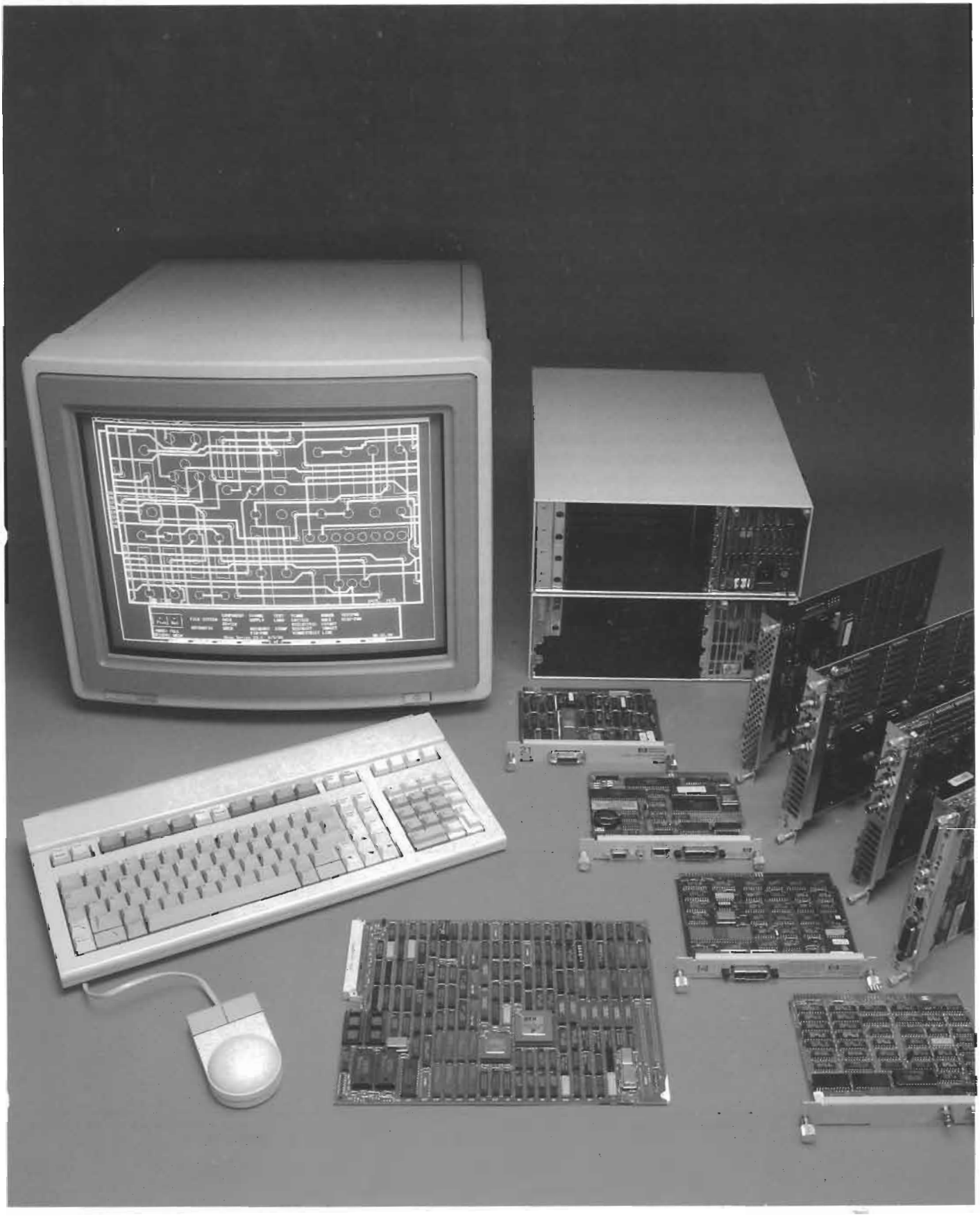
HEWLETT-PACKARD
JOURNAL

SEPTEMBER 1986

Computer Museum



HEWLETT-PACKARD
JOURNAL

September 1986 Volume 37 • Number 9

Articles

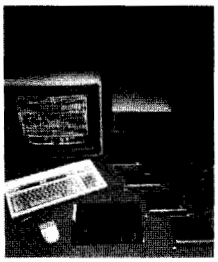
Departments

Editor, Richard P. Dolan • Associate Editor, Business Manager, Kenneth A. Shaw • Assistant Editor, Nancy R. Teater • Art Director, Photographer, Arvid A. Danielson • Support Supervisor, Susan E. Wright
Illustrator, Nancy Contreras • Administrative Services, Typography, Anne S. LoPresti • European Production Supervisor, Michael Zandwijken

© Hewlett-Packard Company 1986 Printed in Holland

Computer Museum

In this Issue



Engineering workstations are computers that are found on engineers' desks, but personal computers they are not, except for the superficial visual resemblance. A case in point is this month's cover subject, the HP 9000 Series 300 family of modular workstations. As the cover photo suggests, the Series 300 offers a range of options that would bewilder the typical PC buyer. Designing the Series 300 Computers and the production process so that not only does each customer get a correctly configured computer, but also any option can be installed later in the field, was a significant engineering challenge that's discussed in the article on page 4. The same article tells how the Series 300 came to its present definition, starting out as a proposed low-cost member of the Series 200 family and evolving into a modular replacement for the entire Series 200 that offers performance far beyond any Series 200 Computer. The performance of the Series 300 Computers begins with the Motorola 68010 and 68020 microprocessors and the 68881 floating-point coprocessor for the 68020. Model 310, the lower-performance member of the family, uses the 68010, which has 16-bit address and data buses and does 32-bit operations. The design of the Model 310 processor board is the subject of the article on page 9. Model 320, the higher-performance Series 300 family member, uses the 68020 and the 68881, which are full 32-bit processors. Model 320's design is described in the article on page 12. The issue of porting Series 200 software to the Series 300 was a major design concern that is addressed in the article on page 22. Powerful graphics capabilities, which are mandatory for the CAD/CAE applications that are an engineering workstation's daily fare (the display in the cover photo shows a printed circuit board design application), are provided by a bit-mapped display based on two custom VLSI chips. In the article on page 17, the chips' designers tell us how they dealt with requirements to support both monochrome and color monitors, each with either medium or high resolution, while maintaining compatibility with the Series 200, which doesn't have a bit-mapped display.

Electronic mail, along with word processing, spreadsheets, and graphics, is one of the basic services expected of an office automation system. Here at the *Hewlett-Packard Journal*, we began to use HP's own electronic mail product, HP DeskManager, early in 1983. Today, we'd be hard-pressed to get along without it. It has cut the time it takes for written communication with Europe or Japan from about a week to overnight. It is the way we prefer to receive article manuscripts, because we can copy them directly to HP's word processing product, HP Word, for editing, and then either mail them right back to the authors for review or dump them to our typesetting system, saving the multiple retypings that we used to have to do. HP Desk (we prefer the shorter name) is integrated with word processing, spreadsheet, and graphics software so that files created using those products are easily mailed electronically to any of the more than 50,000 HP Desk users worldwide. According to published reports, that number of users makes HP's network comparable in size to the larger commercial electronic mail networks. The paper on pages 30 to 48 is a comprehensive discussion of the lessons learned in implementing the HP network successfully. Topics discussed include strategy, specific tactical advice, and potential pitfalls. The paper should be valuable to any organization implementing, planning, or thinking about an electronic mail system.

-R. P. Dolan

What's Ahead

Computer networking will be the theme of the October issue. Topics will include HP AdvanceNet, which is Hewlett-Packard's overall networking strategy, and various networking products and services for HP 1000, HP 3000, and HP 9000 Computers.

The HP Journal encourages technical discussion of the topics presented in recent articles and will publish letters expected to be of interest to our readers. Letters must be brief and are subject to editing. Letters should be addressed to: Editor, Hewlett-Packard Journal, 3200 Hillview Avenue, Palo Alto, CA 94304, U.S.A.

Advanced Modular Engineering Workstations

This workstation system allows the user to choose the processor, display system, memory, interface cards, peripherals, and operating system most appropriate for the application.

by Gilbert I. Sandberg, Daryl E. Knoblock, John C. Keith, Michael K. Bowen, and Ronald P. Dean

THE HP 9000 SERIES 300 is a modular, high-performance, technical workstation family (Fig. 1) that can be configured to meet the needs of a wide range of technical applications. An engineer or scientist can choose from two SPUs, six displays, six I/O slots, and a wide range of input devices to meet exact needs, and can later upgrade the workstation in any of the options with only a few minutes of assembly effort. This article discusses the impact of such a large choice of options on the Series 300's development.

Development History

The development of a new low-cost member of the HP 9000 Series 200 Computer family was first proposed in 1981. Using a high degree of both physical and logical integration, this machine was to sell for only $2000 (U.S.). It was proposed that this computer consist of only two printed circuit boards enclosed within a video monitor: a monitor electronics board and a single-board SPU (system processing unit) tapping its power from the monitor board. Adding a keyboard and a disc drive would produce a complete system capable of running the HP-UX operating system, HP's enhanced version of AT&T Bell Laboratories' UNIX™ operating system. As the required logic was analyzed, two areas were singled out for proprietary LSI development: the address translation subsystem needed to run HP-UX, and the video display subsystem.

At that time, there was a growing awareness of the system advantages of a new type of display subsystem called a bit-mapped display. This type of display subsystem holds text character images in the same memory used for graphics pixels, rather than generating them on the fly from ASCII characters stored in a separate memory. Although there is far more flexibility in the generation and use of character fonts, much additional hardware is needed to maintain the performance of character placement and scrolling. Although this type of display subsystem seems inconsistent with a low-cost product, it is far more easily integrated into an LSI chip than its predecessors, because the memory for storing ASCII character values and the ROM needed to convert them to graphics images on the fly can be elimi-

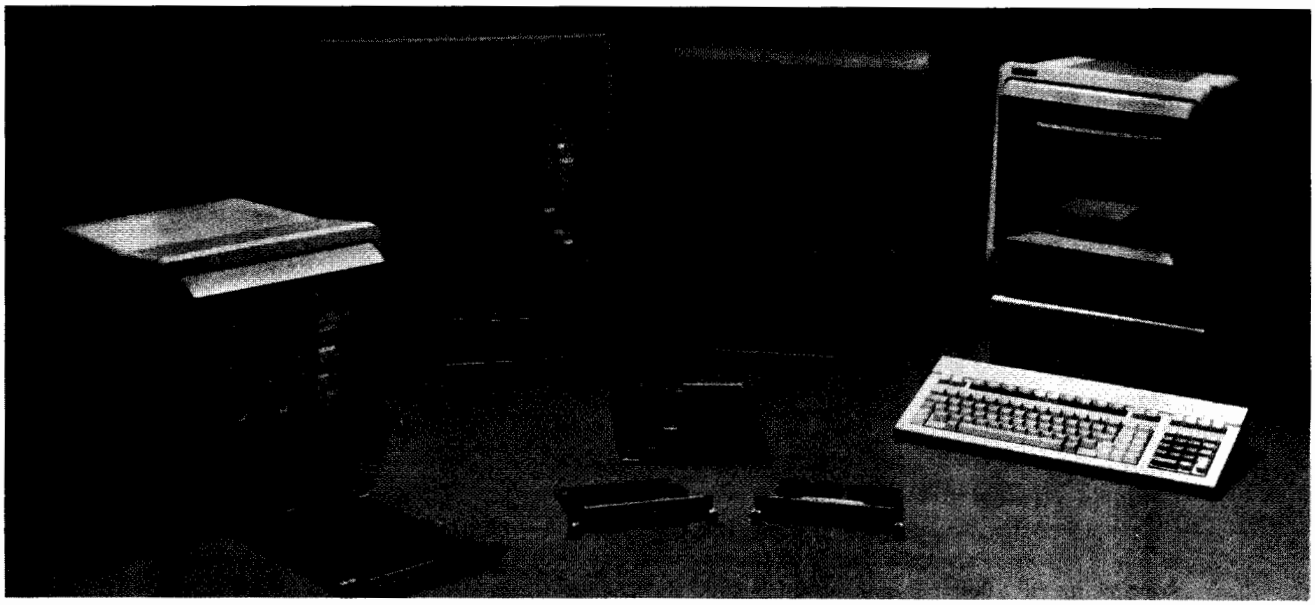


Fig. 1. *The HP 9000 Series 300 is a modular family of high-performance technical computer systems designed for instrument control and CAD/CAE applications. A variety of processors, graphics displays, I/O options, mass storage devices, and other peripherals are available to tailor a system for a particular application.*

Computer Museum

nated. Therefore, it was decided to develop such a chip, based on one that had already been developed at HP's plant in Corvallis, Oregon, but capable of higher resolution, color graphics, and much faster character manipulation.

The address translation subsystem (called a memory management unit, or MMU) also began development at this time, and these two LSI efforts essentially drove the schedule for the first two years. But time has a way of changing the factors that define a project, and projects that contain large LSI efforts are particularly vulnerable because of their long duration. Two such factors had a dramatic effect on the Series 300 definition: advances in monitor technology and the development of a high-performance 32-bit microprocessor.

By 1984, Hewlett-Packard was developing a standard, uniform packaging strategy, known as the ITF family, for use within most of its computer systems. The main benefit of this concept is the ability to leverage modular components designed and manufactured at other HP facilities and still produce a system that looks as if it were designed by a single team (which in fact it was, although the team was spread over a large geographic area). This family includes low-cost color and monochrome monitors that match the needs of the Series 300 product, and because of their high production volumes, result in a lower total system cost than the original integrated product definition. As a result, the Series 300 was redefined to be a modular rather than an integrated product.

With this change to a modular definition, striking similarities showed up between the Series 300 and a parallel effort to define a new high-performance member of the Series 200 family based on Motorola's proposed 68020 microprocessor. In May of 1984 it was proposed to merge these two projects into a single project which offered in one product a choice of two processor boards: a single-board computer capable of running the UNIX operating system based on the Series 300 LSI technology and Motorola's 68010 processor (see article, page 9), and a high-performance three-board computer based on the full 32-bit 68020 processor (see article, page 12). This product proposal, which included a choice of four displays and I/O expansion from zero to twelve I/O slots, was capable of being configured to match the characteristics of every member of the existing Series 200 family.

With a single product, HP could consolidate its manufacturing of technical computers, offer better price/performance ratios in each market segment served by each member of the Series 200, and allow customers to tune a machine to their exact needs. In addition, marketing techniques and design features were proposed to allow customers to upgrade their processor, display, or I/O subsystems at any time after their original purchase of a Series 300 model.

Achieving Completeness

As the concept for the HP 9000 Series 300 Computer family evolved into the goal of completely replacing the earlier Series 200 family of computers, the task of providing a complete solution grew rapidly. HP's Fort Collins Systems Division (FSD) had completed many major successful programs in the past,[1,2] but never before had the division worked on a program to replace a complete family of computers. There were five different models in the Series 200 family that had to be replaced at one introduction event. The new family of computers had to be able to cover the complete range of functionality that the previous family provided, and, ideally, do it at a lower cost and with improved performance. At the same time, our market direction was being expanded to include design automation (CAE/CAD) as well as the traditional customers served by the Series 200. This placed additional requirements on providing a complete solution.

The most noticeable requirement imposed by the CAE market is in the display offering. Although the high performance needed for CAE applications is available from the 68020 processor, the medium-resolution color and monochrome displays selected at that time to replace existing Series 200 capability did not provide enough resolution for the detailed graphics images required for CAE displays. Therefore, high-resolution color and monochrome displays were added to the family definition.

Since the medium-resolution monochrome and color displays were developed at HP's Roseville Terminals Division, a major area of responsibility and needless diversion of resources was avoided at FSD. Only one part-time engineer was needed to act as a liaison with the monitor design group to ensure that these monitors would interface satisfactorily with the Series 300.

FSD already had displays that met the high-resolution requirements of the CAE marketplace. All that was required to use these displays for the Series 300 was to design the appropriate hardware driver boards and the custom integrated circuits (see article on page 17) that were necessary to reduce the physical size of these boards so they could fit in the product package.

