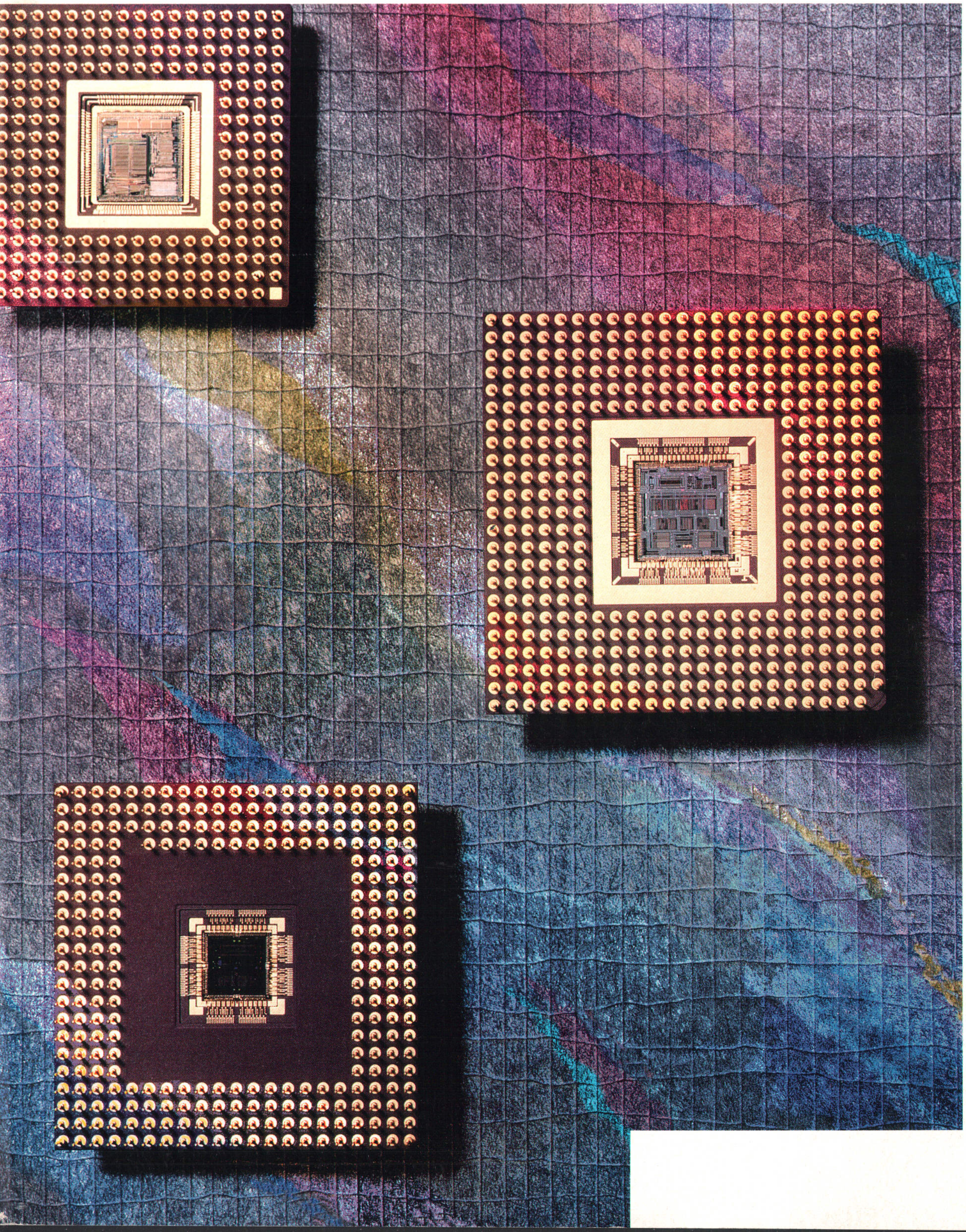
HEWLETT-PACKARD
JOURNAL

August 1992



HEWLETT
PACKARD

HEWLETT-PACKARD

JOURNAL

August 1992 Volume 43 • Number 4

Articles

Editor, Richard P. Dolan • **Associate Editor**, Charles L. Leath • **Publication Production Manager**, Susan E. Wright • **Illustration**, Renée D. Pighini
Typography/Layout, Cindy Rubin • **Test and Measurement Organization Liaison**, J. Michael Gospe

The *Hewlett-Packard Journal* is printed on recycled paper.

Departments

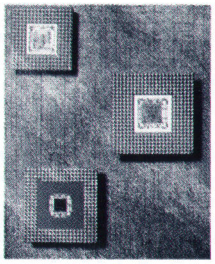
The Hewlett-Packard Journal is published bimonthly by the Hewlett-Packard Company to recognize technical contributions made by Hewlett-Packard (HP) personnel. While the information found in this publication is believed to be accurate, the Hewlett-Packard Company disclaims all warranties of merchantability and fitness for a particular purpose and all obligations and liabilities for damages, including but not limited to indirect, special, or consequential damages, attorney's and expert's fees, and court costs, arising out of or in connection with this publication.

Subscriptions: The Hewlett-Packard Journal is distributed free of charge to HP research, design and manufacturing engineering personnel, as well as to qualified non-HP individuals, libraries, and educational institutions. Please address subscription or change of address requests on printed letterhead (or include a business card) to the HP headquarters office in your country or to the HP address on the back cover. When submitting a change of address, please include your zip or postal code and a copy of your old label. Free subscriptions may not be available in all countries.

Submissions: Although articles in the Hewlett-Packard Journal are primarily authored by HP employees, articles from non-HP authors dealing with HP-related research or solutions to technical problems made possible by using HP equipment are also considered for publication. Please contact the Editor before submitting such articles. Also, the Hewlett-Packard Journal encourages technical discussions of the topics presented in recent articles and may publish letters expected to be of interest to readers. Letters should be brief, and are subject to editing by HP.

Please address inquiries, submissions, and requests to: Editor, Hewlett-Packard Journal, 3200 Hillview Avenue, Palo Alto, CA 94304 U.S.A.

In this Issue



As promised in this space in the June issue, this issue presents the hardware design of the HP Apollo 9000 Series 700 workstations. For readers who didn't see the June issue, the Series 700 is a group of computers that surprised the workstation world early last year with a huge increase in speed over what had been available, coupled with very competitive prices. Three factors combined to give the Series 700 machines their speed advantage. One is a new version of HP's PA-RISC architecture that includes enhancements specifically aimed at improving workstation performance (the PA stands for precision architecture and the RISC stands for reduced instruction set computing). The second factor in the new computers' speed is a set of VLSI (very large-scale integration) integrated circuit chips capable of operating at clock rates up to 66 MHz. Called PCX-S, the chipset includes a 577,000-transistor CPU, a 640,000-transistor floating-point coprocessor, and a 185,000-transistor memory and system bus controller. The third Series 700 speed enhancer (which was covered in the June issue) is a new version of the HP-UX operating system that takes advantage of the architectural enhancements and includes additional compiler optimizations to make programs run faster.

In the article on page 6, two Series 700 project managers give us an overview of the system design and performance of the HP 9000 Models 720, 730, and 750, the three original members of this product family. The PCX-S chipset is the subject of the article on page 12. The article on page 23 deals with the design of the all-important clock system that keeps everything working together, and the article on page 26 describes the built-in input/output subsystem, which provides several different standard interfaces, an audio system, and a real-time clock. Verification of the chip and system designs by computer simulation as described in the article on page 34 resulted in first silicon chips that were able to boot the operating system with no workarounds, a new record for HP. The package design of the original Series 700 workstations is covered in the article on page 43, and the manufacturing engineering, which included setting up a new surface mount process for printed circuit board production, is discussed in the article on page 49. There are now two entry level members of this product family in addition to the three original models. The system design and performance of the HP 9000 Models 705 and 710 workstations are described in the article on page 55.

The HP DeskJet and DeskWriter printers are low-cost thermal inkjet printers that deliver 300-dot-per-inch laser-quality printing on plain paper. While the original models printed only black, a color revolution has been taking place in the printer marketplace, creating a demand for an affordable, plain-paper color printer. Engineers at HP's InkJet Components Division were given the job of developing a color print cartridge for the DeskJet and DeskWriter printers, and their colleagues at the Vancouver Division were charged with adapting the printer platform to use the new cartridge. The color cartridge had to be interchangeable with the existing black print cartridge but contain three ink reservoirs instead of just one. Also, because the printers accept only one cartridge, the color cartridge couldn't depend on having a black cartridge around to help out, and had to be able to print a good black using only cyan, magenta, and yellow inks. As summarized in the article on page 64, these challenges were met, resulting in the new HP DeskWriter C and DeskJet 500C printers. The details of the design of the color cartridge and the inks are presented in the article on page 69. Manufacturing thermal inkjet print cartridges in high volume poses some formidable problems. The assembly process for the color cartridge is outlined in the article on page 77 (except for a few proprietary details). Many of the assembly steps depend on machine vision, as described in the article on page 87. Adhesives and encapsulants and the equipment to dispense them in tiny volumes to meet exacting requirements for the color print cartridge are the subjects of the article on page 84. To adapt the printer platforms to use the color cartridge, major changes were made in the software drivers. The driver for the DeskWriter C printer for Apple Macintosh computers is described in the article on page 93.

As the designers explain it in the article on page 103, "In a typical manufacturing environment, planners and buyers receive MRP (materials requirements planning) reports after running a regularly scheduled MRP program. These reports, called action reports, show suggested actions for work orders and purchase orders to meet future demand for inventory. Planners and buyers typically have to wade through mountains of paper reports to determine the suggestions they would like to act on. Then, they make any changes on paper and manually enter these changes into an MRP package. HP MRP Action Manager software automates this process." The HP MRP Action Manager software design is based on the client/server model. The planner or buyer uses a client machine (a personal computer running the HP NewWave environment) which communicates over a network with the server (an HP 3000 computer running the MPE/iX operating system).

