Urbana. He has held faculty positions at Howard University and the University of Arkansas and was with IBM Corporation for two years. A member of IEEE and the American Association of University Professors, he is author or co-author of nearly 50 papers, articles, and reports. His research at UC is in GaAs materials, devices, and circuits.

resolves changes of a few hundred trapped electrons. Application during the development of new semiconductor processes and devices to detect such trapping problems as early as possible might well avoid considerable delays and expense in the overall development cycle.

Acknowledgments

References

1. C.T. Sah, L. Forbes, L.L. Rosier, and A.F. Tasch, Solid-State Electronics, Vol. 13, 759 (1970).
2. C.T. Sah, Solid-State Electronics, Vol. 19, 975 (1976).
3. D.V. Lang, Journal of Applied Physics, Vol. 45(7), 3023 (1974).
4. L. Forbes, R. Brown, M. Shiekholeslam, and K.W. Current, Solid-State Electronics, to be published.
5. K. Maeda, "An Automatic Precision 1-MHz Digital LCR Meter," Hewlett-Packard Journal, March 1974.
6. C.T. Sah, Proceedings 3rd International Conference on Silicon, Semiconductor Silicon, Vol. 77-2, pp 868-893, The Electrochemical Society, 1977.
7. J.W. Chen and A.G. Milnes, Solid-State Electronics, to be published.
8. G.M. Martin, A. Mitonneau and A. Mircea, Electronics Letters, Vol. 13(7), 191 (1977).
9. A. Mitonneau, G.M. Martin and A. Mircea, Electronics Letters, Vol. 13(22), 667 (1977).
10. P.K. Vasudev and R.H. Bube, Solid-State Electronics, Vol. 21(9), 1095 (1978).
11. A. Damestani, C.D. Chang and L. Forbes, Solid-State Electronics, to be published.
12. S. Sato and C.T. Sah, Journal of Applied Physics, Vol. 41, 4175 (1970).

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