The keyboard design was done in a similar fashion, this time working with the personal computer group in HP's Singapore facility. Keyboard design and manufacturing activities have been centralized in Singapore to maximize volume and make efficient use of resources for tooling and manufacturing robots. FSD uses the ITF keyboard from Singapore[3] on the earlier Series 200 Model 217 Computer. This same keyboard is used on the Series 300 without changes. Again, all that was required in the form of resources from FSD was someone to act as a liaison with Singapore, since although the keyboard was already in production, design changes continue to be made in the keyboard to improve reliability and customer satisfaction.

The main system processor unit needed to continue the concept of extensibility and flexibility. To achieve this, two CPU boards were designed to fit into the same system slot. With the 68010 single-board computer described in the article on page 9, many of the configurations of the earlier Series 200 family could be replaced with a unit that had higher performance and lower cost. However, by substituting the 68020 processor board with high-speed cache memory and a full 32-bit-wide memory bus described in the article on page 12, twice the performance is provided for complex applications.

The ability to incorporate extensive I/O capability was important for the measurement automation applications that are served by the Series 200, yet posed significant cost

burdens at the low end. The maximum number of I/O slots of the SPU package had to overlap the number in the earlier Series 200 family, but also needed to match the low cost of the least-expensive Series 200 model. Since the 68010 processor board has most of the I/O capability required by many customers, this problem was solved by making an I/O card cage that is removable from the basic system package. Then, for those applications that require the high-performance 68020 SPU, which has no I/O capability and also frequently requires large amounts of memory, a passive I/O expander was designed. This expander fits on top of the 68020 SPU, and has mechanical and electrical connections that can be made by the customer with no special tools.

ID Module

Although only the major system needs in providing a complete family have been discussed, there were several other needs that, while smaller in scope and complexity, are just as important in terms of completing the functional replacement of the earlier Series 200 family. One such need is the ability of the Series 200 processors to return, upon command by the software operating system, a unique serial number. This feature was provided so that application programs that had the capability of checking this variable could be made to restrict their operation to only those machines that had had their serial numbers encoded in that software. This is important to prevent unauthorized use of application programs that have been supplied by third-party or independent software vendors.

As the development of the two processor boards proceeded, it became obvious that with the complexity and the physical size limitations of those two boards it would be impossible to include this self-identification feature on the Series 300 processor boards. This created a problem, since there are no other boards that always reside in the mainframe that could be called upon to host this function. What evolved to solve this problem is a small package known as the ID Module, which interfaces with the machine via the same mechanism as the keyboard. This ID Module also has the capability to return a unique number (actually its own serial number) to the software operating system for the same use as mentioned earlier. Having this capability in a separate package from the SPU, rather than integrated with it, provides two benefits. First, being portable, the ID Module can be moved from one machine to another. Thus a customer who legitimately has authorization to use a certain software package is not limited to using it on just one machine. If there is need to travel to an off-site location and run this same software on another Series 300 Computer, all the user needs to carry is the ID Module and that software. Second, if a machine were to fail, a user can, if more than one Series 300 Computer exists at the user's site, simply move to another computer and not be prevented from using authorized software.

Series 200 Display Compatibility

Compatibility with the Series 200 method of driving the CRT display was another requirement. The Series 200 uses separate memory planes for the alphanumeric and the graphics displays, while the Series 300 uses a single bit-mapped memory plane for both. Running software programs on the Series 300 that are designed for the Series 200 and make use of both the graphics and alphanumeric planes simultaneously could cause serious problems with the resulting display on the Series 300 screen. To resolve this compatibility issue, the same boards that are used in the Series 200 to provide the alphanumeric and graphics planes of memory are supported as an option to the Series 300, but with the addition of a software-controllable switch that allows either the bit-mapped driver or the two separate planes to be connected to the CRT display. The Series 300 software operating systems also incorporate features to make application programs written for the Series 200 compatible with the Series 300 (see article on page 22).

Design Verification

The Series 300 had to succeed in four areas: marketplace contributions, quality, schedule and factory cost. It was decided to test the Series 300's definition early in the marketplace to ensure that its contributions were adequate. Focus panels, step studies, and completeness surveys were done both inside and outside HP. The product was reviewed in detail with key customers.

To ensure early availability of software and applications on the Series 300, fifty hardware prototypes were developed and distributed inside and outside of Hewlett-Packard, worldwide. These units gave valuable feedback to the designers and allowed software development and application porting to Series 300 to begin months before introduction.

Product Design

The Series 300 represents the second generation of 325-mm-wide computers in the HP 9000 family. The primary objective of the product design was to make a Series 300 system easy to expand, manufacture, and service.

Expandability is the ability of the computer to support numerous configurations. At introduction there were two processor boards and four display boards. Each board has its own segment of a rear panel and conforms to the same dimensions. Captive screws that can be turned by hand are used to secure all of the cover plates.

The Series 300 uses the Series 200 DIO cards. The four-slot card cage can be deleted and added later. An eight-slot expander can also be added to the four slots in the computer. These expansions require the removal of the top cover held on by two captive screws. Installing the four-slot card cage requires a screwdriver for two flathead screws, but installing the expander does not require a tool. Fig. 2 shows a typical step-by-step expansion of the Series 300.

The box goes together very quickly, as illustrated in the exploded view of the product, Fig. 3. The fans and motherboard are fastened to the fan plate, which then is assembled to the chassis. Front and rear feet are attached to the chassis and the power supply slides in. The DIO card cage and the printed circuit boards are installed and the unit is tested. To close the box, the top cover and power supply door are added and the front panel is snapped on. The unit is then packaged and shipped to the customer.

The total assembly time is 30 to 45 minutes, depending on the configuration. Assembling the I/O expander takes only 30 minutes and requires almost no test time. In con-

Computer Museum

trast, a configuration with a 68020 processor board and a high-resolution color monitor board requires 45 minutes (15 minutes of test time). This assumes that the printed circuit boards have already been built and tested. The printed circuit boards are aged at the board level, not at the instrument level.

Assembly Process

The assembly process uses computerized process control. A serial number plate and bar code sticker are printed for the unit. A monitor displays what option is to be built. The computer only lets shippable orders onto the production line. A shippable order must have all peripherals available for coordinated shipments. The computer determines

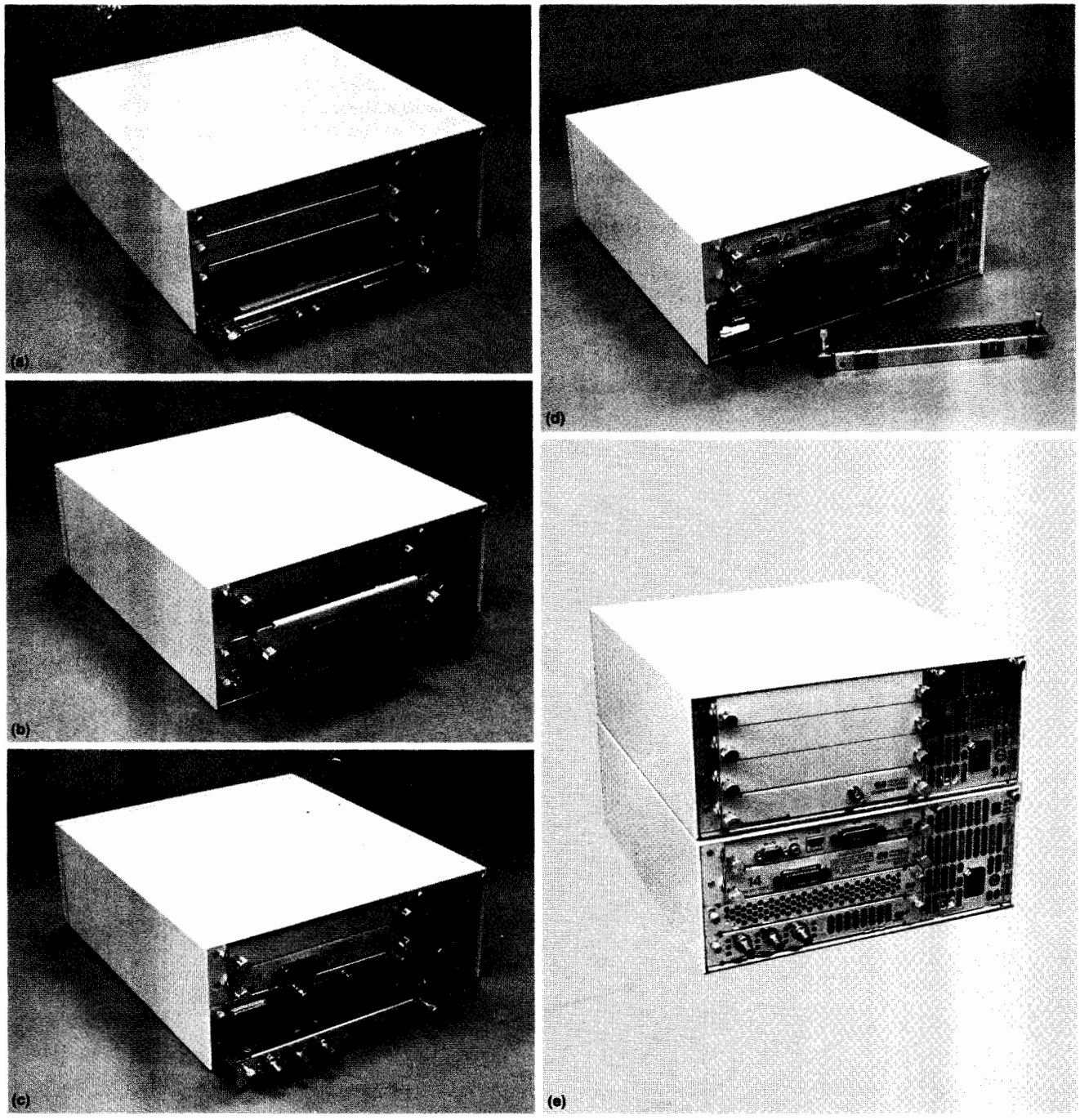


Fig. 2. *(a) The minimum Series 300 configuration is the single-board computer. (b) Expand the system by adding the DIO card cage. This allows the addition of more memory, I/O cards, or accessory cards. (c) Upgrade with any of four optional graphics interfaces—high- or medium-resolution, color or monochrome. (d) Upgrade to the 68020 processor. This requires the DIO card cage for the HP-HIL board and memory. (e) Expand again with the DIO expander for a total of twelve DIO slots.*

the sequence of configurations to be built to maximize throughput. Since testing is such a large proportion of the build time, a large backup would occur at the test station if units with long test times were built in sequence.

After the basic box is assembled, it is configured with the appropriate printed circuit boards. The operator reads the bar code into the computer and the display shows what boards to install. At the test station the test operator attaches the necessary cables, then reads in the bar code. This starts the test. The operator must verify the operation of the fans, the power-on indicator, the audio signal, and the video output.

The same operator then performs the final package closure. The bar code is read, and the computer prints the packing label (which also has a bar code on it). The top cover, front panel, door, and labels are installed. Finally, high-potential testing is completed.

At packaging, the bar code is read and the computer displays which cables need to be shipped with the unit. The bar coded cable kits are verified as they are packaged with the computer. The unit is then ready to go to the shipping department, where the bar coded shipping label helps generate a pick list of other products to include with the unit.

Package Design

The Series 300 computer and expander use the same design. Both boxes share the same tooling and parts, and were on the same development schedule. Since the DIO card cage is exactly the same height as two system boards, the expander just substitutes an additional four DIO slots for the two system boards. The same brackets and card guides are used twice in the expander. One innovative technique used to conserve space is that the connection to the backplanes is made on the back side of the board. Connectors are press-fitted in the spaces between the pins of

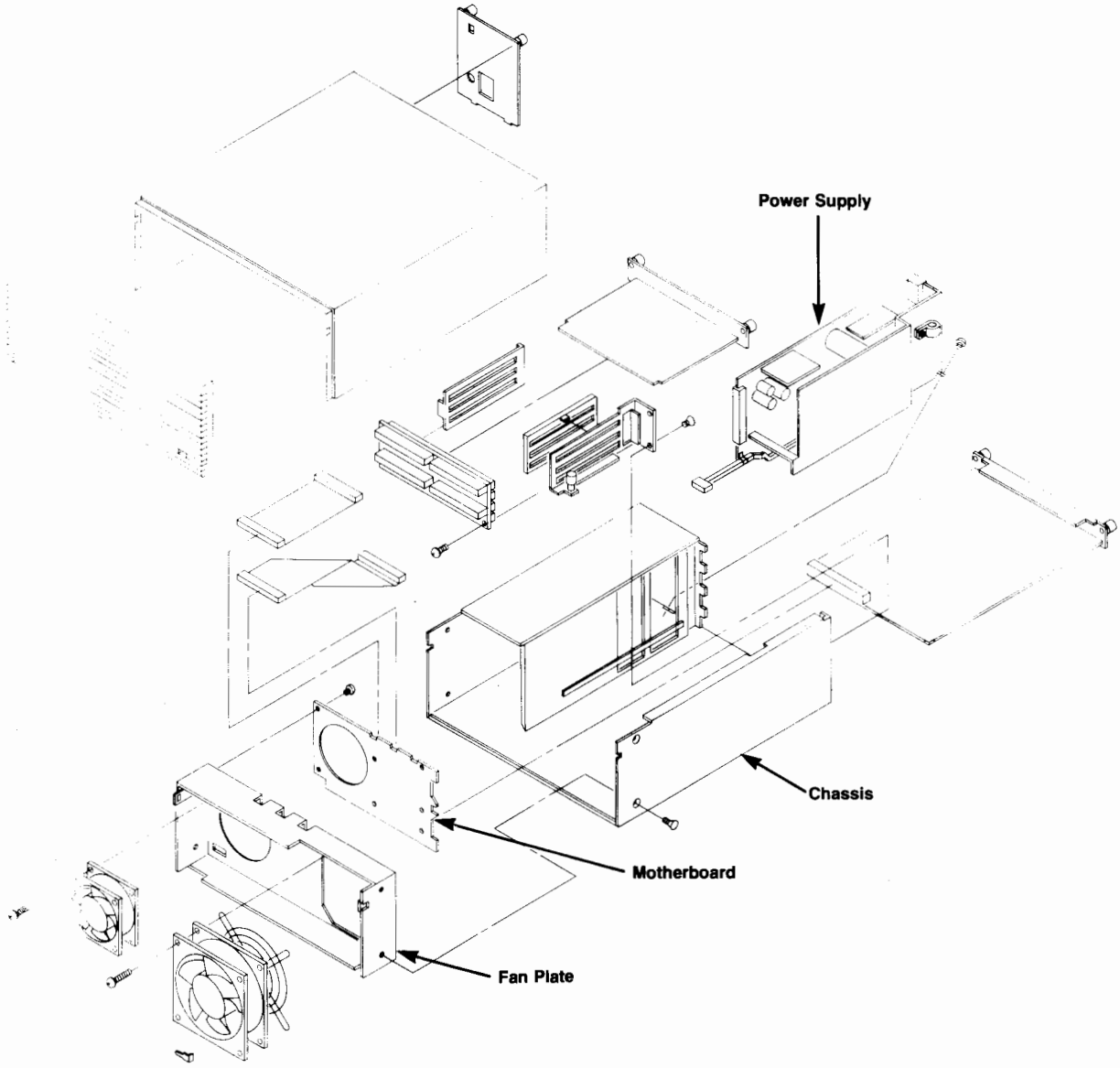


Fig. 3. *Exploded view of a Series 300 Computer.*

Computer Museum

adjacent DIO card connectors.

The expander top cover and chassis have an added hole to bus signals from the Series 300 computer to the expander. The expander cover gets swapped with the computer during the installation and sliding clips hold the boxes together. The motherboards use the same board blank, but to reduce cost, the unused connectors are not loaded. The expander motherboard also differs because there is a power cable to the backplane. The rest of the sheet metal, all plastic parts, and the power supply are identical.