R.P. Dolan
Editor

Cover

The PCX-S chipset for the HP Apollo 9000 Series 700 workstations includes a CPU, a floating-point coprocessor, and a memory and system bus controller. The chips implement the PA-RISC 1.1 architecture and run at 66 MHz.

What's Ahead

Featured in the October issue will be:

- The HP 71500A microwave transition analyzer, a new instrument architecture for pulsed RF measurements
- The HP 4980 Series Network Advisor protocol analyzers, an expert-system-based family of instruments for digital network troubleshooting
- HP Visual Engineering Environment, a software package that allows engineers to design test programs by drawing familiar block diagrams
- The HP E3560 digital performance monitoring and remote test system, a system for surveillance of the quality of European digital telecommunications networks
- The HP HDMP-1000 gigabit-link chipset, two easy-to-use chips that implement a 1.4-gigabaud serial digital data link.

Midrange PA-RISC Workstations with Price/Performance Leadership

The HP 9000 Models 720, 730, and 750 workstations achieve exceptional performance ratings on industry-standard benchmarks through a combination of a high CPU clock rate (up to 66 MHz) and tuning of the subsystem, compiler, and operating system designs. This article presents an overview of the hardware design.

by Andrew J. DeBaets and Kathleen M. Wheeler

Three HP 9000 Series 700 workstation computers were introduced in March 1991. These products were milestones in three respects. First, they were the first HP PA-RISC computers designed specifically as workstations (PA-RISC is HP's reduced instruction set computer architecture). Second, the product design cycle from inception to system release was executed in only 18 months. Third, they surprised the workstation world with their unprecedented combination of price and performance.

Fig. 1 shows the three workstation models. All are considered midrange computers, higher in performance than the low-end or entry-level models described in the article on page 55, which were introduced later.

The Model 720 is a 50-MHz desktop workstation. It is designed as a general-purpose workstation for CAD/CAE, scientific, and CASE applications.* It provides in a single enclosure a cached uniprocessor, onboard I/O, up to two internal SCSI 3.5-inch hard disks, an optional flexible disk drive, up to 128M bytes of RAM, an optional EISA (Extended Industry-Standard Architecture) interface, several graphics options, and a power supply.

The Model 730 is a 66-MHz general-purpose desktop workstation. With its additional performance, it makes a very competitive 3D graphics application engine. It is configured identically to the Model 720, except that the EISA slot is included in the base unit. The enclosures for the Models 720 and 730 are identical and are the result of new industrial design specifically for these products.

*CAD, CAE, and CASE stand for computer-aided design, engineering, and software engineering, respectively.

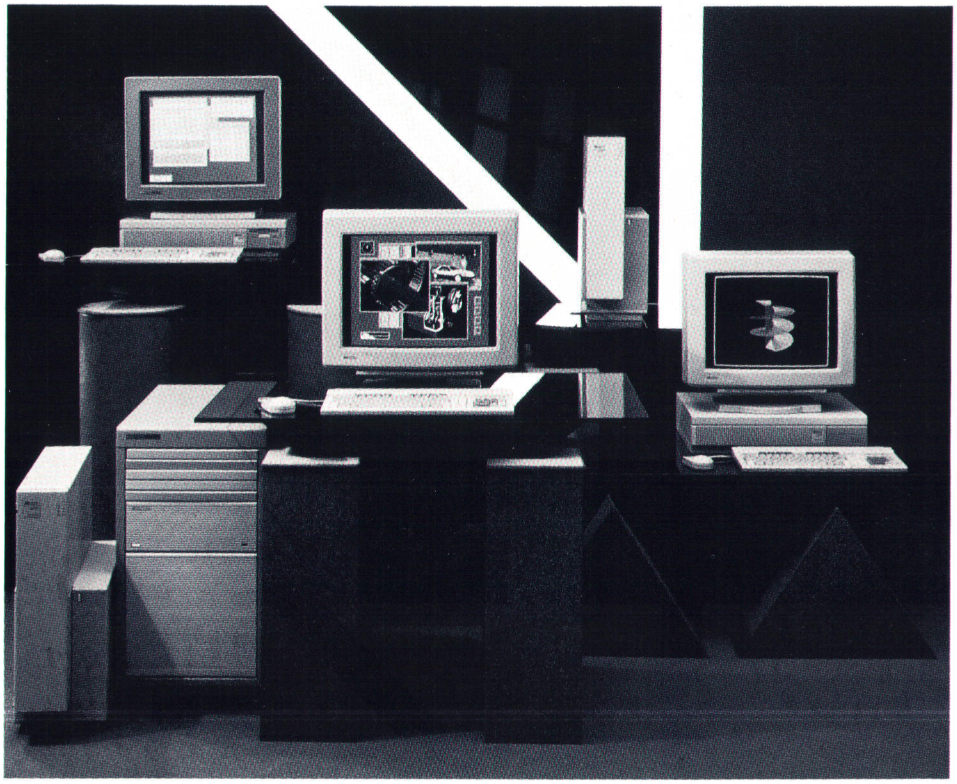


Fig. 1. HP 9000 Models 720, 730, and 750 workstations.

The Model 750 is a 66-MHz deskside workstation and file server designed as a high-performance 3D graphics workstation or as a workstation file server. Its enclosure is a vertical tower on a pedestal. The product design was leveraged from the HP 9000 Series 400 workstations. On a single motherboard the Model 750 provides all the functionality of the Model 730, including a cached uniprocessor, built-in I/O, and EISA interface. The memory cabability is increased to a maximum of 384M bytes. In addition, the Model 750 provides a total of four EISA slots, two graphics options, and up to four mass storage devices.

The design and implementation of these three workstations were undertaken with the following objectives, in order of priority:

- Time to market (schedule)
- Performance
- Product cost
- Development cost.

Quality was an overriding assumption of the project. The following sections provide an overview of the resulting system design, and accompanying articles in this issue give further technical details of the system components.

System Overview

The desktop Model 720 and 730 workstations are packaged in a new mechanical design. One of the key features of this package is the ease with which modules can be removed and replaced. This simplifies manufacturing and field support. The replaceable components include the CPU board, the graphics board, the system I/O board, the EISA expansion bus and EISA-compatible I/O cards, the disk drives, and the power supply. This makes it possible to define, design, and deliver component upgrade products quickly. These products can be offered as simple "board swap" upgrades to the installed base.

The deskside Model 750 workstation is packaged in a slightly modified HP Apollo 9000 Series 400 mechanical design. This existing package design offered sufficient volume, power, and cooling to contain all of the functionality of the Model 730 with increased expandability. Additional graphics, DRAM, mass storage, and EISA cards are supported within this larger package. The design of the Model 750 also allows motherboard "board swap" upgrades.

The industrial and product designs of the Model 720/730 and Model 750 packages are described in the article on page 43.

The Model 720 provides the following features:

- 50-MHz PA-RISC CPU
- 50-MHz PA-RISC 64-bit floating-point coprocessor
- 128K-byte instruction cache
- 256K-byte copyback data cache, 64 bits wide
- 8M to 128M bytes of error correcting memory
- 5-Mbyte/s SCSI II interface
- 10-Mbit/s IEEE 802.3 LAN interface
- Two RS-232 modem-control serial ports
- 350-kbyte/s bidirectional parallel port
- Audio tone generator
- HP-HIL device interface for keyboard, mouse, and other input devices
- One high-performance graphics slot, with a variety of graphics options
- One optional industry-standard EISA slot, with a variety of I/O cards
- Two integrated 3.5-inch 420-Mbyte fixed disk drives
- One optional 3.5-inch PC-compatible flexible disk drive (in place of one fixed drive).

The Model 730 increases performance and functionality over the Model 720 by adding the following features:

- 66-MHz PA-RISC CPU
- 66-MHz PA-RISC 64-bit floating-point coprocessor
- One standard EISA slot.

The Model 750 increases performance and functionality over the Model 720 by offering the following additional features:

- 66-MHz PA-RISC CPU
- 66-MHz PA-RISC 64-bit floating-point coprocessor
- 256K-byte instruction cache
- 16M to 384M bytes of error correcting memory
- Two high-performance graphics slots
- Four standard EISA slots
- Up to four mass storage devices. Two full-height 5.25-inch devices can be supported with two additional half bays that support either 3.5-inch devices or half-height 5.25-inch devices. Up to three of these bays can support removable media, and the two half bays can also be converted to support a third full-height 5.25-inch mechanism.

A configuration with entry-level graphics and EISA consists of four major printed circuit board assemblies in the Models 720 and 730 and two major printed circuit board assemblies in the Model 750. These boards contain six fully custom VLSI chips and three semicustom VLSI chips developed to implement the features listed above.

To meet the aggressive schedule objectives, the quality of the nine core chips needed to be very high. Management set a very clear expectation that the chips would be "right the first time" and provided resources to verify the designs thoroughly before tape release. As a result of the hard work of the design and verification teams, there were a number of significant successes. The first CPU chip booted the HP-UX* operating system four minutes after it was installed in a processor board. All of the hardware was released for volume production only three months after the final VLSI chip arrived—and this VLSI chip was the heart of the system. A number of chips were released to production without changes. These successes allowed the aggressive schedule to be met. Additional details regarding the chips' design and verification can be found in the articles on pages 12 and 34, respectively.

In addition to the hardware, a common set of firmware was implemented on the three products. This firmware provides a number of functions, including self-test, booting, and console handling. A description of this firmware is provided on page 9.

Processor and Memory Subsystem

Fig. 2 is a block diagram of the processor and memory subsystem. The processor and memory designs for the Models 720, 730, and 750 are identical except for frequency. The Models 720 and 730 implement the functionality on a single printed circuit board. The Model 750 implements the same design, with I/O and EISA interface, on a motherboard twice the size of the Model 720/730 processor board.