Other measures of manufacturability are the type and quantity of screws used in the unit. The assembly of the basic box requires four flathead screws to hold the sheet metal pieces together, seven panhead screws to hold the motherboard, and one screw to hold the power supply in place. Each fan requires two screws, and the DIO card cage, if present, requires six. The front panel snaps onto some spring clips and the interior side of the DIO card cage uses only tabs, slots, and hooks for attachment.

The sheet metal enclosure reduces RFI (radio-frequency interference) without the use of extensive gasketing. Each system board has two custom clips that contact the cover plate on the board below, or connect directly to the chassis. The power supply contacts some different clips on top and bottom. Those clips are the only items added specifically to suppress RFI. The extensive use of overlaps in the sheet metal parts allows this minimal addition of special parts. This avoids the addition of gasket material, which is often a concern to the manufacturing organization, being time-consuming and at times having a high scrap rate.

The Series 300 computer and expander are easy to manufacture and consequently they are easy to service. All of the system boards and DIO cards slide in and out from the rear of the unit. The power supply also slides out from the rear and there are no cables to disconnect. The fans and the LED power-on indicator can be replaced from the front of the box, even if the unit is underneath a stack of other boxes. The front panel is easily removed, giving the service person access to the cables and all of the necessary screws. However, the motherboard is very difficult to remove, requiring that the box be completely disassembled. Fortunately, the motherboard has only connectors, resistors, and diodes which have very low failure rates.

Acknowledgments

Doug Buhler, Series 300 Program coordinator, played a key role in that the program held schedule the last year of development. Danny Darr of the Fort Collins Engineering Operation provided valuable CAD/CAE inputs and Sandy Chumbley of the Colorado Networks Operation provided networking guidance.

References

1. HP 9845 Computer, complete issue, *Hewlett-Packard Journal*, Vol. 31, no. 12, December 1980.
2. Series 500 Computers, complete issue, *Hewlett-Packard Journal*, Vol. 35, no. 5, May 1984.
3. L. Dunn and M.R. Perkins, "A Standard Keyboard Family for HP Computer Products," *Hewlett-Packard Journal*, Vol. 35, no. 8, August 1984.

Modular Computer Low-End Processor Board Design

by Martin L. Speer and Nicholas P. Mati

THE HEART OF THE HP 9000 Model 310 system processing unit (SPU) is the processor board. With the exception of the power supply, no other major electrical subsystems need exist within the Model 310 SPU box. By adding a medium-resolution monochrome video monitor, an HP-HIL keyboard, and mass storage, a complete and useful workstation capable of running Pascal, BASIC, or the HP-UX operating system can be constructed.

The Model 310 processor board is a complete single-board computer with the following features:

- 10-MHz MC68010 microprocessor
- Up to 1M bytes of high-speed RAM
- Model-320-compatible memory management unit (MMU)
- HP-IB (IEEE 488/IEC 625) interface
- RS-232-C/V.24 interface
- HP-HIL keyboard interface
- Programmable sound generator (beeper)
- Battery-backed real-time clock
- Bit-mapped monochrome display electronics with 1024-dot-by-400-line resolution
- Programmable timer module (used by HP-UX operating system)
- Up to 128K bytes of boot ROM (64K bytes are currently being used).

Putting all these features on one board makes the Model 310 system more manufacturable, reduces the cost, and allows increased performance for both the RAM and the memory management unit. Yet, a system designed around the Model 310 processor board is still expandable by means of the Model 310's second system board slot, a four-slot DIO backplane, and an eight-slot DIO expander. (DIO is an asynchronous bus based on the 8-MHz MC68000 micro-

processor.)

Several physical design challenges were posed by the Model 310 processor board. First, all the features listed above had to fit within 93.4 square inches and meet component height restrictions that ranged from 0.325 inch to 0.775 inch. Second, restrictions were placed on component location because of thermal design constraints imposed by the system box and a need to minimize trace lengths for critical signals. Improved timing margins, reduced cross talk, reduced trace capacitance, and reduced EMI (electromagnetic interference) were the key benefits of complying with the component location and trace length restrictions.

Increasing the performance of the Model 310 processor board had its share of design challenges when coupled with board area limitations and cost goals. The main performance contributions were made by adding a 10-MHz 68010 processor, a local RAM bus, and hardware to assist HP-IB parallel polls. The decision to use the 68010 microprocessor was made because earlier 8-MHz products were viewed as being too slow. To take advantage of the increased clock frequency of the 68010, a fast local RAM bus and RAM controller were designed. The local RAM bus supports faster memory accesses than are possible on DIO, whose timing is based on 8-MHz operation. RAM accesses to memory on the Model 310 board add no wait states while RAM accesses to memory on the DIO bus require two wait states.

Hardware to assist HP-IB parallel polls was added to improve the performance of the HP-UX operating system when the internal HP-IB interface is used as the disc interface. Without this additional hardware, the HP-UX system conducts software parallel polls periodically. Once a parallel poll is detected, the operating system must synchronize the pending disc transfer with the disc drive. This is a time-consuming task necessitated by the disc drive's requiring data within a short time after the parallel poll response. The hardware designed for the Model 310 processor board conducts a parallel poll and generates an interrupt when an HP-IB device responds. Interrupts can be serviced within the response time required by disc drives and the resynchronization software cycle becomes unnecessary. A performance increase for disc accesses of approximately 50% is observed because of this circuitry.

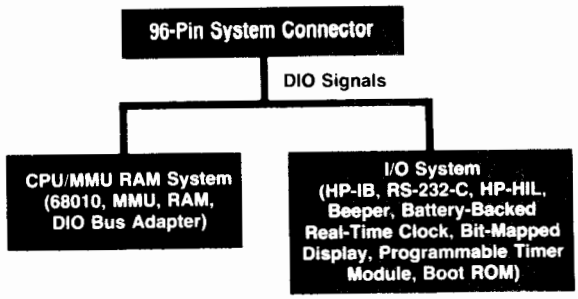


Fig. 1. *Basic system partitioning of the Model 310 processor board.*

A major design challenge was that of maintaining software compatibility with earlier HP 9000 Series 200 systems. With two exceptions, the final design of the Model 310 SPU board maintains a high degree of hardware architectural compatibility with Series 200 machines. The two exceptions are the bit-mapped display electronics and a Model-320-compatible memory management unit (MMU). All

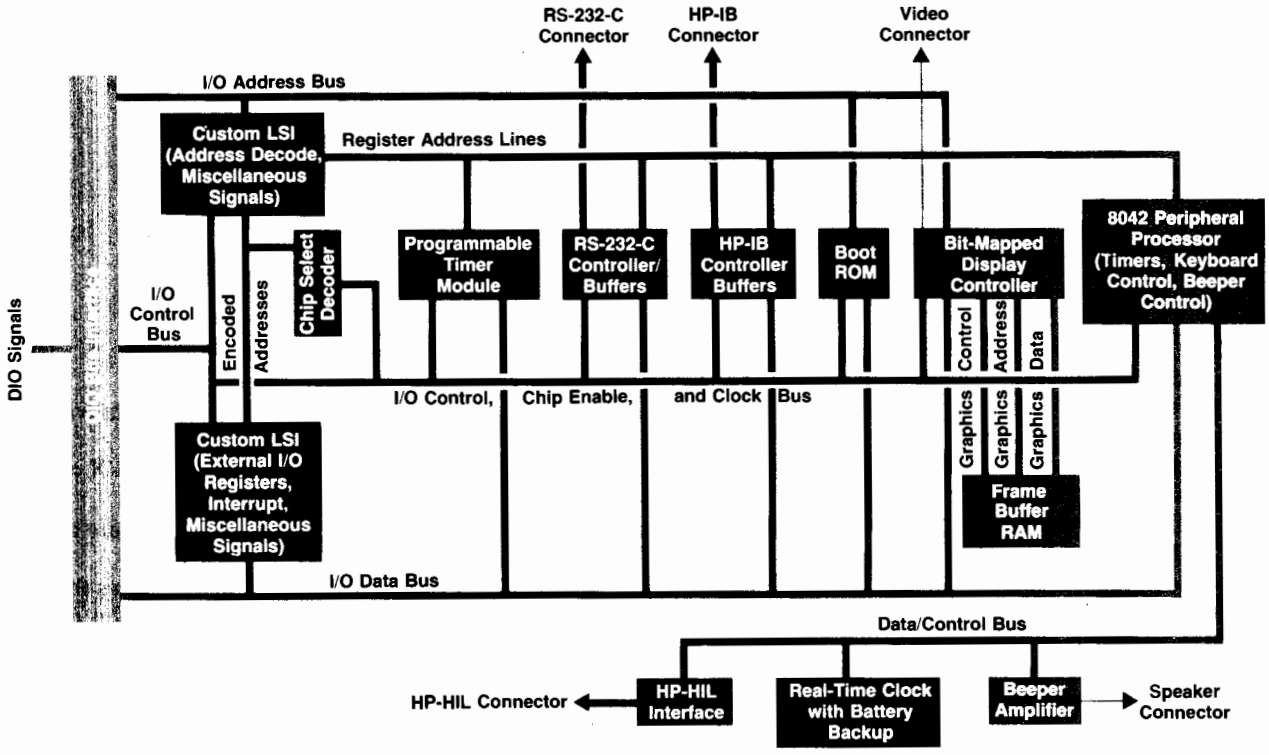


Fig. 2. *Block diagram of I/O subsystem.*

Computer Museum

other hardware architecture changes are transparent to the operating systems and thus compatible. Cost and performance were the reasons for choosing the bit-mapped display architecture that is present on the Model 310 board. Performance, improved capability, and compatibility within the Series 300 family were reasons for the move to the Model 320 MMU.

Cost reduction and manufacturability were high priorities during the design of the Model 310 processor board. The use of custom LSI circuits, careful system partitioning, and careful component type and package selection were three important actions that contributed to cost reduction and manufacturability. By carefully partitioning the system, most of the support logic could be incorporated into four custom LSI circuits. The design of each custom IC was done with the total system needs in mind, thus further reducing the external components needed. Plastic DIP components were selected when available. These components are typically less expensive than other package types and they can be loaded onto printed circuit boards by existing production machinery.

The complexity of the design effort was greatly reduced by partitioning the Model 310 SPU board into two independent subsystems, the I/O subsystem and the CPU/MMU/RAM subsystem (Fig. 1). These two subsystems are connected only at their DIO interfaces, but share some commonly generated clocks. To ensure Series 300 family compatibility, the Model 310's I/O subsystem design was leveraged in developing the human interface card that is used in Model 320 systems.

I/O Subsystem

The I/O subsystem (Fig. 2) consists of the following sections on the processor board:

- HP-IB, RS-232-C, and HP-HIL interfaces
- Programmable sound generator (beeper)
- Battery-backed real-time clock
- Bit-mapped monochrome display electronics
- Programmable timer module
- Up to 128K bytes of boot ROM.

Two of the sections listed above have architectural changes that were added to get additional performance—the HP-IB section and the bit-mapped monochrome display section. As mentioned earlier, the HP-IB section on the Model 310 processor board has additional hardware for performing an HP-IB parallel poll. The article on page 17 provides information about the bit-mapped display controller and what hardware features were incorporated in it to improve its performance.

To keep the device count down and provide all the I/O features listed above, two custom LSI ICs were designed using Texas Instruments' standard cell technology. These two ICs generate all the chip enable signals, DIO handshake signals, DIO buffer control signals, and HP-IB DMA (direct memory access) support signals needed. In addition, all the architected DIO registers needed by the different I/O sections and the HP-IB parallel poll hardware are part of these two custom ICs. These two ICs make it possible to put all the I/O functionality mentioned above on the Model 310 board while still leaving room for the CPU, MMU, and 1M bytes of RAM.

Although the Model 310 is the low-end SPU of the Series 300 family, it has up to 85% of the performance of the earlier high-end Series 200 machine at a cost less than the low end of the Series 200. Unlike the Model 320, the Model 310 does not have a cache memory to improve performance. The addition of a cache would have been prohibitive in terms of expense and board area. Instead, the Model 310 is highly tuned to operate with its 1M bytes of on-board RAM. The 10-MHz 68010 processor runs no-wait-state memory cycles (maximum 68010 performance) out of the on-board RAM even when memory mapping is enabled. The Model 310 can also access RAM over the DIO bus, but these accesses take approximately 1.5 times longer and cause the 68010 to insert wait states in its memory cycle.

MMU/RAM Subsystem

Fig. 3 shows a block diagram of the MMU/RAM subsystem. All 68010 bus cycles to either DIO or high-speed RAM must pass through the MMU where the translation of addresses takes place. The architecture of the MMU is functionally identical to the discrete 32-bit MMU on the Model 320, which is described in the article on page 12.

The MMU/RAM controller was integrated into a Motorola MCA2800ALS ECL gate array because of space restrictions. The MCA2800ALS gate array provides 120 TTL-compatible I/O pins and high-speed operation. The gate array design is highly self-contained, requiring only TLB (translation lookaside buffer) RAMs and a few TTL support chips to form the complete MMU/RAM controller. Generation of system functions such as bus error timeout and bus master arbitration, as well as control of the DIO bus interface and execution of DIO bus cycles are performed by the gate array.

Full support for 256K-bit 120-ns dynamic RAM is provided by the gate array. It is designed to accommodate the 512K-byte or the 1M-byte loading options of the Model 310 board by sensing whether a pull-up or pull-down resistor is present at each RAS line during power-up. The presence

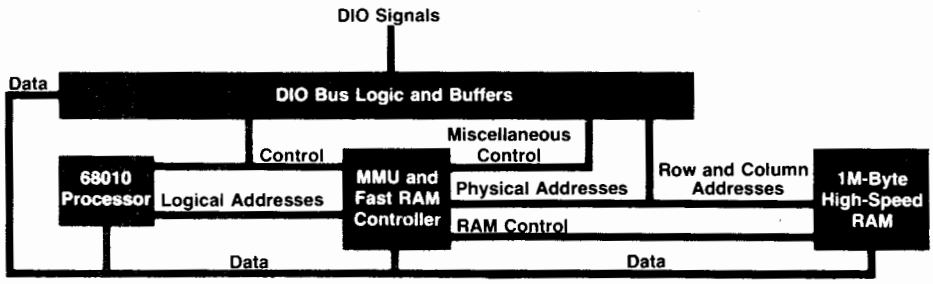


Fig. 3. *Block diagram of MMU/RAM subsystem.*

of loaded parity RAM is sensed by a transition on the parity data lines. Only after a transition occurs can parity fault checking begin.

I/O pins are a precious commodity on the gate array. Rather than allocate five pins to define the starting address of the on-board RAM, the RAM is instead automatically located in a fashion similar to the RAM on some Series 200 processor boards. By scanning down through the memory space, the boot ROM locates the starting address of the on-board RAM by finding a location where no device or RAM responds and a bus time-out occurs. On the first time-out after power-up, the MMU asserts HALT along with BUS ERROR, forcing the 68010 to rerun the bus cycle. At the same time, the RAM's address select decoders are latched at the vacant address. When the cycle is rerun, the on-board RAM is now located at the formerly vacant address and responds to the 68010.

Since there is no cache memory on the Model 310 board, a great deal of attention was paid to optimizing on-board RAM performance. At the beginning of a mapped 68010 bus cycle, the nine low-order address bits from the processor are allowed to ripple through the MMU directly into the RAM array to set up the row address. The upper bits of the logical address simultaneously pass through the TLB RAM. If the translated address selects the on-board RAM, then the appropriate RAS line is immediately asserted to the RAM array since the row addresses have met the setup time. The memory cycle to the on-board RAM is under way even before the DIO bus cycle has commenced. Once the DIO cycle has begun and addresses are latched in the DIO buffers, the column address for the on-board RAM is multiplexed onto the nine low-order address bits out of the MMU and into the RAM array. CAS is then asserted and the memory location accessed.

The dynamic RAM refresh controller uses the eight lower address bits out of the MMU to run a RAS-only refresh cycle to the on-board RAM. As a first priority, the RAM controller tries to initiate a refresh cycle during a DIO cycle. If it cannot bury a refresh during a DIO cycle within 4 μs, then it holds off further MMU activity while a RAS-only refresh cycle is run.

The position of the various control signals to DIO and on-board RAM can be precisely tuned for optimum performance since the gate array's state machines run at the system clock rate of 60 MHz. This fine resolution allows events to occur with little dead time and eliminates the need for analog delay elements on the Model 310 processor board.

High-Performance SPU for a Modular Workstation Family

by Jonathan J. Rubinstein

THE HP 9000 MODEL 320 COMPUTER is the high-performance member of the Series 300 family. It is based on a 16.67-MHz MC68020 microprocessor and an MC68881 floating-point coprocessor. The processor board is a full 32-bit implementation that uses a 16K-byte high-speed cache memory to allow the processor to operate at full speed. A 32-bit memory management unit (MMU) provides up to four gigabytes of virtual address space.

The Model 320 processor board is fully compatible with the 68000 DIO bus architecture, allowing it to replace the lower-performance Model 310 processor board based on a 10-MHz MC68010 microprocessor without any change to the system. This compatibility also allows any HP 9000 Series 200 memory board or I/O card to be used in the Model 320.

The 68020 is the 32-bit implementation of Motorola's 68000 microprocessor architecture. In addition to new 32-bit instructions and addressing modes, the 68020 contains a 256-byte instruction cache (I-cache) and coprocessor support. To increase its performance, a three-stage instruction pipe and instruction overlap are included in the 68020.[1]

The 68881 is the floating-point coprocessor for the 68020. It provides an extension to the 68020 instruction set with full IEEE 80-bit floating-point support, including transcendental functions. Since the 68881 is a coprocessor, the programmer is unaware that it is separate from the 68020 and thus sees the pair as a single processor.

The latest-generation microprocessors have a performance level that matches that of mainframes of a few years ago. To use this performance fully, architectural features similar to those seen in mainframes must be used. A memory hierarchy and memory management are examples of the types of features that can provide higher performance without adding excessive costs or constraining the size of the main memory subsystem. In agreement with this trend, the Model 320 contains a high-speed external cache and memory management unit. (For a general discussion of cache architectures, see reference 2.)

Cache Architecture

After careful analysis of performance, price, and board area, we selected a 16K-byte cache. The cache is implemented with 2K×8 RAMs, which are readily available and do not consume an excessive amount of power because

Computer Museum

of their power-down capability. To keep the complexity down, a "write through" memory update policy is used. This policy implies that a write executed by the processor is written to the cache and the memory in parallel. A write access incurs a penalty equivalent to one or two memory accesses, depending on the size. The Model 320 cache buffers both instruction and data accesses.

The 68020 is capable of accessing memory in three clock cycles. However, it is difficult to implement a cache for the 68020 that can be accessed without additional clock cycles (wait states). If a physical cache is implemented, it is almost impossible with current technology not to introduce wait states. The trade-off is to add one or more wait states for address translation or use a logical cache. If a logical cache is used, the hit rate of the cache is lower because of the cache purges required. However, simulation shows that the hit rate of a logical cache is not lowered to the point of reducing the performance below that which would be obtained if one wait state were added to use a physical cache. If address translation adds two wait states, then the logical cache is the clear choice over a physical cache.

A typical argument against a logical cache is the added software complexity. However, with the 68020, which already contains a logical instruction cache, little additional operating system support is needed for external cache support.

Given these considerations, a logical cache implementation was chosen for the Model 320 system. To achieve no-wait-state access, 35-ns and 45-ns 2K×8 RAMs from Toshiba are used. To build the valid bits for each entry, 25-ns AMD9150 1K×4 RAMs were selected. These RAMs have an additional clear capability, allowing the cache to be purged in one bus cycle.

Once we chose a logical cache architecture, an effort was made to increase the cache hit rate. We found through simulation that if the supervisor or user entries in the cache could be purged separately, the hit rate of the logical cache is only slightly lower than the hit rate of a physical cache. The assumption is that the operating system only purges the portion of the cache that requires it. To implement this enhancement, we use two valid bits for each entry in the cache, either of which can be set or cleared. In other words, every entry in the cache can contain data or instructions from either supervisor space or user space. Under software control all of the user or supervisor entries can be purged.

We selected a line size of 32 bits. The line size is the width of a cache entry and a 32-bit line size allows the processor to access the cache with a single bus cycle. The choice of a set size was more difficult. A set size of either one or two could be easily implemented given the available RAM technology; hence the decision had to be based strictly on the price/performance trade-off. After comparing the increased performance provided by the higher hit rate with the cost of the additional RAM and comparators required for a set size of two, we decided to use a direct mapped (set size of one) approach.

Memory Management Architecture

The Series 300 uses an HP-defined MMU architecture, which provides four gigabytes of virtual memory for each process in the HP-UX operating system, HP's enhanced version of AT&T Bell Laboratories' UNIX™ operating system. The page size is 4K bytes and a two-level, 32-bit table entry, paged MMU is used. The MMU used on the Model 320 is completely compatible with the Model 310 implementation, allowing identical HP-UX kernels.

The table structure supported by the Series 300 is similar to that of the earlier Series 200 Computers. This similarity helps minimize the software effort required to port the operating systems. In addition, the Series 300 MMU is also a subset of the Motorola 68851 PMMU definition, ensuring a compatible growth path with Motorola products in the future. The two-level table walk is shown in Fig. 1.

The 32-bit logical address is divided into three offsets: bits LA22 to LA31 are the offset into the segment table, bits LA12 to LA21 are the offset into the page table, and bits LA0 to LA11 are the offset into the page. The user or supervisor root pointer is chosen by FC2, which is part of the logical address.

The root pointer contains the upper 12 bits of the starting address of the segment table. The segment table offset is concatenated with the selected root pointer to find the segment table entry. The page table offset is concatenated with the segment table address entry (page table pointer), which chooses a page table entry. The page table entry-

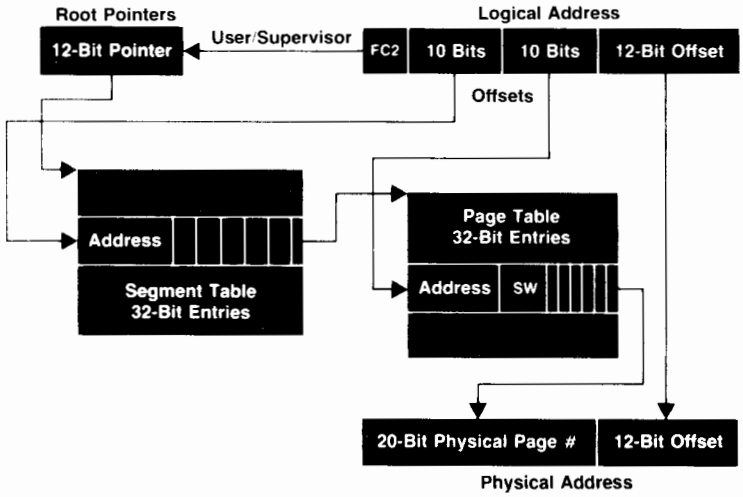


Fig. 1. *Two-level table walk.*

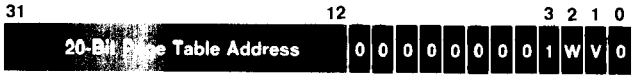


Fig. 2. *Segment table entry.*

contains a pointer to a physical page, and the page offset is used to access the correct byte address.

The segment table entry is defined in Fig. 2. The address entry is the upper 20 bits of a 32-bit page table physical address. The W bit is the write protect bit and the V bit is the valid bit.

The page table entry is defined in Fig. 3. The address portion of the entry is the upper 20 bits of the physical page address. The next four SW bits are ignored by the hardware and are allocated for software use. The D bit is the dirty bit and is set by the hardware when a page is written. The R bit is the referenced bit and is set by the hardware when a page is referenced. As in the segment table entry, the W bit is the write protect bit and the V bit is the valid bit.

A performance enhancement is made to the MMU by adding a cache inhibit bit to the page table entry. The cache inhibit bit CI blocks the cache from loading an entry from the associated page into the cache. This capability is used to keep I/O pages out of the cache and can be used to prevent pages used for DMA (direct memory access) activity from being cached. By not caching pages used with DMA (such as disc buffers), the need to purge the cache on completion of DMA is eliminated.

To speed up address translation, a translation lookaside buffer (TLB) is used to store recently generated translations. In the Model 320, the supervisor and user spaces each have a 1024-entry TLB. A TLB of this size allows eight megabytes of physical space to be mapped simultaneously and simulation shows that the miss rate of the TLB is less than 1%.

In the simplest form of logical cache, the MMU is not accessed until after the cache is accessed and a miss occurs. When this implementation is used, the cache must be purged more often because of the additional purges necessary whenever the operating system is executing MMU housekeeping activities. We avoided these additional purges by accessing the cache and TLB in parallel. The processor cannot use the data in the cache unless there is a valid entry in both the cache and TLB. Although it is impossible to guarantee that if the data is in the cache, the translation will be in the TLB, it is typically so. Thus, having both buffers accessed in parallel leads to little performance degradation caused by TLB misses and increases the hit rate of the cache by reducing the number of purges.

Cache Simulation Description

To design the Series 300 family, we developed a cache and TLB simulation that allowed us to make correct design trade-offs and characterize possible system configurations. To simulate the behavior of the 68020, actual 68000 data traces were obtained by monitoring the backplane of a Model 236 Computer. A special bus interface card was designed to monitor transactions across the bus and allow a second Model 236 Computer to store the data.

Once a 68000 address trace is collected, it is converted to a 68020 trace to simulate the external cache hit rate. We wrote a program to make this conversion and store a 68020 trace. This program simulates the instruction pipe and the I-cache of the 68020. To analyze the external cache, we wrote a cache simulation program that allows parameters such as cache size, line size, set size, replacement algorithm, and associativity to be varied. We later modified the cache program to allow similar simulations for TLB analysis.

Since the hit rate of a cache is not only dependent on the cache configuration, but also on the specific application, different traces are required for different applications. To characterize the Series 200 family, we traced the three operating systems available: HP stand-alone interpreted BASIC, HP Pascal Workstation, and HP-UX.

Thirteen traces were generated: one from BASIC, two from the Pascal Workstation, and ten from HP-UX. The trace from BASIC was a fast Fourier transform with graphics display. Since BASIC is interpreted and stands alone, the cache hit rate is less dependent on the application program than it is with other operating systems. The interpreter uses most of the system resources and thus a larger sample was not required. From the Pascal Workstation, two types of tasks were chosen: a large compile, which is representative of general processing, and the recalculation of a VisiCalc™ spreadsheet, which is computation bound.

To characterize the HP-UX operating system, thirty programs were selected. The HP-UX traces were generated using the following types of applications: floating-point and nonfloating-point intensive arithmetic programs such as B1D and Sieve, the Pascal, Fortran, and C compilers, the vi editor, several graphics intensive programs, disc intensive programs such as find and grep, and other system utilities such as nroff. With all thirteen traces, a total of 200 million accesses were analyzed.

An investigation simulating all possible cache organizations would have been overwhelming. Instead, we chose a few of the most promising cache organizations. Table I shows the cache hit rates for four organizations. The hit rate is calculated by dividing the number of hits by the total number of accesses coming from the 68020, and then multiplying by 100. For the larger caches, the hit rate is more dependent on the percent of writes since these accesses are always counted as a miss. Because of a 64% I-cache hit rate, the hit rates are lower than documented for many systems in the past.

Predicted and Actual Performance

To characterize the performance of systems with similar or identical processor architectures, we use the million-in-

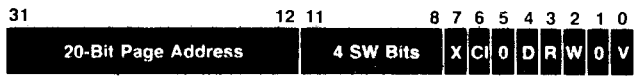


Fig. 3. *Page table entry.*

Computer Museum

Table I
External Cache Hit Rates

Cache size (bytes):	8K	16K	16K	32K
Set size:	1	1	2	1
BASIC system	76.7	84.0	86.7	85.0
Pascal Workstation	65.9	71.7	74.2	76.1
HP-UX: (average)	63.9	69.2	72.0	73.5
(standard deviation)	5.4	4.7	5.3	4.1
Worst hit rate	58.0	65.1	67.0	69.2
Best hit rate	76.7	84.0	86.7	85.0
Average of all traces	65.3	70.8	73.6	74.9
(standard deviation)	6.2	6.1	6.3	4.9

structions-per-second (MIPS) metric. To calculate a MIPS value for the various implementations, the average access time of the processor must be calculated. This access time is defined as the average interval from when the address is valid to when the data is sampled. This time is based on the minimum access time of the processor degraded by the access time to memory or cache. For the cache-based Model 320, the average access time can be calculated as follows:

$$AAT = E_{hr}C_{acc} + (1-E_{hr})\times[W_r(W_{wr}M_{acc16} + (1-W_{wr})\times(M_{acc32})] \quad (1)$$

where E_{hr} is the external cache hit rate, C_{acc} is the cache access time (120 ns), M_{acc16} is the memory access time (540 ns), M_{acc32} is the 32-bit access time (1200 ns), W_r is the percent of misses that are writes, and W_{wr} is the percent of writes that are less than 32 bits.

From the 68020 simulations it was found that with a 64% I-cache hit rate, 17.4% of all 68020 accesses are writes, of which 37% are of byte or word size. Using the above data, Table II shows the possible average access times for the Model 320.

From these average access times and using an I-cache hit rate of 64%, we can calculate the MIPS value using the following equation:[3]

$$MIPS = 1.08(\text{clock frequency})\times[(\text{average clock pulses/instruction}) + (\text{average bus cycles/instruction})\times((AAT/\text{clock period})-2))]^{-1} \quad (2)$$

where the average clock pulses per instruction is 7.159 and the average bus cycles per instruction is 1.201. Table III shows the MIPS value for the system configurations specified above for a 16.67-MHz 68020 processor, using equation 2 and values from Table II for average access time.

Table IV compares measured 68020 performance with that of an 8-MHz 68000 system with one wait state and an 8-MHz 68010 system with 1.5 wait states (0.5 wait state for the MMU), all running identical benchmarks. The 68000

Table II
Average Access Time (ns)

Cache size:	None	8K	16K	16K	32K
Set size:		1	1	2	1
	540	452	392	362	348

benchmarks were run in Pascal on the Pascal Workstation. The 68010 benchmarks were run using C and Fortran on HP-UX. All benchmarks shown use 32-bit integers and 64-bit real numbers.

In choosing benchmarks to compare processor performance, computation-bound programs are preferable since I/O throughput tends to be disc, not processor, dependent. The integer benchmarks (Sieve, Acker, Puzzle, and Search) were collected by the University of California at Berkeley[4] and results have been published for several systems.[5] The UNIX benchmarks were published in *BYTE*[6] and the results are in real times rather than clock cycles. B1D is a standard Whetstone floating-point benchmark (10 million executions) and LFP is a very large, computation-bound, floating-point-intensive program.

The performance of an 8-MHz 68000 system with one wait state or an 8-MHz 68010 system with 1.5 wait states is about 0.5 MIPS. The calculated MIPS value for the Model 320 system is 1.5, which is about three times faster than the 68000/10 systems. From the benchmark comparisons, the measured average relative performance increase is 3.5 (Table V). Note that for the small integer benchmarks, the 68020 has higher performance than typically seen, because the program is loaded entirely into the internal and external caches.

As always, the choice of benchmarks can dictate how well a system fares compared to other systems. The intention of the benchmark data presented here is not to characterize the performance of the Model 320 system, but to correlate the theoretical results with the results from an actual system.

68020/68881 Qualification

To qualify a new system requires many forms of testing. This testing becomes more complicated if a new processor is being used. Not only must the system be proved reliable, but it also must be shown that the new processor executes correctly.