The Model 720 CPU clock frequency is 50 MHz and the Model 730/750 CPU clock frequency is 66 MHz. Therefore, one instruction is executed every 20 ns and 15 ns, respectively. The clock system for the entire system is implemented with ECL logic and a crystal oscillator running at twice the CPU frequency. Very careful attention was paid in the design of the printed circuit boards to ensure that the clocks are synchronous at the key components of the system. Delays were calculated and adjusted in the clock trace lengths. The article on page 23 describes the design details of the clock system.

The uniprocessor design uses a VLSI CPU chip whose architecture is referred to as PCX-S (see article, page 12). The PCX-S chip is fabricated in HP's CMOS26 process and is housed in a 408-pin ceramic PGA (pin-grid array). The processor architecture is 32 bits wide. Key features include cache controllers onboard, separate TLBs (translation lookaside buffers) for data and instruction memory management, a five-stage pipeline, and test and diagnostics support.

This is the first PA-RISC processor design optimized for workstation requirements. The performance modeling looked at key requirements for the workstation marketplace. Both the floating-point coprocessor and the caches are optimized for workstation applications. The PCX-S CPU and floating-point coprocessor implement the PA-RISC 1.1 architecture, which includes enhancements to the original PA-RISC architecture resulting from a study of workstation requirements by HP and Apollo engineers. Therefore, this design represents many of the best practices of both HP and Apollo, now both represented by the HP Workstation Business Unit.

The cache is implemented as a single-level, off-chip cache memory (as opposed to many industry workstations, which implement a primary cache on the processor chip and a secondary cache off-chip). It is configured as 256K bytes of data cache and 128K bytes of instruction cache for the Models

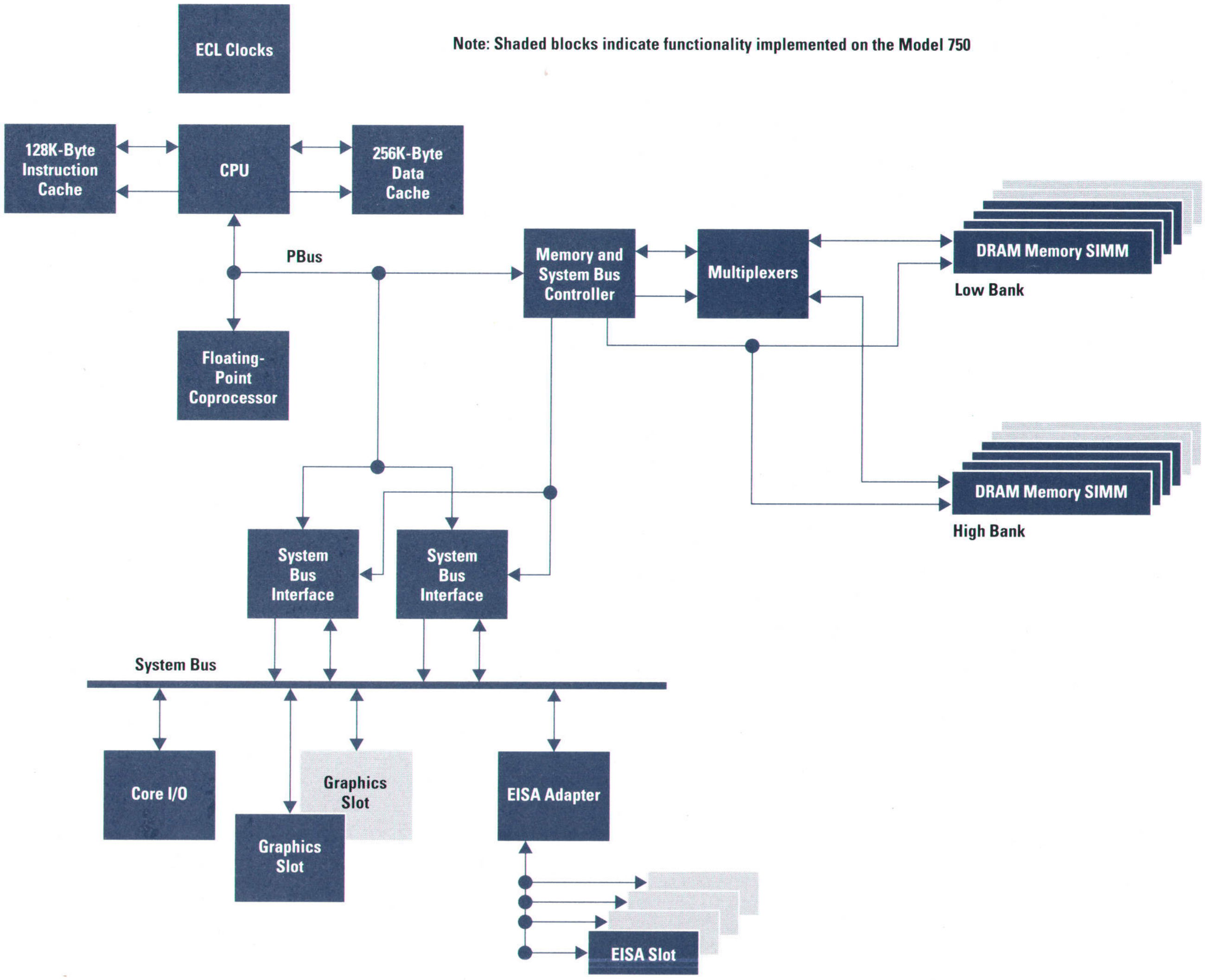


Fig. 2. Series 700 workstation processor and memory subsystem block diagram.

720 and 730. The Model 750 is configured with 256K bytes of both instruction cache and data cache. The cache is directly mapped and has parity for error detection. The instruction cache is 32 bits wide and is organized as lines of four double words (32 bytes). The data cache is 64 bits wide and is also organized as lines of four double words (32 bytes). Parity is generated for each 32-bit word in both caches. Associated with each line in the caches is a tag, which provides 20 bits of addressing information and an additional two bits of status. 32K × 8-bit and 8K × 8-bit SRAMs are used in the Models 720 and 730, while the Model 750 uses 64K × 4-bit and 8K × 8-bit SRAMS. The SRAMs used to implement the instruction and data caches have an access time of 15 ns for the Model 720 and 12 ns for the Models 730 and 750. The fast access rates are required so that instructions and data can be input on every clock cycle.

The floating-point coprocessor was a joint design effort of HP and Texas Instruments, Inc. It has a 64-bit internal architecture and has direct access to the instruction and data caches. It is housed in a 204-pin ceramic PGA. This chip is also the result of careful implementation of previously unimplemented features to optimize workstation performance. These enhancements, described in PA-RISC 1.1 specifications, expand the floating-point instruction set.

The processor communicates with the memory and I/O subsystems via the PBus (processor bus). The 32-bit-wide PBus multiplexes the address and data lines. The transfer of data to memory and the rest of the system is controlled by a single-chip memory and system bus controller, fabricated in CMOS26 technology and housed in a 272-pin ceramic PGA. To optimize the efficiency of memory transfers, the width of the memory array was set at 144 bits (128 data bits and 16 bits of error correction code). This scheme was most effectively implemented with available hardware by creating two banks of user-installable SIMM (single inline memory module) cards.* Each card is 72 bits wide. The current SIMM implementation uses 1M-bit DRAMs (8M bytes/SIMM) and 4M-bit DRAMs (16M bytes/SIMM). The theoretical maximum memory allowed in the system using 16M-bit DRAM technology is 1G bytes. The systems implement a fixed number of slots for SIMMs (8 slots in the Models 720 and 730 and 12 slots in the Model 750), so the actual maximum memory is less than 1G bytes. Each SIMM uses a proprietary memory interface chip to buffer the addressing of the two-bank implementation.

The memory and system bus controller chip is also a bus converter and communicates with the system bus through two proprietary system bus interface chips. The system bus is a high-bandwidth bus capable of operation up to 33 MHz (the maximum data transfer rate is 33 Mwords/s). Since the PBus is a multiplexed bus and the system bus has separate lines for the address and data signals, the primary function of the system bus interface chips is to demultiplex the PBus to the system bus and vice versa. The built-in I/O, graphics, and EISA interface all attach to the system bus.

* The memory SIMMs are a proprietary design and do not resemble the industry-standard memory SIMM.

HP 9000 Series 700 Workstation Firmware

The firmware for the Series 700 workstations described in the accompanying article includes what is formally known as processor dependent code (PDC) and I/O dependent code (IODC). It is a mixture of C and PA-RISC assembly language code, and it resides in two 128K-byte EPROMs on the system board. The firmware bridges the gap between the machine hardware and the operating system software by providing basic machine-specific test, initialization, and bootstrapping services.

The Series 700 firmware is a close cousin of the Series 800 firmware. The overall design and architected software interface (with a few additions) have been preserved to minimize the impact on the primary users of PDC, which include the diagnostics, the operating system loaders, and the operating system itself.

The special contributions of the Series 700 firmware are:

- The addition of several new procedural entry points
- A revised user interface suitable for PA-RISC workstations
- A strategy for booting through the EISA interface.

For the Series 700, the primary user of the system may also function as the system administrator. Several new features were added to the firmware user interface to make system administration tasks at the PDC level easy and intuitive. Default boot and console devices can be specified through friendly module names (e.g., scsi, eisa, graphics) rather than the traditional HP NIO/CIO path notation. Another special contribution of the Series 700 firmware functionality is an optional I/O scan for potentially bootable devices.