For the Series 300 project, we spent considerable time checking the 68020/68881 processors for correct operation. We also spent many months doing stress/life (strife) testing, and RAM/DMA tests. These tests were in addition to the standard HP Class B environmental tests.

Table III
16.67-MHz 68020 MIPS Values

Cache size:	None	8K	16K	16K	32K
Set size:		1	1	2	1
	0.80	1.30	1.43	1.50	1.54

Table IV
Benchmark Execution Time (milliseconds)

Benchmark Name	Operating System	8-MHz 68000/10	16-MHz 68020	16-MHz 68020 and 68881
Sieve	Pascal	679	158	
Acker	Pascal	5660	1730	
Puzzle	Pascal	23,410	5450	
Search	Pascal	3.4	0.8	
B1D	Pascal	376,480	105,000	14,700
Pipes	HP-UX	9000	4200	
SCall	HP-UX	11,900	4700	
FCall	HP-UX	1300	400	
Loop	HP-UX	11,500	2700	
B1D Fortran	HP-UX	413,500	113,500	14,500
LFP Fortran	HP-UX	2,079,600	732,900	345,600

To qualify the 68020, it was necessary to prove that the 68010 portion of the 68020 was correct before the new instructions and addressing modes could be checked. We used the test code and operating systems available for the Series 200, making only changes required for operation. Once the test code, BASIC system, and Pascal Workstation were operating, the HP-UX operating system was modified to run on the Series 300.

HP-UX tests not only instruction integrity, but also the virtual capability of the processor (i.e., instruction continuation). Any problems found were immediately verified with Motorola so that the problem could be fixed on the next revision of the part. During this testing, the new 68020 instructions and addressing modes were added to the test code and the operating systems were compiled with the newer 68020 compilers. After this process was repeated a few times, a 68020 was available for customer shipments that correctly ran all the Series 300 operating systems and had no known problems that could affect operation.

To help Motorola increase the speed and reliability of the 68020, we also did margin testing for each new mask revision. This testing consisted of low-voltage, high-temperature, and high-frequency tests. The frequency would be increased until the part failed. The failure was then analyzed and reported to Motorola. Using this process, we were able to attain 68020 parts capable of full 16-MHz operation much sooner. This same process was used to qualify the 68881. However, all new code had to be written since it was an entirely new part.

Table V
Benchmark Relative Performance

Benchmark Name	Operating System	8-MHz 68000/10	16-MHz 68020	16-MHz 68020 and 68881
Sieve	Pascal	1	4.3	
Acker	Pascal	1	3.3	
Puzzle	Pascal	1	4.3	
Search	Pascal	1	4.3	
B1D	Pascal	1	3.6	25.6
Pipes	HP-UX	1	2.1	
SCall	HP-UX	1	2.5	
FCall	HP-UX	1	3.3	
Loop	HP-UX	1	4.3	
B1D Fortran	HP-UX	1	3.6	28.5
LFP Fortran	HP-UX	1	2.8	6.0
Average performance:		1	3.49	
HP-UX average:		1	3.10	

Strife and RAM/DMA Testing

Strife testing consists of temperature cycling, power cycling, and vibration testing. The units are placed in a chamber that is cycled from −30° to 65°C. These are eight-hour cycles with three hours at both high and low temperature. During the temperature cycle, the power to the unit is cycled at various times to ensure the maximum temperature swing in the unit. When power is applied to the unit, a self-test program is executed that reports failures to a monitoring system outside the chamber. In addition to temperature cycling, the units are vibrated at 2g random acceleration for 10 minutes every few days.

Strife testing was initiated on the first 16 prototype units. These units were tested for over 80 cycles. When a failure occurred, the testing was terminated until the failure was analyzed and a fix was implemented on all the units. After the 80 cycles were completed, the first 16 production units were tested for 30 temperature cycles with no repeated problems. Naturally, any new failures found in the second strife test were analyzed and promptly corrected.

In addition to strife testing, the integrity of the Series 300 was verified using both long-term RAM error-rate tests and DMA tests. The RAM tests consist of various fill patterns and checking. The DMA tests use two HP-IB cards connected together and various test patterns are transferred back and forth from memory.

These tests were run on the strife units when the testing was on hold or completed. We tested the units at −25°C, room temperature, and 65°C. A total of 35,000 system hours of RAM/DMA tests were run with the failure rate in the range expected for soft RAM errors and no other significant failures were observed.

References

1. D. MacGregor, D.S. Mothersole, and B. Moyer, "The Motorola MC68020," *IEEE Micro*, Vol. 4, no. 4, August 1984.
2. A. Smith, "Cache Memories," *ACM Computing Surveys*, Vol. 14, no. 3, September 1982.
3. D. MacGregor and J. Rubinstein, "A Performance Analysis of MC68020 Based Systems," *IEEE Micro*, Vol. 5, no. 4, December 1985.
4. P. Hansen, et al, "A Performance Evaluation of the Intel iAPX432," *Computer Architecture News*, Vol. 10, no. 4, June 1982.
5. A. Gupta and H. Toong, "An Architectural Comparison of 32-bit Microprocessors," *IEEE Micro*, February 1983.
6. D.F. Hinnant, "Benchmarking UNIX Systems," *BYTE*, Vol. 9, no. 8, August 1984.

Computer Museum

Custom VLSI Circuits for Series 300 Graphics

by James A. Brokish, David J. Hodge, and Richard E. Warner

THE DESIGN OF THE DISPLAY SUBSYSTEM for the modular HP 9000 Series 300 Computers was driven by the need for new levels of performance and flexibility. The overall design approach for this new family of workstations dictated that the display subsystem not only support both monochrome and color monitors, but also support both medium-resolution (512×390 pixels) and high-resolution (1024×768 pixels) displays. Another goal was to make the medium-resolution monochrome system as inexpensive as possible. Compatibility, both within the new family and with the earlier Series 200 products, is important. It was necessary to reduce the component count to improve reliability and make the single-board 68010 processor subsystem possible. To achieve these goals, we decided to implement a bit-mapped system with a custom display controller chip. A second custom chip provides the color map and video digital-to-analog converter (DAC) functions.

A bit-mapped system displays alpha characters on the screen using only graphics display techniques. This is less expensive and much more versatile than the traditional approach of having separate hardware for text and graphics images. Having every pixel on the display directly accessible by the CPU gives the ability to mix text and line drawings on the same screen, and to have multiple character fonts of different sizes and shapes. It also provides the raster support needed by window-oriented human interface programs.

The color display subsystem block diagram in Fig. 1 shows the architecture of the Series 300 display subsystems. The CPU address and data buses are tied to each display controller chip, which in turn moves data to and from the frame buffers. The frame buffer looks like a section of memory to the CPU, where each byte corresponds to a single pixel on the screen. This architecture allows displays from one plane up to eight planes (current implementations provide four or six planes). In a monochrome display only one bit of the pixel byte is relevant and a typical color display has four planes. In color displays, the parallel video from the display controller chips is run to the color map/video DAC chip. The function of the color map/video DAC chip is to map the data from the frame buffers into a specific color on the screen. The display ROM serves two purposes. First, it supplies display characteristics such as initialization constants and the number of display planes to the system software. Second, it supplies default character fonts appropriate for the display.

Display Controller Chip

The display controller chip is a custom integrated circuit

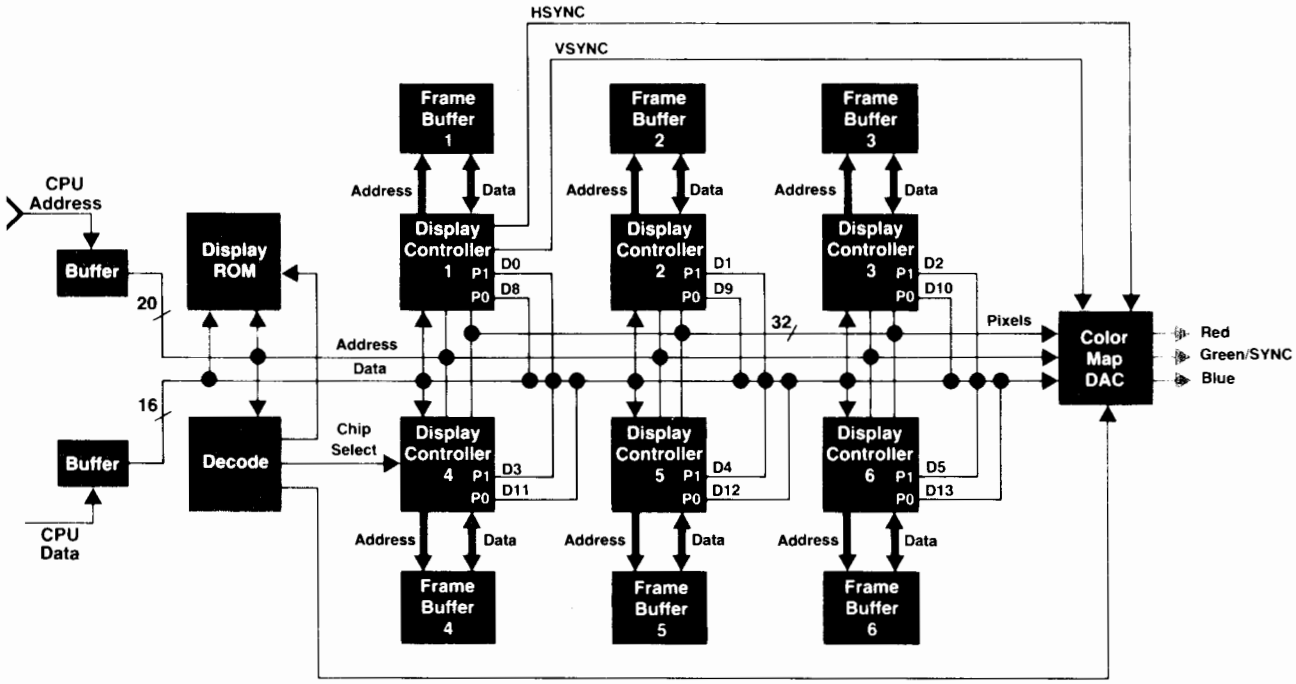


Fig. 1. *Block diagram of color display subsystem for the HP 9000 Series 300 Computers.*

built with HP's NMOS-IIIB process technology. It provides CRT control, frame buffer management, cursor, and bit-BLT (bit block transfer) functions for a bit-mapped display. A single display controller chip can control a monochrome display and multiple chips can be used for color displays.

Each of the display controller chips has two ports to the processor data bus. The first is a typical 16-bit-word data port used for most register accesses. The second is a 2-bit port called the pixel port (see the P0 and P1 pins in Fig. 1). Each display controller's pixel port is tied to two different I/O lines on the processor data bus for sensing and control. This configuration allows the Series 300 system processor to read or write two pixels to up to eight planes with only one access. For writes to the frame buffer, each controller has a pixel logic section to perform logical operations with the incoming data and the existing data in the frame buffer. Random pixels can be written to the display at 0.5 megapixels/s. Pixels written in address order will approach a rate of 2 megapixels/s.

Each display controller chip controls its own frame buffer independently, but in synchronization with the other controller chips. Accesses to the frame buffer consist of two interleaved cycles. The first cycle is a read that is used to refresh the display. The data from the read goes to an internal video shift register which sends video to the color map/video DAC chip. All other accesses to the frame buffer occur during the second cycle. Although a new generation of RAMs gets around this interleaving by integrating the shift register into the RAM, they were not used in this design, primarily because of cost.

The pixel port also serves as a port to single-bit registers inside each display controller chip. This allows the system processor to read or write the same single-bit register of each controller chip in only one access. One of the single-bit registers in the controller chip is the frame buffer write enable bit. To write pixel values only to certain planes, we set the frame buffer write enable bit only on those planes.

Since the word-wide controller chip registers are at the same address for all controllers, we needed a way to determine which controller accepts and/or drives the data on the data bus. To write to the word-wide registers only on certain planes, we set the register write enable bit on those planes. This provides the software a very powerful feature that allows different controller chips to perform different functions at the same time. For example, two different window moves can take place on two different planes at the same time.

On a word register read, only one display controller chip should be allowed to drive the 16 bits of the data bus since the same registers in different controllers may have different data. This is accomplished using the register read enable bit. The software sets only one of the the read enable registers to drive the bus. If the software attempts to enable more than one controller for 16-bit reads, a population detect circuit in each controller senses that more than one of the bits on the data bus is high. This error condition is resolved by returning all controllers to their previous state.

A potential problem with bit-mapped systems is slow character scroll speed. A hardware bit-BLT was implemented to correct this problem. A specified portion of the display is copied to another location on the screen. The bit-BLT feature is useful not only for scrolling up large sections of the screen, but also for adding new characters to the screen. This is accomplished by storing the character font in the areas of the frame buffer not displayed on the screen. The characters are then quickly moved from the off-screen memory to the displayed memory. The bit-BLT feature also acts as a powerful graphics primitive, which can be useful for area fill, line drawing, icon and cursor tracking, and animation.

Word-wide registers inside the controller chips are set up with the source window address, destination window address, window height, and window width. We then write to a window move enable register through the pixel ports to begin the window move on the selected planes. The data moved into the destination window is a function of the replacement rule that specifies a Boolean logical function between the source and destination windows. The replacement rule can be source only, NOT source, NOT destination, source XOR destination, or another Boolean operation. The completion of a window move is flagged by a DONE bit or by an interrupt to the system processor. The controller chip performs window moves at 30 megapixels/s.

A hardware cursor is supported by the display controller chip. The software sets up the position and length of the cursor in selected registers of the controller. Correct substitution of the cursor for the normal video at the rate at which the video leaves the controller is a difficult problem. To work around this problem, the frame buffer data on each side of the cursor is prefetched during vertical retrace. Operations are then performed to combine the cursor with the prefetched data. When the time comes to shift out the cursor, the manipulated data is substituted for the normal data from the first half of the interleaved cycle.

The display controller chip also provides the signals for CRT control. Horizontal and vertical sync signals and a

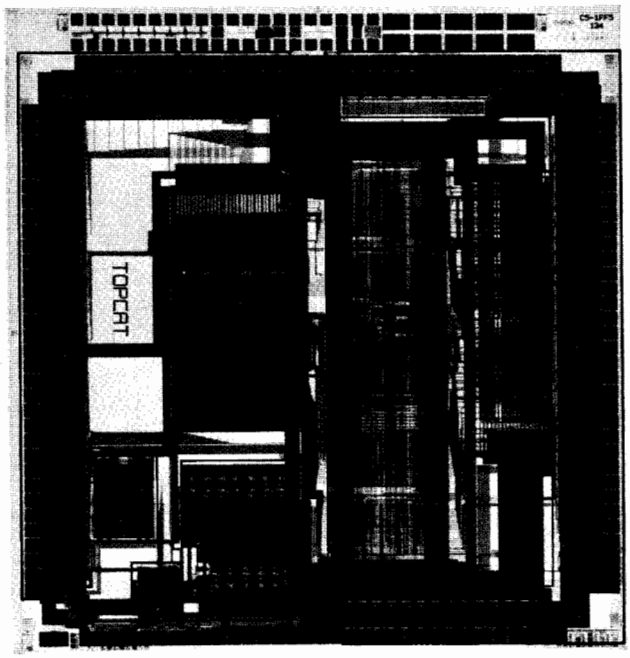


Fig. 2. *Photograph of custom display controller chip.*

composite blanking signal can be programmed to drive virtually any monitor.

Display Controller Architecture

A photograph of the display controller chip is shown in Fig. 2. The key components are a 16-bit address stack, a 32-bit data stack, the main PLA (programmable logic array), and the test PLA. The 16-bit address stack is used to calculate the addresses for frame buffer reads and writes. It also contains the majority of the registers directly accessible by the system processor.

The 32-bit data stack primarily contains data going to or coming from the frame buffer. The display refresh cycle described above dumps the data into a shift register on the data manipulation stack. A cyclic redundancy check (CRC) register on this stack performs a CRC on the video data as it is loaded into the shift register. This register can be read by the system processor to give very high confidence that the display subsystem is functioning properly. Also in the data stack are a 32-bit barrel shifter and a logic unit used for window move operations.

The main PLA is the largest section of the chip and controls all normal operation of the display controller chip. The test PLA and diagnostic interface port allow a chip tester to halt and single-step the chip operations. They also allow the tester to preset the chip to a certain state.

Color Map and Video DAC

The color map/video DAC chip is fabricated using HP's NMOS-IIIA process technology. It combines a 256-entry color map and three eight-bit DACs onto a single integrated circuit. It can drive a 1024×768-pixel, 60-Hz display with video data at approximately 64 MHz. Its RAM can supply data at rates up to eight times faster than conventional NMOS-III RAM designs. The keys to this eight-fold increase in speed are the chip architecture and RAM design.

The chip contains five major functional blocks: the processor interface and control, the video data pipeline, the RAM, the DACs, and the test logic. The processor interface allows the system processor to read and write the color map RAM, inquire about chip and display status, and perform signature analysis on a variety of signals within the chip. The chip provides a set of registers that are mapped into the processor's memory address space. The two basic types of operations are register accesses, which allow the system processor to read and write the address and data registers within the chip, and control accesses, which cause the chip to perform an internal operation such as reading or writing the RAM. The control accesses are the same as register writes from the point of view of the system processor, except that the data has no significance. The status register can be read to provide information about the internal state of the chip. It has bits that indicate the current states of several display timing signals and whether the RAM is busy performing a read or write operation.

Video data is supplied to the chip four pixels at a time. This simplifies the display subsystem by reducing the external clock rate. The chip has 32 video data pads, so each of the four pixels can have up to eight bits. Video data from the pads is routed (at approximately 16 MHz in high-resolution systems) to the input multiplexing and masking logic. First, the video data is split into two data streams of two pixels each. One stream consists of the first and third (odd) pixels, and the other the second and fourth (even) pixels. The multiplexing logic then alternately selects the pixels in each data stream and passes them to the masking logic. This logic provides a programmable mask that can be used to select which of the eight bits per pixel are routed to the color map RAM. For displays with fewer than eight planes, this feature is used to mask off the data bits for planes that are not present. It can also be used to disable the display of selected frame buffer memory planes. At this point, there are two data paths, each passing video data at approximately 32 MHz to separate RAMs.

The chip contains two identical 256×24-bit dynamic RAMs, one for the even pixels (see Fig. 4) and one for the odd pixels. This effectively doubles the RAM bandwidth. The RAM structure and cells are based on the four-transistor NMOS-III RAM described by J.W. Wheeler, et al.[1] The RAM arrays are arranged as 32 rows by 8 columns for each of the 24 bits. The basic clock rate of the color map RAM is approximately twice that of the RAM described by Wheeler for the HP 9000 Series 500 Computers (32 MHz versus 18 MHz), which again doubles the effective RAM bandwidth.

The operation of the color map RAM is very similar to the operation of the memory array of the Series 500 RAM. The basic memory cycle is four clock phases (two clock cycles). The first phase precharges the RAM array and the sense amplifier. During the second phase the row and column selects are driven and the cell data is driven to the sense amplifier. During the third phase the sense amplifier is disconnected from the array and allowed to stabilize, and during the fourth phase the output data is driven from the sense amplifier. To increase the bandwidth, two sense amplifiers are used, and the RAM cycle is pipelined so

(continued on page 21)

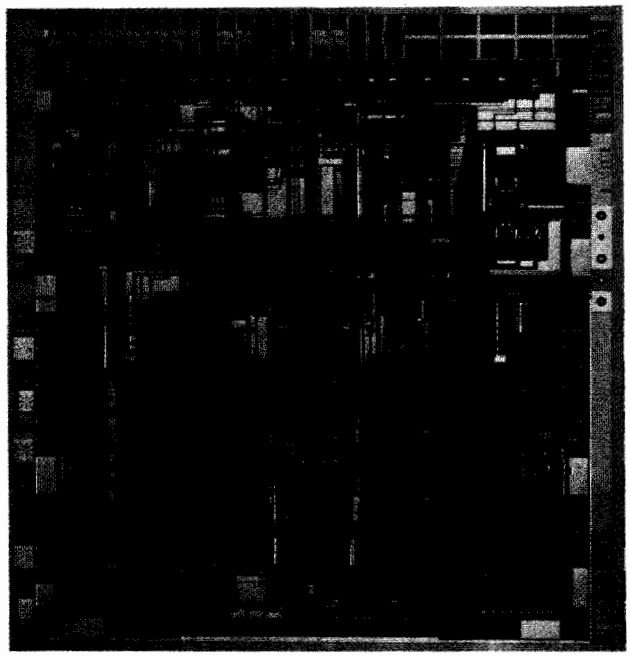


Fig. 3. *Photograph of the color map/video DAC chip.*

Display Custom IC Design Methodology

The display logic subsystem was targeted for significant price/performance improvements early in the definition of the Series 300 workstations. Previous HP workstations used two printed circuit boards with over 200 components to implement the display logic. Early estimates were that a custom display controller chip could provide the equivalent logic with less than 20 components for one quarter of the cost, while also reducing power consumption and improving reliability.

The architecture of the display controller chip dictated the design of several special logic circuits. These circuits include a logic unit, an adder, a bit field extracter, a comparator for greater-than-or-equal, and a video shift register. The high-speed requirements for these circuits led to the selection of HP's custom VLSI NMOS-III process.[1,2,3] The design methodology used was an enhanced version of that used for the design of the NMOS-III ICs for the HP 9000 Series 500 Computers. The tools used feature hierarchical design and layout, and run on HP 9000 Series 200 and Series 500 workstations with HP's Shared Resource Management (SRM) system. The methodology was developed to produce fully functional parts on the first pass, a goal that was achieved for the display controller chip.

A major feature of the design methodology is a transistor-level logic simulator which ultimately was used to verify the functionality of the entire chip. Proper logic functioning and sequencing were verified with the entire chip described at the transistor level. The simulator, called LISIM, simulates the transistors as unidirectional and bidirectional switches. LISIM uses two states (zero and one) and 63 strengths to resolve conflicts between multiple driving transistors. It solves the charge sharing problem on circuits driven by precharged buses by assuming that, when the strength is zero (no active drivers), ones have a higher strength than zeros. This results from our methodology in which zeros are always actively driven while buses may be precharged to one and allowed to float. LISIM was able to simulate the entire 77,000-transistor design on a Model 236CU Pascal workstation with two megabytes of memory at 19 seconds per input vector.

The LISIM transistor-level simulator allowed the design team to go directly from an algorithmic design to transistor-level schematics (for those cells not already in the NMOS-III library), bypassing the need to create and verify a logic-gate-level description. LISIM also made it possible to use the test vectors developed during logical design on the netlist that was extracted directly from the physical artwork.

The sequencing and logic operation of the display controller chip are controlled by a programmable logic array (PLA). The PLA was specified in a Pascal-like program. This program was then compiled using a program called QuickPLA that produced optimized logic equations. These PLA equations, along with test vectors, were used to simulate the logical operation of the chip. They were also used, along with a signal order list, as input for the PLA module generator. This program generated optimized artwork directly from logic equations in approximately two hours. Several iterations, including whole chip simulations, were required before the final PLA artwork was produced. Nevertheless, several months were saved by using the PLA generator.

The remainder of the chip is partitioned into blocks that the PLA controls. These blocks were designed using an in-house schematic capture package called SCIP. Beside schematic capture, SCIP features a schematic evaluator that outputs a FET list and a netlist, which it formats for both the HP Spice analog circuit simulator and for the LISIM digital logic simulator. Once each block's logic design functioned correctly within the chip simulation, the transistor sizes were chosen for speed, zero level, and power and then simulated using HP Spice.

Once the logic design was done, the physical design (artwork) was next. This was simplified by using stretchable cells and FET primitives. After the initial layout was done, the artwork was submitted to the hierarchical artwork system for design rule checking, netlist and component evaluation, and encapsulation. Encapsulation hides design details not relevant to higher-level blocks, so the designer sees the encapsulated block as simply an outline with bristles for its ports. This makes higher-level composition, design rule checking, and encapsulation a much faster process since the design rules need only be checked at block boundaries.

The chip floor plan was manually generated, taking into account optimal pad positions for signals as well as multiple power and ground pads. The clock and test pads were constrained by the methodology to occupy standard pad positions to ease testing. The power supply routing was modeled with current sources and resistors representing the current drawn by the cells and the size of the metal power lines routed between them. Then the routing model was simulated using HP Spice. The PLA and the routing channels between it and the register stacks on either side were automatically generated within the form factor constraints of the floor plan. While the PLA routing was not 100% complete, the router did save us many hours of manual editing.

Hierachical netlists described by a block description language were generated from both schematics and artwork. These netlists were automatically compared at each level of the hierarchy. These comparisons were sufficient to find all errors in intercell connections, logic implementation, and FET sizes within manually generated cells. Some of the cells with automatically generated artwork (such as the PLA) created simplified netlists for initial functional simulation that did not exactly match the extracted artwork. These cells were, however, verified by the final LISIM simulation using the actual artwork.

The LISIM simulator was rerun to verify the tools just before mask release on the netlist extracted from the artwork. In future designs this step could be omitted since all of the artwork has already been compared automatically with the corresponding logical schematic.

The custom Series 300 display controller chip does not use special circuitry for testing its data path cells. All cells with access to the main data buses are loaded from and dumped to a scannable register in the test section of the chip. The inputs and outputs of the PLA are made scannable so that the PLA itself can be tested and can control all of the register load and dump signals. This allows control and observation of the entire data path without additional circuitry. The pads were also made scannable so they can be completely tested from the serial test port without requiring an expensive high-speed tester. The pad scan path also made it possible to test the first-pass chips with the same vectors used during the design simulation. This was accomplished by scanning them into the pads serially and then single-stepping the chip.

References

1. J.M. Mikkelson, et al, "NMOS-III Process Technology," *Hewlett-Packard Journal*, Vol. 34, no. 8, August 1983.
2. J.P. Roland, et al, "Two-Layer Refractory Metal IC Process," *ibid.*
3. H.E. Abraham, et al, "NMOS-III Photolithography," *ibid.*

Bruce P. Bergmann
Member of the Technical Staff
Technical Workstation Operation
Kenneth P. Sandberg
Member of the Technical Staff
Fort Collins Systems Division

Computer Museum

that during phase three, the array is again precharged to begin another RAM cycle using the second sense amplifier. Since screen refresh is performed continuously, this allows the RAM to provide a new data word each clock cycle (32 MHz). This increases RAM bandwidth by an additional factor of two, so the total color map RAM throughput is approximately eight times the throughput of the Series 500 RAM.

The outputs of the even and odd RAMs are alternately selected and routed to the DACs. The RAMs also have refresh counters and control logic. Refresh occurs automatically during retrace when no video is being generated. When the processor performs a read or write access on the color map, the address and data are latched until the next retrace, when the cycle is performed. Since an access can occur during each horizontal retrace, it is possible to update the entire color map in one screen refresh period.

The 24 bits of RAM output are distributed to three DACs in the Series 300. One DAC is used for each of the primary colors (red, green, and blue). Each DAC has eight bits of resolution, so the display has 16 million possible displayable colors. Each DAC has 255 current-steering transistor pairs for driving the video levels and 28 pairs for driving the blank level. The transistor pairs are built up in slices of eight for a single input bit, and the slices are combined into a stack with the slices arranged so that the transistor pairs for the higher-order bits are distributed within the array to minimize the effects of process variations. Reference current generators are distributed through the DAC array. The generators are calibrated by an external reference current supplied to the chip.

The chip DAC outputs are guaranteed to be monotonic, which means that for any two input values, the output current for the larger input will not be less than the current for the smaller input value. The linearity specification for each of the DACs requires the output current for a given input value to be within ±2.5% of the ideal current for that value. Tracking measures the difference between the output currents of two different DACs on the same chip when both have the same input, and is specified to be ±3.5% for any two DACs on the same chip.

Testing can be a problem for VLSI circuits, so the color map/video DAC chip incorporates some additional features to provide testability. One is a signature analysis capability with an input multiplexer that allows signatures to be taken on a number of different internal signals. The signals that can be tested include all of the input data bits and all of the RAM output bits. The input data bit signature can be used to verify correct operation of the input data path from the input pads through the mask and multiplexing logic, as well as the integrity of the data stream from the frame buffer. The RAM output signatures can be used to test each individual bit of the RAM.

The color map/video DAC chip also incorporates a serial diagnostic port similar to the one used in the custom CPU for the HP 9000 Series 500 Computers,[2] but uses only four pads on the chip (input data, output data, enable, and data strobe). This version of the diagnostic interface port was designed for the color map/video DAC chip and the same design and its descendants have been used in a number of other HP chips, including the display controller chip. The diagnostic interface port can be used to scan values into any of the internal I/O registers, and to execute commands that simulate all the normal chip functions and several purely diagnostic functions.

Acknowledgments

Art Dumont, Tony Walker, Bill Cherry, and Mark Coleman designed the color map/video DAC chip. Doug Buhler and Jim Jackson were part of the display controller chip design team. Brad Reak designed the display boards that

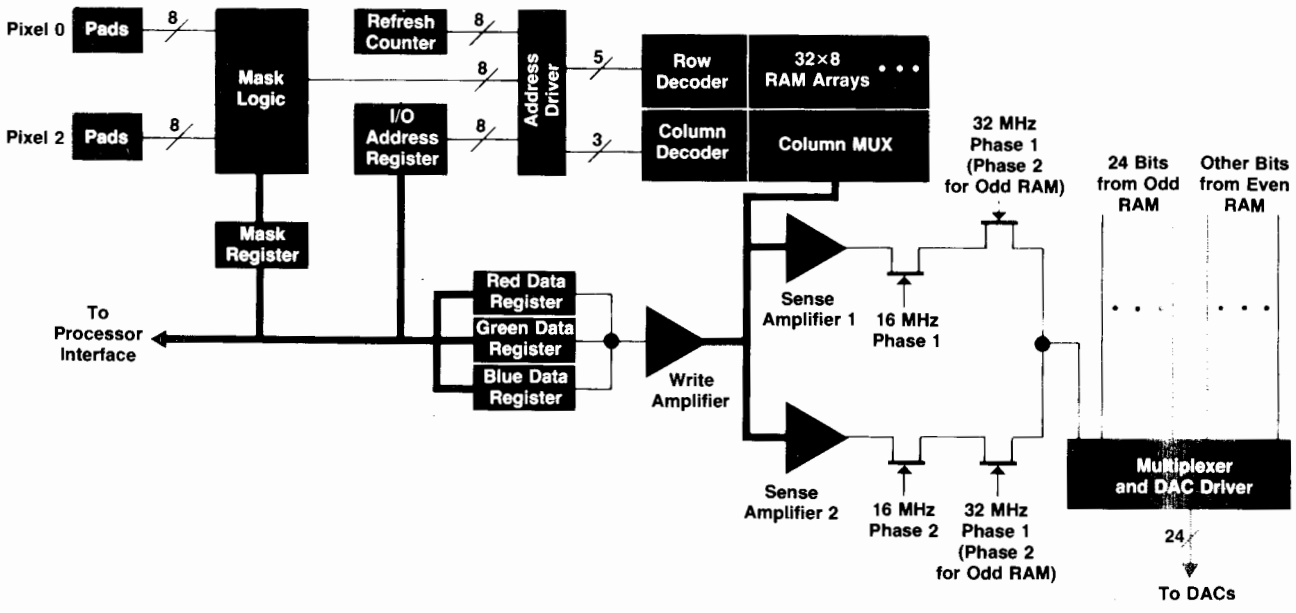


Fig. 4. *Block diagram of even-pixel RAM on color map/video DAC chip.*

use the chips. Dan Griffin managed the color map/video DAC chip project and Gary Taylor managed the display controller chip project. Many others in the Fort Collins IC Division's R&D lab and MDE organization made crucial contributions to the success of both chips.