Several new procedural entry points were developed for the Series 700 to preserve the external PA-RISC view of the I/O subsystem as seen through the PDC procedural interface. Unlike Hewlett-Packard's proprietary CIO and NIO, the Series 700 I/O modules do not reside at fixed address intervals (slots) in I/O space. That is, Series 700 I/O modules do not necessarily reside on successive 4K boundaries. For the Series 800, the operating system uses this fixed position knowledge to convert a representation of an I/O module (in the traditional bus converter, physical module, device dependent layer notation) to the corresponding start address. Many of the PDC entry points that provide auxiliary information about I/O modules require as an input parameter this hard physical address. Rather than change the PDC interface (and force all callers of PDC to change also), a special new entry point, PDC_MEMORY_MAP, provides the necessary translation of a logical designation of an I/O module to the hard physical address. Thus, software need deal only with the logical mapping of I/O modules and need not derive or have a priori knowledge of the actual physical address of the module.

A special strategy was derived to facilitate a boot through the EISA interface, independent of the EISA configuration file information inherent in most PC EISA boot models. This strategy involved a definition for PA-RISC EISA IODC (consisting of architected IODC with a special header and an extra entry point) for the card of interest, a new PDC entry point, PDC_EISA, which provides machine-dependent information for the EISA IODC, and a cooperative scheme between the PDC and the operating system to allow a "cold" boot through EISA (before having run the on-line EISA configuration utility). This strategy is not exclusive to the Series 700.

The Series 700 firmware was developed on HP 9000 Model 840 and 835 computers using the PA-RISC simulator with extensions for I/O modeling. A complete simulated boot of the firmware from power-up to initial program load was achieved before arrival of first silicon. Simulation greatly reduced the time needed to bring up the firmware on the actual hardware.

Deborah A. Savage
Software Development Engineer
Workstation Systems Division

Graphics

The Model 720/730/750 system design is optimized for graphics performance. Workstation application performance was a fundamental objective of the project and graphics is a key component. As a result, special features have been added to the memory and system bus controller chip, and the system bus specification is customized for graphics applications.

The memory and system bus controller implements a core set of functionality to optimize and thereby accelerate graphics performance on the system. The two main functions are interpolation of Z buffer and frame buffer data. Another core function is block move transfers between memory and the graphics hardware.

The drive to create a "one box on the desktop" workstation also led to the creation of a low-cost, one-board, graphics subsystem, which occupies the graphics slot configured in the Models 720, 730, and 750. The performance of this graphics option (CRX) exceeded expectations at 1.15 million 2D or 3D vectors per second (8 bit color). An article on this subsystem is planned for a future issue of this journal.

Built-in I/O

One of the objectives for the I/O subsystem was to use an industry-standard I/O bus. A number of alternatives were considered, and a brief, spirited, period of discussion and analysis led to the selection of a bus from the PC world: EISA (Extended Industry Standard Architecture). A bus adapter design, using Intel's original EISA chipset, was created to connect the EISA bus to the high-performance system bus.

This selection of an industry standard bus (EISA) drove a strategy to make the PA-RISC version of the HP-UX operating system conform more closely to industry standards. This was done by modifying the operating system's I/O subsystem to provide a software interface for drivers that is much more like other vendors' systems.

The functionality provided on the EISA bus now includes IEEE 802.3 LAN, FDDI LAN, IBM token ring LAN, X.25, SNA, fast/differential SCSI, serial multiplexers, and the HP-IB (IEEE 488, IEC 625). A number of these cards and drivers are provided by independent vendors. An article on the EISA card design is planned for a future issue of this journal.

The core I/O subsystem, which is configured on the system I/O board and is independent of the EISA controller card, was selected to meet 80% to 90% of workstation users' needs without additional I/O cards. The system I/O board provides two serial ports, a bidirectional parallel port, an IEEE 802.3 LAN interface, an HP-HIL keyboard/mouse interface, and an SCSI interface. This robust offering of integrated I/O meets a wide variety of needs. The standard SCSI interface allows support for disks, DAT drives, CD-ROMs, and a PC-compatible flexible disk drive.

The core I/O subsystem also benefited from the move towards a more open operating system. This competitive focus from the operating system and networking teams provided driver support for a system I/O board design based on the latest high-performance industry VLSI controllers. The key components are new SCSI controllers from NCR (53C700, used on the system I/O board, and 53C710, used on the EISA card) and a new LAN controller from Intel (82596). These off-the-shelf devices interface efficiently and inexpensively to the high-performance system bus.

The combination of a high-performance CPU, focused I/O driver development, high-performance industry I/O controllers, and an industry-standard bus has provided impressive I/O performance. The Models 720, 730, and 750 provide sequential disk performance eight times greater than previous-generation products. NFS benchmarking (which uses networking and disk I/O) shows that configurations of the Model 750 can outperform dedicated NFS file servers. Additional details regarding the design of the core I/O subsystem can be found in the article on page 26.

Performance

The performance of the Models 720, 730 and 750 is the result of a systems approach to creating a product that addresses the needs of workstation applications. Although the system clock rate is a factor in the performance numbers for the products, it is not the whole story. Directed design and tuning of the operating system, compilers, and subsystems, as discussed in this issue and the June 1992 issue of this journal, resulted in the industry leading performance numbers of the Series 700 workstations.

One of the workstation industry's standard measures of performance is a set of benchmarks called SPECmarks from the Systems Performance Evaluation Cooperative. This set of programs is run on the target machine and the individual results are combined to derive an overall SPECmark for the target machine. Table I shows the HP SPEC results for power desktop products as of February 10, 1992, along with the more basic MIPS (millions of instructions per second) and MFLOPS (millions of floating-point operations per second) ratings. Four of the SPEC benchmarks have little floating-point activity and six are floating-point-intensive. The performance ratings for these categories are listed separately as SPECint and SPECfp, respectively.

Table I
HP 9000 Series 700 SPECmark Performance

		Model 720	Model 730	Model 750
SPECmark	(rel. 1)	59.5	76.8	77.5
SPECint	(rel. 1)	39.5	51.2	51.5
SPECfp	(rel. 1)	78.3	100.6	101.6
SPECint92	(rel. 2)	36.4	47.8	48.1
SPECfp92	(rel. 2)	58.2	75.4	75.0
MIPS		57	76	76
MFLOPS (double)		17.9	23.7	23.7

Other performance measures used to compare workstations are those tied to graphics performance. Two such metrics measure those functions contributing to X Window performance (principally 2D) and the manipulation of forms in 2D and 3D graphics applications. The metrics for the CRX option of the Models 720, 730, and 750 workstations are shown in Table II.

Table II
HP 9000 Series 700CRX Graphics Performance

	Model 720CRX	**Model 730CRX**	**Model 750CRX**
X11perf 1.2 Overall Operation	8,474	10,904	10,904
X11perf 1.2 General Graphics	8,079	10,297	10,297
3D Vector Performance	**Thousands of 3D Vectors per Second**		
Sustained	820	1,090	1,090
Peak	1,160	1,180	1,180

Acknowledgments

The authors wish to thank Steve Foster, R&D section manager, and Russ Sparks, R&D lab manager, for their leadership and support during the development of the Series 700 products. Special thanks to the engineers of the Series 700 project, whose hard work and dedication were key to the products' success. Additional thanks are due people throughout HP and Apollo for their contributions. Finally we must thank Denny Georg for providing the competitive focus and organizational value set to allow these products to be built.

VLSI Circuits for Low-End and Midrange PA-RISC Computers

The major VLSI chips for the HP 9000 Series 700 workstations include a central processing unit with 577,000 transistors, a floating-point coprocessor with 640,000 transistors, and a memory and input/output controller with 185,000 transistors.

by Craig A. Gleason, Leith Johnson, Steven T. Mangelsdorf, Thomas O. Meyer, and Mark A. Forsyth

Processor and memory designs for today's low and midrange engineering workstations and multiuser systems are constrained by many different, sometimes conflicting, requirements. High performance is critical for very large and computation-intensive user applications. Increasingly, customers expect more cost-effective solutions, requiring the right balance between cost and performance. In addition, product requirements dictate that processor and memory subsystems be compact, consume little power, and require a minimum amount of support hardware. The VLSI (very large-scale integration) circuits described in this paper were designed to meet these requirements and to have the flexibility to be used in a number of HP PA-RISC computers, including members of the HP 9000 Series 700 and 800 families and the HP 3000 Series 900 family.

Overview

The core processor and memory subsystem for these products is shown in Fig. 1. The major VLSI chips include a central processing unit (CPU), a floating-point coprocessor (FPC), and a memory and system bus controller (MSBC). Collectively, these chips are called the PCX-S chipset. Key features of these devices are shown below:

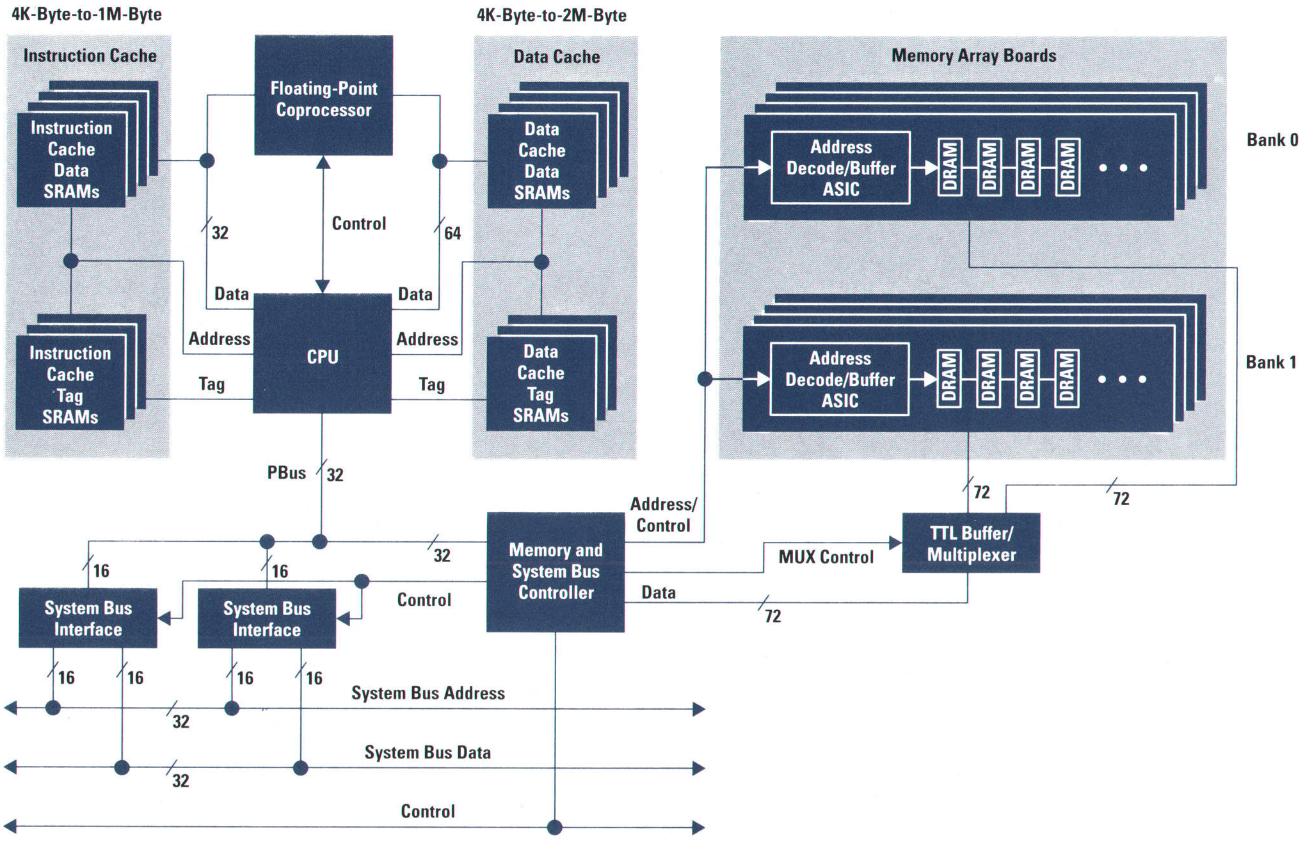


Fig. 1. Core CPU, cache, floating-point coprocessor, and memory subsystem block diagram.

	CPU	FPC	MSBC
Technology	HP CMOS26 1.0 µm	TI EPIC-2 0.8 µm	HP CMOS26 1.0 µm
Number of Transistors	577,000	640,000	185,000
Die Size	14.0 by 14.0 mm	13.0 by 13.0 mm	9.5 by 9.5 mm
Package	408 CPGA	207 CPGA	272 CPGA
Maximum Operating Frequency	66 MHz	66 MHz	66 MHz

CPU and Cache Design

All processor functions except for floating-point execution and cache memory storage are performed by the CPU. This chip interfaces to both the FPC and the cache memory arrays. It also includes an interface to the memory and I/O subsystem. The CPU implements HP's PA-RISC version 1.1 architecture, which includes extensions to the original PA-RISC architecture focused on enhancing workstation performance.

Because time to market was a key design focus, many features of the PCX-S CPU design were drawn from the PCX chipset used in HP's high-performance multiprocessing systems.[1] The design was modified to accommodate a lower-cost system more closely aligned with the needs of the engineering workstation and low-end multiuser computer markets. Where the previous design employed two-way set-associative cache memories requiring VLSI support chips, the new chip is designed to interface directly to TTL-I/O SRAM chips. Tuning of the system to operate at higher frequencies with faster SRAMs was also facilitated through this change. The on-chip TLB was increased in size and made fully associative to allow the removal of the second-level TLB in the original design. New features and instructions (including the PA-RISC 1.1 extensions) were added to improve user interface performance. All proposed changes were prioritized according to their schedule impact and potential performance improvement, and those that offered the best ratio of performance improvement to design effort were chosen. Because of this schedule-driven choice of features and the high leverage from the previous chip, only fourteen months were required to reach first silicon.

Physically, the CPU is divided into seven major blocks: two data paths, two memory management units, a cache control programmable logic array (PLA), a pipeline control PLA, and an input/output block containing the various interfaces to cache memory storage and the memory and system bus controller. Several small control and logic blocks composed of standard cells and custom logic are also included. The design methodology used was "structured custom," where blocks are composed largely of library cells but are often hand-tiled and include custom circuits to enhance performance.

Phase clocks are used to divide each processor cycle into two halves or phases. The CPU includes three clock buffers, each fed by differential, ECL-level sync inputs (see article, page 23). The outputs of all three buffers are connected on-chip to reduce clock distribution skew. Each buffer outputs two complementary signals, CK1 and CK2, representing the two clock phases.

Most registers on the chip are constructed with transparent latches, which update their outputs while a clock input is true and hold the previous value while the clock input is false. These latches are more easily constructed in CMOS than edge-triggered latches. Another advantage of using the transparent latches is that in critical timing paths, the data tends to flow from one latch to another without waiting for clock edges. This technique allows the system to run at higher frequencies by spreading out timing budgets over multiple phases, rather than being limited by the slowest one-phase budget (as in an edge-triggered system). To avoid race conditions in queues of transparent latches because of overlap and distribution skew between CK1 and CK2, local nonoverlapping clocks are created (see Fig. 2).

Each integer instruction executed by the machine is processed by the CPU's five-stage pipeline. Five instructions can be processed concurrently, one in each of the various stages of execution. An instruction begins with a fetch from the instruction cache followed by a stage to decode the instruction. Branch and arithmetic stages follow, in which all arithmetic and logical operations are done, conditions are set, and branch targets are calculated. In the fifth stage the instruction is completed by writing its result to a general register.

Many architected registers are housed in the two data paths. A stack of 32 general-purpose registers acts as the first level of data storage. All integer operations use these registers for fast access to data. Load and store instructions are provided to move data between the general registers and memory. Eight space registers are used to hold the 16-bit space identifier portion of the 48-bit segmented virtual address. Various control registers are included for interrupt handling, temporary storage, data protection, real-time operation, and other tasks. All of these resources are located in the two data paths.

An interesting feature of the general registers is the addition of seven shadow registers to aid in state saving and restoring. Each shadow register is located next to a companion general register. Anytime a trap occurs, the operating system must save the current state of the machine so that it can be properly restored once control is returned to the user's

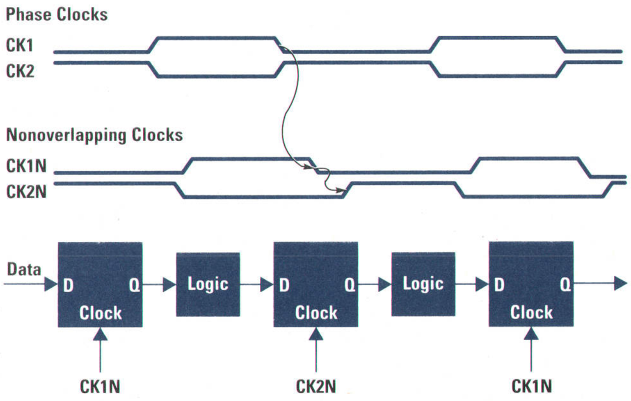


Fig. 2. Local nonoverlapping clocks derived from the basic phase clocks avoid race conditions in chains of transparent latches.

code. Normally this would require many instructions to transfer the data from the general registers to temporary storage or memory (and vice versa on returning control). All seven shadow registers can be updated in parallel while handling a trap, and likewise, the contents of the corresponding general registers can be restored simultaneously from their shadow registers on returning from the interrupt.

Execution Unit. At the heart of the CPU is the execution unit, which handles arithmetic and data manipulation operations. It consists of an arithmetic/logic unit (ALU), a shift/merge unit (SMU) to perform shifting and bit-field extraction operations, and various registers to feed these circuits with data and collect the results. Other processing resources include an adder for calculating branch target addresses.

The ALU is a high-speed, general-purpose arithmetic unit performing addition, subtraction, and Boolean logic operations. It includes logic to set conditions based on the results of instructions. These conditions can be tested to control branches, traps, and nullification. The ALU is fed by two input buses. Most often the input buses are driven with data from the general registers, but in some cases the required data has not yet been written to the general registers and exists only in various pipeline registers representing the results of previous ALU operations or loads. In these cases the data is bypassed around the general registers to the ALU. Using bypassing, an instruction can use the result of the previous instruction as one of its operands without causing the CPU to stall. A load followed by an immediate use of the load target as an operand will cause a single cycle penalty.

Because branches are an important component of any instruction stream, an effective branch prediction algorithm is implemented. On any program-counter-relative branch (where a positive or negative displacement is added to the current program counter to calculate the branch target), the CPU predicts that backward branches will be taken and forward branches will be untaken. Intuitively this is a good choice since most backward branches will be in multiple iterations of loops, while forward branches might be encountered in case statements, for example. Correctly predicted branches incur no pipeline penalty. An incorrectly predicted branch will cause a one-cycle delay.

Cache Memories. No on-chip cache memory is provided, but the CPU includes full data, tag, and address interfaces to separate instruction and data cache memories. Each of these is composed of industry-standard, asynchronous, TTL-I/O static RAMs (SRAMs), providing a wide variety of cache speed and size options. Included in the cache interfaces are tag compare circuitry, parity encoding and decoding logic, store byte merging logic, and cache fill and copyback circuits.