References

1. J.W. Wheeler, et al, "128K-Bit NMOS Dynamic RAM with Redundancy," *Hewlett-Packard Journal*, Vol. 34, no. 8, August 1983.
2. K.P. Burkhart, et al, "An 18-MHz, 32-Bit VLSI Microprocessor," *ibid.*

Software Compatibility for Series 200 and Series 300 Computers

Several software obstacles exist for the Series 200 user who wants to move to HP's new family of modular workstations, the HP 9000 Series 300. This article identifies these obstacles and describes the features of BASIC 4.0 (the latest release of HP's enhanced version of the BASIC language system) designed to overcome them.

by Rosemarie Palombo

PRESERVING THEIR INVESTMENT in software is a primary concern of most computer users. They also want the flexibility to migrate to new more powerful computer systems and take their software with them. As the installed base of HP 9000 users grows, it is imperative that these needs be addressed.

Our main compatibility goal is to preserve the software investment of our customers by giving them the ability to run their existing software, without change, on state-of-the-art hardware. In addition, compatibility should be achieved without giving up any of the functionality of the new computer system and should extend throughout the entire family of workstations.

Hardware Differences

The Series 300 family of computers differs from its predecessor, the Series 200 family, primarily because the opportunity was taken during the development of the Series 300 to incorporate new technology into the existing line of HP 9000 Computers. The new hardware design resulted in some differences in machine characteristics between the two families of workstations. These differences are:

- Series 300 displays have bit-mapped planes with combined alpha and graphics. The Series 200 family has separate alpha and graphics planes.
- Some Series 200 alphanumeric highlights are missing from Series 300 displays. Gone are blinking mode (except for the alpha cursor) and half-bright intensity.
- The new Series 300 displays have different graphics resolutions. For example, on a Series 300, medium-resolution graphics displays have 512 horizontal by 400 vertical graphics pixels, whereas many of the Series 200 graphics displays have a resolution of 512 by 390 pixels.
- The new Series 300 displays have different alphanumeric and graphics hardware addresses.
- The Series 300 color map is different from that of the earlier Model 236C Computer, in that the Series 300 color map is used for alpha as well as graphics.
- Series 300 computers do not have a built-in ID PROM for software security. However, an equivalent feature is provided by an optional HP-HIL device—the HP 46084A ID Module (see article on page 4).
- Two new processor boards are available with the Series 300. One contains a 10-MHz, 32-bit MC68010 microprocessor and the other, a high-performance board, contains a 16-MHz, 32-bit MC68020 microprocessor with a new internal instruction cache, a different on-board external cache memory, and an MC68881 floating-point arithmetic coprocessor. This difference doesn't affect software portability, but some changes may be required to achieve specific performance goals.
- The serial RS-232-C/V.24 I/O interface on the Series 300 differs from the serial I/O interfaces of the Series 200 Computers in that the Series 300 interface has no hardware configuration switches.
- The HP-HIL keyboard differs from Series 200 keyboards, except for the Model 217 and Model 237 Computers.

BASIC Language

The Series 200 BASIC Language system was first introduced in 1981 concurrently with the HP 9826 Computer. It is a language well-suited for a wide range of instrument control applications, computer-aided design needs, and general computation.

Computer Museum

Series 200 BASIC gained popularity for many reasons. Among them are its enhanced I/O capabilities, easy-to-use program development environment, structured programming features, and interactive, friendly human interface complete with knob (rotary pulse generator), softkeys, and softkey labels.

HP has continually added BASIC support for new members of the Series 200 family. HP's version of BASIC has been revised several times to support new hardware and provide additional software capabilities. In every case, support of older hardware was retained. Eventually, the gap created by the dissimilar hardware systems had widened enough (because of new state-of-the-art technology) to warrant the development of an entirely new family of HP 9000 Computers — the Series 300. Maintaining software compatibility presented a challenge for the developers of BASIC 4.0. This latest release of BASIC provides support for this new family of HP 9000 Computers and adds several new human interface capabilities, some designed especially with software compatibility as their goal.

Display

One area of change imposed by the Series 300 is the display technology. The integrated alpha and graphics displays of the Series 300 are produced by bit-mapping hardware and normally cannot be independently toggled. The Series 200 implementations have separate graphics and alpha screens where the graphics screen is bit-mapped and the alpha screen is produced by character-generation hardware. This makes it possible to turn the Series 200 alpha and graphics screens off and on independently. For many Series 200 users, the use of the separate ALPHA and GRAPHICS commands to turn the alpha or graphics displays on or off are fundamentally natural acts. A solution was needed that would allow Series 200 BASIC software that uses these features to run on the new Series 300 hardware.

One response to this need is the HP 98546A Display Compatibility Interface. This display interface provides Series 300 users with the separate alpha and graphics capability of Series 200 computers (except the Model 237, which has a bit-mapped display like the Series 300). The design of the HP 98546A is similar to that of the HP 98204B video board set for the Model 217 Computer, with the addition of an electronic video switch to allow switching between this Series 300 video board and perhaps another Series 300 video board. The switch is controlled by a register on the

Fig. 1. *The HP 98203B (top) and HP 46020A (HP-HIL, bottom) keyboard layouts.*

board and can be activated by a software write. This will shut down the current signal to the monitor and switch to the other video signal. The HP 98546A has the same alphanumeric and graphics characteristics as the HP 98204B. Therefore, any Series 200 program that is display-compatible with the HP 98204B will run on a Series 300 with the HP 98546A Interface and an appropriate monitor.

Any Series 300 Computer can be configured with the HP 98546A Interface. It can be the only video board present or it can be used in conjunction with any other Series 300 video board, medium- or high-resolution, monochrome or color. In the case where the Series 300 Computer contains both the HP 98546A and a high-resolution video board, two separate monitors are required. Use of the HP 98546A does not restrict the functionality of the Series 300 in any way, nor does it require any program changes. A limitation of this solution is that the HP 98546A only drives a monochrome display.

BASIC 4.0 supports the HP 98546A Interface and provides an easy way to select display boards whenever the HP 98546A and a Series 300 bit-mapped display board are present. The execution of a single CONTROL statement selects the specified display. The BASIC 4.0 system performs all the tasks required to switch displays and initialize alpha and graphics on the new display to the power-on defaults.

Some members of the BASIC 4.0 development team investigated another solution to the display compatibility problem. Was there any way that separate alpha and graphics could be provided on bit-mapped hardware by the system software itself? If so, this would eliminate the need for an HP 98546A Interface. The answer is that in some cases it is possible to simulate this behavior. Thus, another compatibility alternative for applications software that depends on the ability to manipulate separate alpha and graphics planes evolved and was designed into BASIC 4.0.

With multiplane displays it is possible to designate some planes for alpha and the remaining planes for graphics and subsequently perform alpha-only or graphics-only operations. The display controllers of the Series 300 color video boards allow selective plane read/write enable. There are plane control registers in which each bit controls its respective plane. By enabling and disabling planes properly, the functionality of separate alpha and graphics planes can be emulated.

Using this information, the BASIC 4.0 team decided to add a new feature that provides the ability to specify which planes to write-enable for alpha and which planes to write-enable for graphics. The graphics write-enable mask, accessed through the GESCAPE statement, indicates the frame buffer planes to be written to by graphics operations, and a CONTROL statement is used to designate the alpha planes (set the alpha write-enable mask).

With a four-plane color display we can designate planes 1, 2, and 3 for graphics and plane 4 for alpha. This provides only eight pure graphics colors, instead of 16, and a single alpha color. Restricting the number of planes that are write-enabled for alpha or graphics to less than the total number available will also restrict the number of color map pens available for use.

On bit-mapped color display hardware, this emulation gives many of the capabilities of a separate alpha and graphics system, including:

- Turning alpha and graphics off and on independently
- Dumping graphics without embedded alpha
- Independent scrolling of alpha and graphics.

This compatibility solution requires no source program changes since the appropriate masks can be set by a short configuration program.

Keyboard

When the Model 217 and Model 237 Computers were introduced, so was a new human interface, the HP-HIL (Hewlett-Packard Human Interface Loop), which included a new keyboard. This keyboard is used by the Series 300 and is different from the HP 98203A/B keyboards used by earlier Series 200 machines. The major differences between the HP 98203A/B keyboards and the HP-HIL keyboard are the number and layout of user and system function keys and the location of the rotary control knob. The number and size of the screen labels for typing aids are also different.

The HP-HIL keyboard has eight physical user function keys, labeled f1 through f8, while the HP 98203B keyboard has ten such keys, labeled k0 through k9 and the HP 98203A keyboard has only five function keys (see Fig. 1). Although the HP-HIL keyboard has fewer physical function keys, it

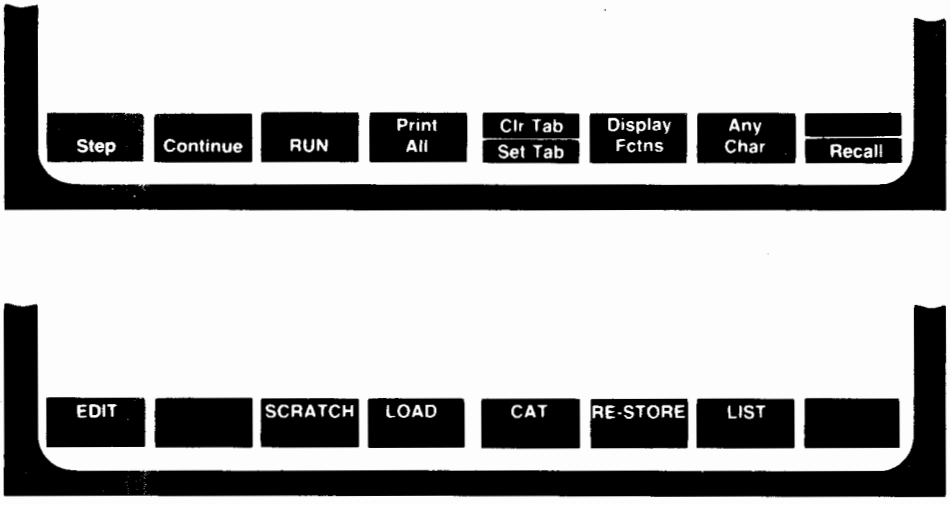


Fig. 2. *The HP-HIL system menu of keys (top) and default typing-aid labels (bottom).*

Computer Museum

has more functionality than the HP 98203B keyboard. This is because BASIC 4.0 and BASIC 3.0 provide one menu of system keys and three menus of user definitions for the eight physical function keys, giving 24 user-definable keys compared to 20 such keys on the HP 98203B.

There are several keys on the HP 98203B keyboards, such as **STEP** and **CONTINUE**, that are not among the keycap labels of HP-HIL keyboards. These functions no longer have dedicated keys, but BASIC 4.0 makes these functions available in the system menu at all times (see Fig. 2). Other system key functions, such as ALPHA and RECALL, are available through dedicated but unlabeled keys (RECALL is also available in the system menu). A keyboard overlay was designed for BASIC users with HP-HIL keyboards to identify unlabeled keys, system menu keys, and some details of the HP 98203B keyboard compatibility mode (see Fig. 3).

A keyboard compatibility mode for BASIC 4.0 emulates some of the missing features of the HP 98203B keyboard when using an HP-HIL keyboard. This mode provides a convenient way of porting Series 200 programs to Series 300 machines without modifying the source program. In particular, it was designed to provide compatibility for Series 200 programs that were written for ten user function keys and their corresponding keylabel display. When a nonzero value is written to keyboard control register 15, keyboard compatibility mode is enabled. The HP-HIL function key row now acts as HP 98203B keys **k0** through **k9** with the HP-HIL **Menu** key acting as **k4** and the HP-HIL **System** key acting as **k5**. Similarly, the HP 98203B softkeys k10 through k19 are accessed by pressing the HP-HIL **Shift** key with the appropriate redefined function key. In this mode there is one row of keylabels for the display. Each label can contain a maximum of 14 characters and is formatted into two rows of seven characters each. If a label contains more than seven characters, it is wrapped around to the second row of characters (see Fig. 4). This softkey label format was chosen because it corresponds closely to the function key layout on the HP-HIL keyboard and it was the first choice of our human factors consultant, lab engineers, and customers who used it during development.

To emulate the HP 98203B keyboard and its softkey behavior fully, only three statements need to be executed to configure the keyboard and keylabel display. In this mode, the HP-HIL system menu of functions is available when the **System** key is pressed along with the **Extend char** key. This is documented on the new keyboard overlay.

However, keyboard compatibility mode is not as effective as we would like because:

- Displayed keylabels may be different since the length and format of the labels are not strictly identical to the HP 98203B version.
- Certain keycodes are not available on an HP-HIL keyboard, and therefore the corresponding keystrokes cannot be trapped.
- The HP 98203B system keys (for example, **RUN**) require two HP-HIL keystrokes, that is, **Extend char** with one of the function keys.
- The rotary control knob is not on the HP-HIL keyboard. However, it is available as a separate input device, the HP 46083A HP-HIL Knob.

Serial Interfaces

Another problem exists because the Series 300 features a built-in RS-232-C/V.24 serial interface that differs slightly from some of its Series 200 counterparts. Since the goal was to provide a low-cost interface in the Series 300, there are no hardware configuration switches for select code, interrupt level, baud rate, and line control parameters. If a Series 200 program depends on serial interface configurations as set by hardware switches, some software configuration is required to run the program on a Series 300.

To work around this difference, BASIC 4.0 sets default values for the baud rate and line control parameters and allows the user to change these defaults, thereby emulating the hardware switches. The select code and interrupt level are hard-wired to specific values and cannot be software controlled. During power-up, the BASIC 4.0 system sets defaults for these values. If a program expects values other than the defaults, they can be set by keyboard execution, a short configuration program, or the AUTOST routine by writing to the appropriate control registers (13 and 14). Only the cycling of power or writing specifically to these registers will change these values.

This compatibility solution requires no source code

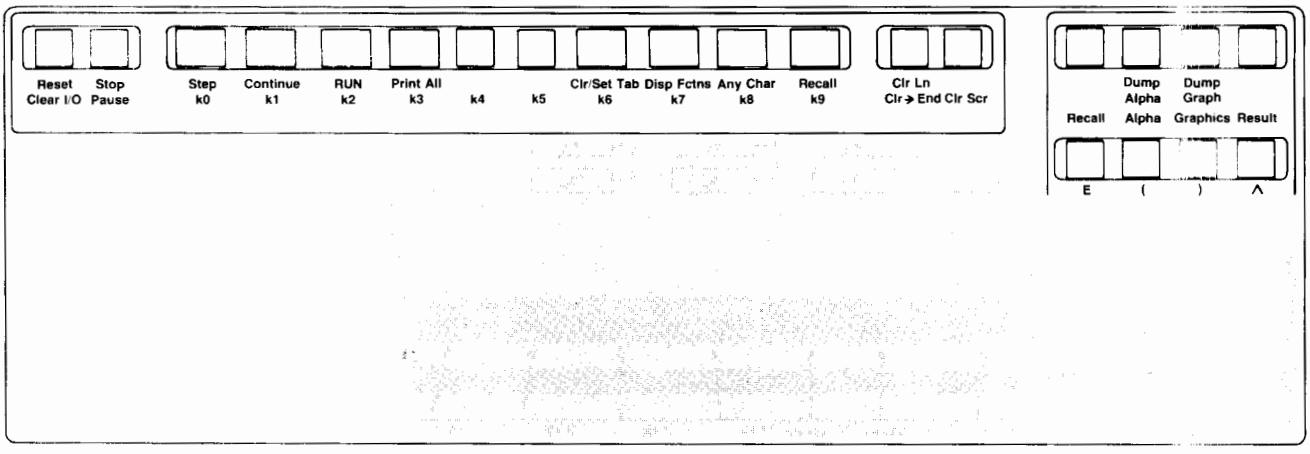


Fig. 3. *The BASIC HP-HIL keyboard overlays.*

changes. Another set of registers (3 and 4) is used to maintain the current values of these parameters, which can differ at any time from the default values. The current values can be changed by writing specifically to the appropriate control registers. However, if the interface is reset with a SCRATCH A statement, the values in these registers are restored to the default values stored.

Security

Many models of the Series 200 have a built-in ID PROM, which allows software security encoding. An equivalent feature is provided for Series 300 Computers by an optional HP-HIL device, the HP 46084A ID Module. This security module plugs into the HP-HIL interface card in every Series 300 Computer.

The ID Module returns a unique character string that contains its product number and serial number in a densely packed format. The ID Module is compatible with code that expects an ID PROM, but the character string returned by the SYSTEM$("SERIAL NUMBER") function is not in the same form as that returned by Series 200 ID PROMS. Therefore, some string manipulation is required to obtain a human readable form. This requires additional statements to be added to any Series 200 code that reads the ID Module for use on the Series 300. The required statements are included in the BASIC 4.0 documentation to make this an easy transition for users.

Limitations

As with any solution to a challenging problem, there are trade-offs to be made. Certain machine characteristics were impossible to duplicate. These incompatibilities are confined to the following areas:

- Series 200 Programs that strictly depend on the presence of an ID PROM will not work without some source code changes.
- BASIC 3.0X CSUBs (compiled subprograms) must be regenerated using the BASIC 4.0 Compiler Subprogram Utilities and the Pascal 3.0 or 3.1 language system, since the entry points to the BASIC system have changed. This has been true for every major revision of BASIC.
- Programs that depend on keycodes that cannot be generated by either the HP-HIL keyboard or the HP 98203B emulation mode (e.g., the HP 98203B EDIT and EXECUTE keys) will require changes.
- Programs that expect an I/O select code other than 9, or an interrupt level other than 5 for the built-in RS-232-C/V.24 serial interface will require changes.

The BASIC 4.0 development team reviewed these incompatibilities and found that a hardware or operating system solution was not feasible. The only viable solution for these problems was to document their likelihood for occurrence and, where appropriate, to specify the source changes required. Hence, a set of compatibility and porting documents was developed to assist customers in dealing with compatibility issues.

Pascal Workstation

BASIC is not the only HP 9000 language system that incorporated Series 200 compatibility support into its latest release. The Series 200 Pascal Workstation had identical overall goals and used similar techniques to meet the challenges of providing software compatibility to its customers. These include support of the HP 98546A Display Compatibility Interface with display selection at boot time, software emulation of the RS-232-C/V.24 hardware configuration switches, documentation, and verification.

Pascal 3.1 also provides support for the two new processor boards. Unlike BASIC, Pascal programs must work with the HP-HIL keyboard. There is no HP 98203B keyboard emulation mode available.

Verification

If the objective is compatibility, then success can be measured by determining how easy it is to run Series 200 programs on a Series 300 Computer. Consequently, we looked at the results of running Series 200 application programs. A suite of BASIC and Pascal application packages was as-

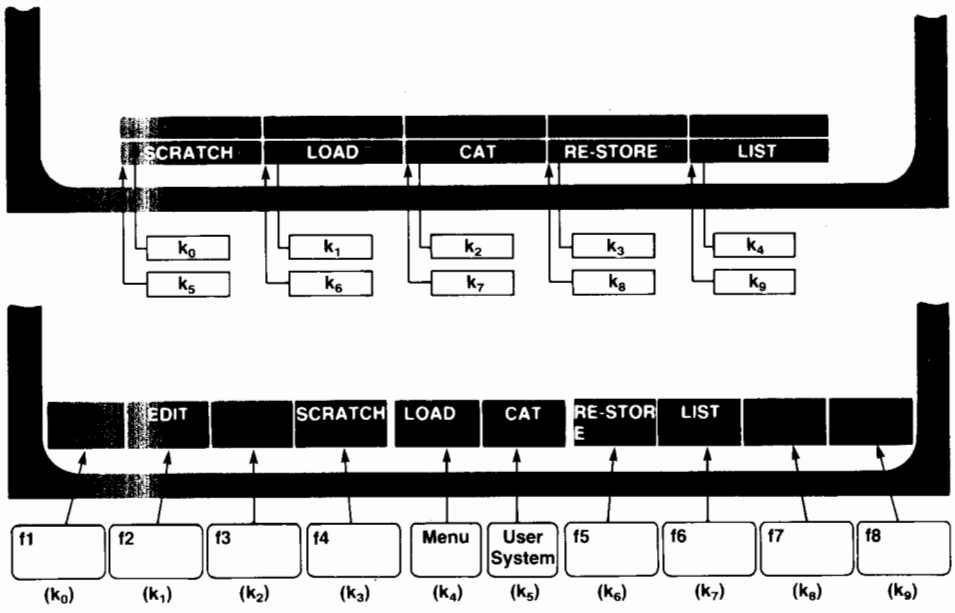


Fig. 4. *The HP 98203B (top) and HP-HIL HP 98203B keyboard emulation mode (bottom) displayed keylabel formats.*

Computer Museum

sembled and used to test the various compatibility solutions. This group of programs included Hewlett-Packard software packages from several different divisions. These programs were augmented by the applications of third-party software developers who were interested in observing the performance of their software running in compatibility mode. The test suite was run on both the Model 310 and the Model 320.

In addition to testing functionality, the test suite also identified porting problems which were subsequently resolved by the team or documented for user correction. The HP 98546A Display Compatibility Interface provided the required functionality for all of the application programs with which it was tested. The HP 98203B keyboard emulation mode provided compatibility for the test suite with limitations as previously noted in the keyboard section of this article. Testing of the separate alpha and graphics emulation mode identified source code problems that were limited to incorrectly specified device selectors in PLOTTER IS... statements. Verification of the other compatibility features demonstrated that we had achieved the desired functionality.

Acknowledgments

Deserving of recognition is the BASIC 4.0 project team headed by Tom Christian. The team members were Ales Fiala, Nancy Madonna, Steve Taylor, Jim Tear, and Jim Whalen. The team never lost sight of the compatibility goals that were so important to the Series 300 project. Dave Dahms should be recognized for his work on the HP 98546A Display Compatibility Interface and Mark Archuleta for his compatibility and porting documentation work.

Bibliography

K.Y. Kwinn, R.M. Hallissy, and R.E. Ison, "The 9826A/9836A Language Systems," *Hewlett-Packard Journal*, Vol. 33, no. 5, May 1982.

Authors

September 1986

Gilbert I. Sandberg

Gilbert Sandberg is an R&D section manager and has been the program manager for the HP 9000 Series 300 Computer. With HP since 1970, he has also contributed to the design of the HP 9817A Modular Computer, the HP 9816A Technical Computer, the HP 9920A Modular Computer, and the HP 9826A/S Technical Computer. He was born in Gettysburg, South Dakota and attended Brigham Young University. His BSEE degree was awarded in 1969 and his MEEE degree was awarded in 1970. A resident of Loveland, Colorado, Gilbert is married and has six children. He's a church leader and is active in local party politics. He enjoys basketball, skiing, and country and western music.

Ronald P. Dean

Ron Dean was born in Dearborn, Michigan and attended the University of Florida, earning a BSME degree in 1977. After coming to HP the same year he worked on a number of products in the HP 9000 Computer product line, including the HP 9000 Model 320 Computer. He's a coauthor of an *HP Journal* article on the HP 9000 Model 520 Computer. Ron and his wife and two children live in Fort Collins, Colorado. He enjoys chess, bridge, tennis, softball, bicycling, and skiing.

Daryl E. Knoblock

An R&D section manager at HP's Fort Collins Systems Division, Daryl Knoblock was responsible for CPU development for the HP 9000 Series 300 Computer. He also managed part of the development effort for the HP 3000 Series 30 and Series 44 Computers and for an HP-IB interface chip. Born in Detroit, Michigan, he attended the University of Michigan and received his BS degree in applied mathematics in 1969 and his MSE degree in computer engineering in 1971. He was a systems programmer at the University of Michigan before joining HP in 1972. A patent has resulted from his work on the HP-IB and he has written two articles on I/O channels and CMOS-on-sapphire technology. Daryl and his wife and three children are residents of Fort Collins, Colorado. He's active in his church and in community activities and enjoys skiing and camping.

John C. Keith

John Keith has been with HP since 1969 and is an R&D project manager at the Fort Collins Systems Division. He has worked on the HP 9815A and the HP 9845A Desktop Computers and has contributed to the design of HP 9000 Series 200 and Series 300 Computers. He's currently working on high-resolution display monitors. He is named coinventor on five patents related to desktop computers and is coauthor of a 1978 *HP Journal* article on the HP 9845A. Born in Des Moines, Iowa, John has a 1969 BSEE degree awarded by Kansas State University. He and his wife and four children live in Loveland, Colorado. He's active in his church and is an amateur radio operator (KD0GD).

Michael K. Bowen

With HP since 1979, Mike Bowen has a BSME degree awarded by the University of California at Davis in 1978. He is an R&D design engineer at HP's Technical Workstation Operation and has worked on the HP 9000 Series 500 and Series 300 Computers. His first HP assignment was in materials engineering. He's a coauthor of a 1984 *HP Journal* article on the Model 520 Computer. Born in England, Mike lives in Fort Collins, Colorado. He and his wife, who is also an HP engineer, have two daughters. He sails and races a Hobie cat, skis, and bicycles around Fort Collins.

9 — Processor Board Design

Nicholas P. Mati

Nick Mati is the designer of the MMU/RAM controller subsystem and the 2800 ALS gate array option for the HP 9000 Model 310 Computer. With HP since 1978, he also designed a RAM board for the HP 9845C Desktop Computer and the CPU board for the HP 9817A Modular Computer. He contributed to the design of the light pen for the HP 9845C and a portion of that work is the subject of a patent. Nick attended the University of Connecticut (BSEE 1976) and the University of Illinois (MSEE 1978). Born in Bridgeport, Connecticut, he now lives in Loveland, Colorado. He's an avid bicyclist and likes sailing.

Martin L. Speer

Born in Del Rio, Texas, Martin Speer studied electrical engineering at Texas A&M University. He completed work for his BS degree in 1977 and for his ME degree in 1979. After coming to HP the same year he contributed to the design of the HP 9816A Technical Computer and was responsible for the design of the processor for the HP 9000 Model 310 Computer. He's an IEEE member. Martin lives in Fort Collins, Colorado and likes photography, fishing, and camping.

12 — High-Performance SPU

Jonathan J. Rubinstein

With HP since 1979, Jon Rubinstein is an R&D engineer at HP's Fort Collins Systems Division. He was architect of the CPU for the HP 9000 Model 320 and Model 236 Computers. He's a member of the IEEE and the ACM and has written two papers related to the Motorola 68020 microprocessor and cache design. Jon is an alumnus of Cornell University. He completed work for his BSEE degree in 1978 and for his MEEE in 1979. He also has a 1985 MSCS degree awarded by Colorado State University.

17 — VLSI Circuits

David J. Hodge

Dave Hodge is a Colorado native who was born in Boulder and now lives in Loveland. He received his BSEE degree from the University of Colorado in 1975. With HP since 1979, he's a specialist in computer graphics and VLSI design. He has worked on graphics firmware and hardware for the HP 9845B Desktop Computer, on the video data pads and test circuits for the color map and DAC chip, and on the BITBLT algorithm and PLA for a display controller chip. Before coming to HP he designed milking machine controllers. Outside the office he's interested in model railroading, softball, and hiking. He recently enjoyed an extended trip to Australia and New Zealand.

Richard E. Warner

With HP since 1980, Richard Warner worked on the HP 9845B Desktop Computer before joining the original design team for the HP 9000 Series 300 Computer. He contributed to the product design, the IC for the display controller, and the graphics display boards. He's currently working on a future graphics product. Richard holds a BSME degree from the University of Colorado (1978) and an MSME degree from Stanford University (1980). Before coming to HP he designed precision machine tools and is coauthor of a research report on pneumatic isolation systems for such tools. He and his wife live in Fort Collins, Colorado and have one child. His outside interests include photography, bicycling, and history.

James A. Brokish

Jim Brokish has contributed to the design of several technical computers since joining HP in 1979. He designed the power supply for the HP 9915A Modular Computer and the alpha display subsystem for the HP 9836A Desktop Computer. He was also a production engineer for the HP 9836A, worked on a video display controller IC, and worked on the system bus design for the HP 9000 Series 300 Computer. He was born in Dickeyville, Wisconsin and attended the University of Wisconsin, receiving a BSEE degree in 1979. He's now working on an MSCS degree from Colorado State University through the HP Resident Fellowship Program. Jim lives in Fort Collins, Colorado, is married, and has one son. He and his son like spending time together in local parks. He's also a skier and white-water rafter.

22 — Software Compatibility

Rosemarie Palombo

With HP since 1985, Rose Palombo is a software engineer at the Fort Collins Systems Division. She has worked on BASIC for the HP 9000 Series 200 and 300 Computers and was responsible for the compatibility verification and documentation for the Series 300 Computer. She was born in Aliquippa, Pennsylvania and graduated in 1972 from Indiana University of Pennsylvania with a BS degree in mathematics. She also earned an MSCS degree from the University of Kentucky in 1985. She was a mathematics and data processing instructor before joining HP. A resident of Fort Collins, Colorado, she says she is a tennis fanatic, plays volleyball, and is learning to ski. She also enjoys travel, good films, literature, music, and computer games.

30 — Electronic Mail System

Robert A. Adams

A native of California, Bob Adams was born in Mountain View and attended California State University at Chico, receiving a BS degree in computer science in 1979. He worked for the National Aeronautics and Space Administration as a programmer/analyst during the same period. He joined HP in 1979 and has been a programmer and a project leader, and now manages messaging utilities for the Office Utilities Group. Bob lives in Mountain View with his wife, who is also a manager at HP. He enjoys chess and bridge.

Kristy Ward Swenson

With HP since 1982, Kristy Swenson has been the project leader for HP Desk implementation at HP's corporate headquarters, has coordinated an executive training program, and has been responsible for office automation user support, all within the Office Utilities

Group. She is currently a regional marketing engineer. Born in Portland, Oregon, she attended Scripps College, receiving a BA degree in Latin American studies in 1980. She also earned an MBA degree from the University of California at Los Angeles in 1982. Kristy lives in Half Moon Bay, California and is a volunteer both for Scripps College and for the American Field Service student exchange program. She's married and enjoys wine tasting, reading, cooking, and making beer.

Amy Tada Mueller

Amy Mueller was born in Honolulu, Hawaii and educated at Stanford University. She has a BS degree in mathematical sciences (1976) and an MS degree in computer science (1977). After joining HP in 1977, her first assignment was working on a manufacturing information system. She later moved to the Office Utilities Group and has been the manager responsible for the companywide implementation of HP Desk. Amy is married and lives in Sunnyvale, California. Outside of work she enjoys backpacking, cross-country skiing, softball, tennis, and golf.

Luis Hurtado-Sanchez

Luis Hurtado-Sanchez started at HP's corporate headquarters in 1973 as a programmer/analyst. He has contributed to the development, support, and management of office information systems, including HP Desk, and now manages the Office Utilities Group. He's the author of six articles and other publications and is a member of the Data Processing Management Association. He has a 1970 BS degree in chemistry from the University of California at Los Angeles and a 1973 MS degree in finance from the Massachusetts Institute of Technology. Luis was born in Guantanamo, Cuba and lives in Sunnyvale, California. His leisure interests include bicycling, bridge, films, and opera.

Rebecca A. Dahlberg

Born in Los Angeles, California, Rebecca Dahlberg studied Chinese history at the University of California at Berkeley (BA 1979) and management and marketing at Claremont College (MBA 1981). She joined HP's Neely Sales Region the same year, where she contributed to the implementation of an order processing system. After a transfer to corporate headquarters, she coordinated the companywide implementation of HP Desk and now supervises the HP Desk network support group. Rebecca lives in San Francisco, California and enjoys fly-fishing, bicycling, and travel.