Given that SRAM technology is constantly improving, the system is designed to make the cache read cycle its critical path. With the 12-ns 256K-bit SRAM parts available in 1991 it is possible to build a 66-MHz system with up to 256K bytes (each) of instruction and data caches. However, with the cache access path limiting system frequency, a growth path is left open to higher frequencies simply by substituting 10-ns, 8-ns, and even faster SRAMs. An additional advantage of the SRAM-limited design is the opportunity to reduce the operating frequency and save cost by using a lower-speed SRAM part (as in the 35-MHz HP 9000 Model 705 workstation and the 32-MHz HP 9000 Model 807 multiuser system).

The cache memories are optimized for read timing, since writes occur less frequently in most code (remember, every instruction fetch involves an instruction cache read). Data cache fills take two cycles per double word, and data cache stores take three cycles, using a read-modify-write operation. Compiler scheduling is used to eliminate penalties because of conflicting loads and stores on the data cache. In cases where this is not possible, a store can result in a one-or-two-cycle penalty if either of the next two instructions is a load.

While filling the cache on an instruction cache miss operation, the CPU can execute the code as it is copied into the cache. This "cache streaming" effectively reduces the miss penalty by up to eight cycles. A similar algorithm is used on the data cache.

To provide the fastest possible cache access, a virtual address is used to index the SRAMs (this allows the CPU to access the cache before it has done a virtual-to-real address translation). The tag contains the real address of the data stored in each 32-byte cache line, as well as a "dirty" bit to indicate that the location has been modified and a "private" bit for multiprocessor support. Each processor must obtain a private copy of a cache location before modifying that location in the cache. The instruction and data caches are direct-mapped, again to speed up the cache access and to reduce the CPU pin count. To reduce bus traffic in multiprocessor systems, a write-back memory update policy is implemented.

A cache access is initiated by driving an address to the SRAMs (see Fig. 3). Data will return (asynchronously) roughly one cycle later, depending on the speed of the address drive and the SRAM access time. The data is collected by transparent latch receivers on the CPU. The timing of the latch clock for the incoming data is critical. If the latch clock falls too early, data will not be valid; too late, and a fast SRAM can change its data before the receiver input is shut off. This latch clock is created by inverting the CPU clock that initiates the cache access and delaying it by

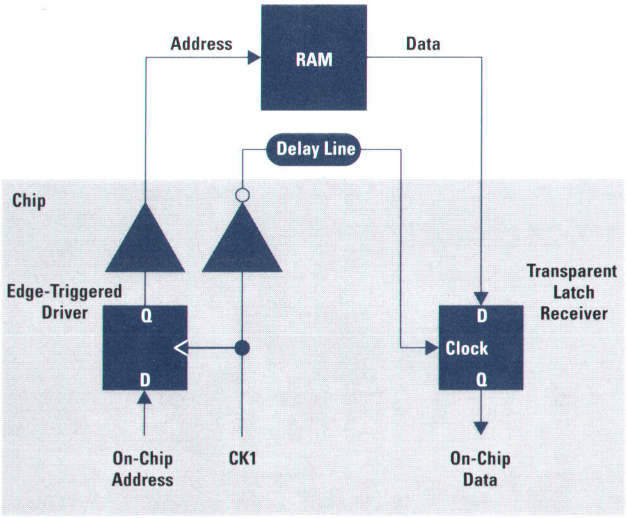


Fig. 3. Cache timing circuit block diagram.

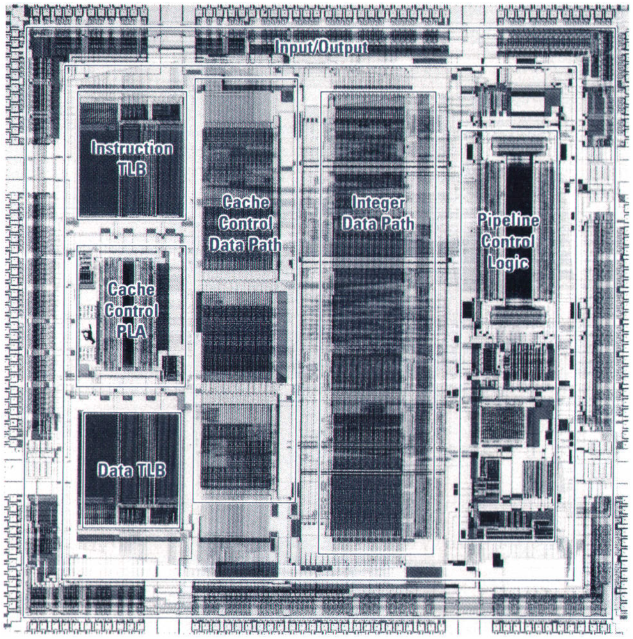


Fig. 4. CPU chip.

means of a long printed circuit board trace. Simulations and characterization data were used to choose the correct latch clock timing for reliable, high-performance operation.

Translation Lookaside Buffers. Virtual-to-physical memory translation is performed by the instruction and data translation lookaside buffers (TLB). A TLB contains 96 fully associative entries, each mapping a 4K-byte page of memory. Each TLB also implements four "super-TLB," or block entries. These are used by software to map large contiguous areas of memory (up to 16M bytes), which would normally require many single-page TLB entries. One common use of the block entries is in mapping the graphics frame buffer.

On an instruction fetch a virtual address is issued to the instruction TLB. Each entry is checked to see if it contains the real translation and protection information for that address. In the case of a match, the real address is driven by the instruction TLB and compared with the real address of the incoming instruction from the cache tag. A similar process is used in the data TLB in the case of a load or store.

To perform a fully associative access in a 66-MHz cycle, the TLBs make use of several interesting circuits. Typically, a fully associative memory is composed of a content-addressable memory (CAM), which holds the "address" of each entry, and a random-access memory (RAM), which holds the data associated with that address. In the case of the TLB, the CAM holds the virtual address of a page (space identifier and virtual page number), and the RAM holds the real page number, protection identifier, access rights, and other information specific to the entry. The virtual address for the current instruction or data access is driven to the CAM and compared with the address contained in the entry. If there is a match, the CAM causes the associated RAM entry to drive its data out of the TLB.

The TLB is required to complete its access in one phase (half cycle) so that the real address can be driven to the cache interface before the data or instruction returns from the corresponding cache access. Because of the two-part nature of the access (CAM/RAM), it was necessary to create a special timing clock to complete the CAM access and initiate the RAM access. This is done by building a dummy CAM entry that always "hits" and fires its match line. Care is taken in the chip layout to ensure that this entry always evaluates more slowly than the others. The dummy match line is buffered to create the midphase clock that will initiate the RAM access. If there is a match in any of the CAM entries, the clock will cause the select line for its RAM entry to be driven.

Entry replacement in the TLB is handled by an algorithm that provides performance similar to a least recently used policy while requiring only a fraction of the overhead. Only one bit per entry is used. The first entry whose replacement bit is zero is chosen for replacement, and following the operation its bit is set to one. If all replacement bits are ones the first entry is chosen and all other replacement bits are set to zero.

HP's high-performance CMOS26 technology was chosen for the CPU fabrication. CMOS26 offers three levels of aluminum interconnect and low-resistance silicided polysilicon and diffusion layers. Typical effective gate lengths are 0.85 micrometers for n-channel transistors and 0.88 micrometers for p-channel transistors. Fig. 4 is a photomicrograph of the CPU chip. The chip is packaged in a 408-pin multilayer PGA (pin-grid array) originally designed to support the PCX chipset. The PGA was modified for the PCX-S chip to reduce cost and reflect lower power dissipation (8 watts at 66 MHz). Multiple power and ground planes are used in the package to reduce inductance and support the high-speed switching of over 200 outputs. Power supply bypass capacitors are attached directly to the ceramic substrate of the PGA to minimize noise.

Floating-Point Coprocessor

The FPC chip contains the hardware for executing floating-point instructions. Fig. 5 is a block diagram of this chip and Fig. 6 is a photomicrograph. The FPC chip includes an FALU data path, an FMPY data path, a large register file, ten instruction pipeline registers, and a gate array for control logic. The basic design of this chip has been described in a previous paper.[2]

The FPC is tightly coupled to the CPU and caches. It connects directly to the instruction cache SRAMs, so it can decode instructions as soon as they are fetched rather than waiting for the CPU to tell it what operation needs to be performed. To maintain synchronism, the FPC and the CPU keep identical copies of the processor pipeline. The FPC's direct connection to the data cache SRAMs helps minimize latency on load and store operations.

Maximizing concurrent execution of instructions was an important goal in the FPC design. There are two fully independent functional units: the FALU and the FMPY. The FALU can be performing add or subtract operations while the FMPY performs unrelated multiplies, divides, and square roots. To minimize processor pipeline stalls, the FPC has a two-entry queue for instructions that cannot execute immediately because of data dependency or resource contention. Up to ten instructions can be simultaneously pending inside the FPC.

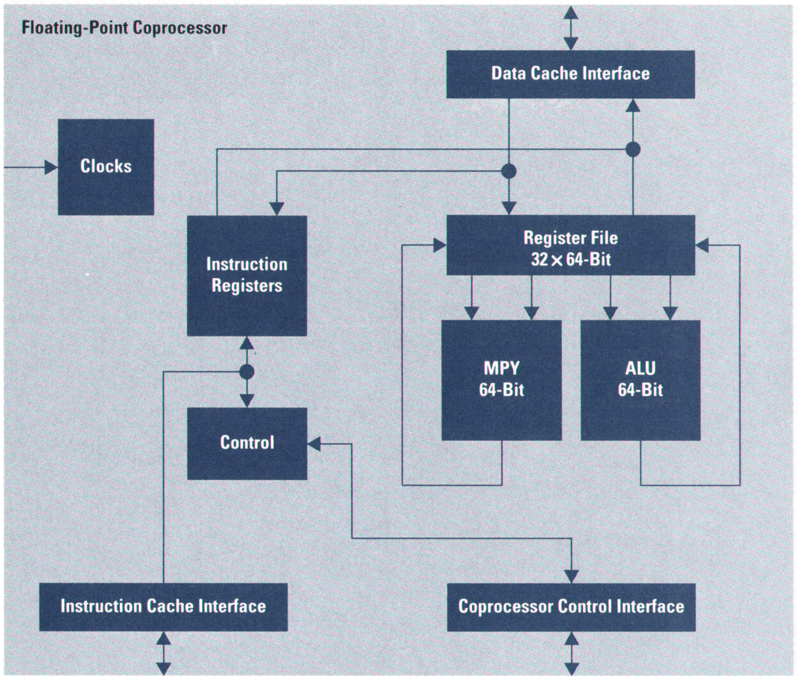

Floating-Point Coprocessor
Data Cache Interface
Clocks
Instruction Registers
Register File 32 × 64-Bit
MPY 64-Bit
ALU 64-Bit
Control
Instruction Cache Interface
Coprocessor Control Interface

Fig. 5. Floating-point coprocessor block diagram showing major blocks and buses.

These features help the FPC achieve a peak throughput of 66 MFLOPs at 66 MHz. This is very important for vectorizable applications, which have a large amount of potential parallelism to exploit. On the other hand, scalar applications have very limited parallelism potential, and for these applications, instruction execution latency is more critical for performance. The FPC has much improved latency compared to previous PA-RISC implementations. Simple add, subtract, and multiply operations execute in only three cycles. Divides require 12 cycles and square roots require 18 cycles.

Architectural Enhancements. A cross-functional team was empowered to identify opportunities to improve system performance through instruction set extensions. This team had representation from the processor, graphics, compilers, performance analysis, and architecture teams. Suggested extensions were validated by analyzing graphics and numerical applications, and hardware and software costs were considered as well.

The enhanced PA-RISC architecture is called PA-RISC 1.1. Enhancements to the core PA-RISC floating-point instruction set include:

- The number of 64-bit floating point registers was increased from 16 to 32. This was done to maximize the effectiveness of compilation techniques such as loop unrolling and software pipelining. These techniques can expose the concurrency present in much floating-point code, but only if the number of registers is sufficient to allow the concurrent computations to be interleaved.
- Both halves of each 64-bit register can be addressed as a 32-bit (single-precision) register. This effectively increases the number of registers for single-precision operations to 64. Moreover, a single 64-bit load or store instruction can be used to load or store pairs of single-precision values, provided they are contiguous and aligned in both memory and the register file. These improvements are particularly important for graphics code, which makes heavy use of single-precision operations.
- It is common for floating-point coprocessor implementations to have at least two independent functional units, one for multiplies and one for adds and subtracts. Two new instructions, FMPYADD and FMPYSUB, were added to execute independent operations in these two functional units in parallel. In other words, a single FMPYADD or FMPYSUB instruction can specify both a multiply and an add or subtract operation. This allows many multiply and add or subtract instructions to be fused into a single instruction.
- An unsigned integer multiply instruction was added which operates on registers in the floating-point register file. Floating-point multipliers can easily perform this operation,

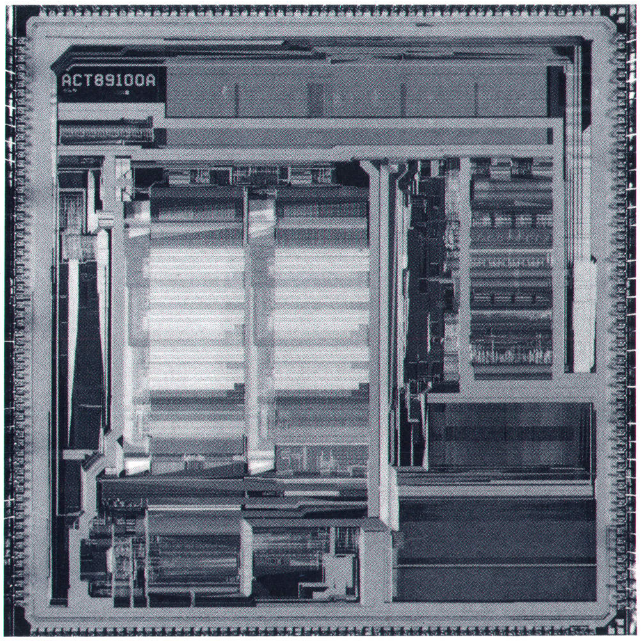


Fig. 6. Floating-point coprocessor chip.

and it is much faster than the formerly used millicode routine for multiplying nonconstant integers.

The remaining enhancements are implementation dependent. These are used only in product dependent code like dynamically linked graphics libraries and the operating system, and are not available directly in application binaries. This ensures that applications will run on future machines that omit or modify these extensions.

- An inverse square root (1/ √x) instruction was added. This is frequently used by graphics code (in coordinate transformations, for example), and it is more than twice as fast as sequential square root and divide operations.
- Graphics clipping operations were accelerated by adding a 12-bit queue of comparison results and a few simple instructions to test conditions within the queue. This doubled the speed of the clipping operation for the most important "trivial accept" and "trivial reject" cases.
- A quad-word (128-bit) store instruction was added. This allows pairs of 64-bit floating-point registers to be stored in five cycles instead of the six cycles it takes using two separate stores. This is important because many codes (like memory-to-memory copy) are limited by load/store bandwidth.

Memory and I/O Subsystem

For typical application workloads, CPU efficiency is of primary importance. Cache-to-memory transfers dominate the system architecture's contribution to overall performance. The memory subsystem is optimized for low CPU-to-memory latency and bandwidth to match CPU requirements.

I/O capabilities must be sufficient to support high-performance networking and disk transfers, and must be flexible and powerful enough to support easy attachment of a wide range of devices. Transfer latencies should be minimized, and moderate-bandwidth DMA should be supported. I/O bus protocols should be simple and use standard signal levels for ease of access.

A high-performance user interface is the hallmark of workstations. High-speed transfers between the CPU or memory and the user interface provide the basis for good windowing graphics performance. Main-memory Z-buffer capability and interpolation support in the memory controller yield high-performance, low-cost 3D graphics.

Basic Architecture. The above requirements led to the block diagram shown in Fig. 1. The CPU is directly connected to the memory and system bus controller by a 32-bit full-frequency bus (PBus). The memory subsystem width is 128 bits plus 16 error detection and correction bits. Standard TTL is used to multiplex this down to 72 bits into the memory and system bus controller. On the other side of the memory and system bus controller is the system bus, a nonmultiplexed, 32-bit-address, 32-bit-data proprietary design.

Three ASICs (application-specific integrated circuits) were designed for the memory and system bus controller subsystem. The main control ASIC, the memory and system bus controller chip (Fig. 7), is housed in a 272-pin PGA. The second ASIC is the system bus interface chip. Two of these 16-bit-wide, custom buffer parts create the system bus. They are packaged in 100-pin plastic QFPs (quad flatpacks). The third ASIC decodes memory addresses and buffers the address and control lines. It is housed in a 68-pin plastic QFP. One buffer part is used for each 18 DRAMs. The number in the system depends on the memory configuration.

Memory and System Bus Controller Functionality. The memory and system bus controller supports standard 80-ns DRAMs of several different densities. Cache miss latency to the first word is 155 ns, and the peak transfer rate is 260 Mbytes/s. Two 72-bit-wide banks are multiplexed and accessed in page mode to provide the necessary bandwidth.

An error detection and correction scheme is employed to control soft errors inherent in DRAM technology. An eight-bit code is calculated from 64 data bits. The code used is capable of correcting any single-bit error, detecting any double-bit error, and detecting any four-bit burst error resulting from a single DRAM failure.

Memory write granularity is 64 bits (72 with error correction code). Therefore, any write of less than a double word must use read-modify-write cycles. Since cache transfers are always aligned on 32-byte blocks, read-modify-write cycles are not necessary for CPU-related transactions.

The protocol of the system bus is fundamentally a single word per transfer. A 4-word-deep buffer is used to gather DMA writes into double-word writes when possible, thus avoiding read-modify-write cycles and achieving good performance.

A single 32-byte buffer is used for temporary storage of cast-out cache lines on their way to memory. This buffer is controlled such that pieces of up to two lines can be present during periods of high flushing activity.

A single 32-byte instruction prefetch buffer is also implemented. Upon an instruction miss, the missing line is fetched, followed by the next line, which is placed in the

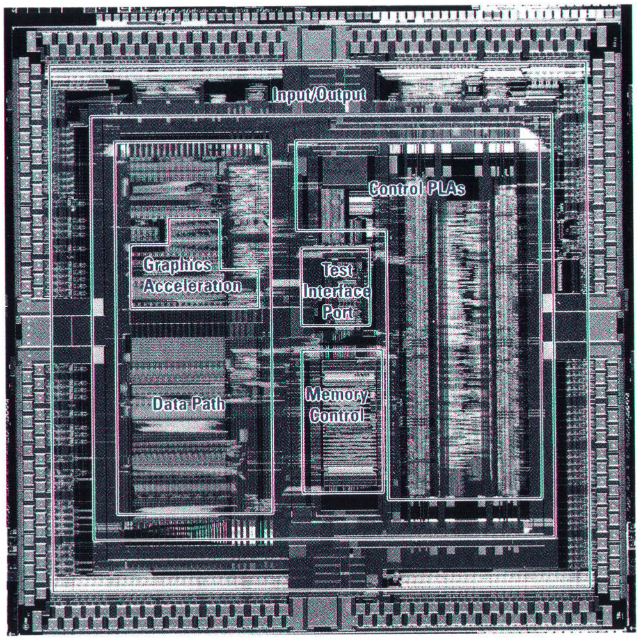


Fig. 7. Memory and system bus controller chip.

buffer. If the next instruction miss hits on the buffer, its contents are passed on to the CPU and the next 32 bytes are fetched from memory.

User Interface Functionality. Several features were added to the memory and system bus controller to improve user interface performance.

User interface adapters connect to the memory and system bus controller via the system bus shown in Fig. 1. This is a nonmultiplexed bus with separate 32-bit address and data paths. The separate address provides sufficient address bandwidth for the nonblock-oriented nature of some user-interface-related transfers.

Windowing and certain graphics functions benefit from high-speed transfers of data from the CPU to the user interface adapter. The CPU and the memory and system bus controller are optimized to perform these writes as quickly as possible.

User interface functions also tend to benefit from high-speed movement of data between the adapter and main memory. Special instructions and hardware were added to the CPU and the memory and system bus controller to move blocks of data to and from the user interface hardware and memory. It was intended that this functionality be available at the user level. As such it is cache coherent and uses virtual addressing.

Graphics functionality is a part of the user interface that is frequently valuable in workstation applications. Hardware to support main-memory Z-buffer, Z, and color interpolation are part of the memory and system bus controller. Good 3D rendering performance can be achieved with low-cost, simple user interface adapters.

Miscellaneous Functionality. EISA is the expansion bus for Series 700 workstations. A special mode is incorporated in the system bus to improve EISA transfer bandwidth.

The memory and system bus controller is the system bus host. In this role it performs arbitration, error management, interrupt management, and a few other housekeeping tasks.

Verification. The memory and system bus controller was the last piece of the system to enter the design phase. Unlike most other pieces of the system, there was little leverage from previous design efforts. Therefore, it was imperative to achieve high-quality silicon on the first release.

A great deal of effort was devoted to verification of the design. A checker program was developed in parallel with the ASIC models. This checker was used for stand-alone chip and system-level simulations in conjunction with an extensive suite of test vectors to verify system behavior.

First silicon was functional, and the operating system booted on the system within 36 hours of receipt of wafers from fabrication. All three memory and system bus interface ASICs were released to production with only minor metal mask revisions.

Technology and Architecture Choices

RISC VLSI design teams are faced with an abundance of choices, ranging from selection of basic core technologies to fine tuning of hundreds of implementation details to give the best performance on the expected customer applications. In making these decisions, designers must consider technology and market trends, core competencies and resources of the design team, customer requirements, schedule and cost goals, and scalability of the design to meet future requirements. Given this complexity, it is not surprising that different design teams come up with radically different solutions to meet the needs of similar products.

Basic design decisions for the PCX-S chip set described in this article were made by carefully evaluating alternatives against system-level cost, performance, and schedule goals. Inputs from manufacturing, technology partners, VLSI and system designers, architects, compiler and operating system designers, performance analysts, and customers were all considered in the initial investigation stage. Some of the more important design decisions and trade-offs made are discussed in the following paragraphs.

PA-RISC Architecture Extensions

A primary goal of the original PA-RISC (version 1.0) architecture definition was to support scalability across all types of computer applications and performance requirements. Before the design of the PCX-S chip set, most PA-RISC implementations had focused on meeting the needs of transaction processing, commercial, and general-purpose multiuser applications. Since PCX-S was to be an implementation also optimized for workstation applications, an extensive analysis of typical customer workstation applications and their associated performance bottlenecks was conducted early in the project (see "PA-RISC Performance Modeling and Simulation," page 21). The results of this study suggested that a small number of critical extensions to the original architecture would provide significant benefits to the customer. The evolved version of the architecture containing these extensions is known as PA-RISC version 1.1 and is fully object-code compatible with applications developed for version 1.0.

The architecture extensions targeted three primary areas of opportunity: improved performance on floating-point-intensive technical applications, improved performance on graphics-intensive and X-windows applications, and improved performance and efficiency of the virtual memory subsystem for lower-cost implementations.

VLSI Technology Selection and Partitioning

The design team quickly narrowed the list of VLSI technologies that could simultaneously meet all of the project goals to just one: high-speed, high-density CMOS technology. While the performance goals could have been easily met with ECL technology, the power, cost, and density penalties incurred would have limited low-cost applications of the chip set and future scalability of the design. Gallium arsenide not only had ECL's drawbacks, but also had not yet been fully proven in a high-volume microprocessor manufacturing environment.

The time-to-market goal also played an influential role in technology selection. Meeting the aggressive schedule requirements for this design required extensive leverage of existing technologies, designer expertise, tools, circuit libraries, and verification suites.

For the CPU chip, the design center selected was a previous PA-RISC processor design,[3] which was first fabricated in early 1989. This design had been optimized for larger commercial and technical multiuser systems, including the HP 9000 Model 870 and HP 3000 Series 980 computers. The processor was fabricated in HP's 1.0-micrometer, three-level-metal, CMOS26 production technology. In addition to the CMOS26 technology, much of the implementation design, circuit libraries, and verification suites, and the multilayer ceramic package design were leveraged to decrease time to market for the new CPU design.

The floating-point coprocessor posed a more difficult challenge for the design team. Since the previous-generation CMOS processor had been designed to interface with higher-cost and higher-power ECL-technology floating-point units, direct leverage of this design was ruled out. It was also determined that the available resources could not support design of both the CPU and the coprocessor on the aggressive target schedule. Fortunately, a semiconductor partner with high-performance CMOS floating-point technology already well along in the design phase was found to codevelop the PCX-S floating-point coprocessor. Time to market was improved by separating the CPU and coprocessor on different chips, while industry-leading performance was achieved by adopting a tightly coupled design approach (floating-point coprocessor interfaced directly to the instruction and data caches) and designing matched, low-skew clock generation circuits on both chips.

Like most workstation designs, the memory and I/O interface functions are implemented on a separate IC. The CPU communicates with this chip over a high-speed synchronous bus, which is functionally and electrically identical to that used in other HP PA-RISC systems. This allowed the processor to be prototyped in existing systems, improving the schedule in the evaluation phase, and made the new design compatible with a variety of other memory systems optimized for different types of applications. Like the CPU and floating-point coprocessor designs, the memory interface IC design used a structured custom approach. Among other benefits, this allowed the inclusion of new features such as embedded graphics acceleration logic.

High-Frequency Design

Although many design techniques were used to improve performance of the processor on customer applications, none was considered as critical as designing the chips for high-clock-frequency operation. Clock speed increases result in across-the-board performance improvements on nearly all compute-bound applications, while architecture and logic features generally benefit a smaller set of specific application types.

There are three basic techniques that are commonly used to increase integrated circuit clock speeds:

- Use a faster VLSI fabrication technology
- Sort or bin parts to select only those at the faster end of the speed distribution
- Improve the efficiency of the design by reducing logic delays, eliminating speed limiting features, and incorporating hand-tuned custom circuit designs and layout in critical timing paths.

The first two techniques were rejected for the CPU chip design because of increased cost, schedule risk, and a more limited growth path. The third method, which emphasizes high-speed design techniques over manufacturing techniques, not only has the advantage of reduced cost over the other two alternatives, but also has an easier growth path to higher frequencies with future technology scaling, since more efficient circuit designs will maintain their speed advantage in faster technologies.

The designers had the advantage of availability of prototypes of a previous design from which the clock circuits, CMOS cell library, and core integer unit blocks were leveraged. Detailed characterization of this device over the specified environmental, operating, and processing conditions pinpointed all of the speed limiting paths and established a baseline operating frequency in excess of 50 MHz. By further tuning of critical paths and detailed worst-case simulations of new features it was determined that the new CPU design could achieve a clock speed greater than 66 MHz for a significant portion of the yield distribution. This speed then became a design goal and all new features considered were first simulated and tuned to determine if they could achieve the same frequency in sufficient quantities to meet anticipated production needs. First silicon prototypes were carefully analyzed and modified to improve all speed limiting paths, with the final result being 100 percent of the the yield distribution operating above the 66-MHz goal. The custom, worst-case design approach used was found to have up to a twofold speed advantage in some cases over more conventional semicustom design methodologies. High design productivity was maintained by resorting to handcrafted custom circuits only in critical timing paths and using other methodologies elsewhere.

Cache Memories

Cache memory organization and speed are also critical to the resulting performance, cost, and schedule of a processor design. In fact, there are many memory-intensive applications in which the performance of the cache memory can play a stronger role in determining application response time than the speed or pipeline efficiency of the core processor. This is becoming even more true as clock speeds rise, instruction pipelines become more efficient, and "typical" customer applications grow in size and memory requirements.

After extensive analysis the design team chose to implement separate instruction and data caches, both of which use a direct-mapped organization and are composed of industry-standard SRAMs external to the CPU and floating-point coprocessor chips. The more conventional approach of implementing the cache memories directly on the CPU chip was considered but rejected for a number of reasons, some of which are outlined below.

Performance. Larger applications generally require access to larger memory working sets, which translates into much higher cache miss ratios for a given cache size. Each cache miss requires a fixed amount of overhead time to service, thereby reducing effective sustained performance. On-chip cache memories, because they are limited to much smaller sizes than can be implemented with external memories, suffer significant performance degradations as the size of applications increases. Simulations showed that a single primary cache of moderate size would outperform a small on-chip cache on real-world applications, even when the latter was backed up by a larger, external, second-level cache.

Frequency and Scalability. An often-quoted reason for the necessity of on-chip cache memories with high-performance RISC CPUs is that clock speed or future scalability potential are limited by using external SRAMs. After reviewing SRAM technology trends and simulating new cache timing designs it was found that, in fact, the data indicated a somewhat different conclusion. First, the speed of available SRAM technology at a given point in time has historically exceeded (and will probably continue to do so) the requirements of the fastest processors built in a similar generation of fabrication technology. The rate of improvement of these devices is actually increasing because of increased competition and expanding markets. SRAMs are increasingly using VLSI technologies optimized for high-speed, high-density memory circuits to their advantage. Prices of competitive devices are decreasing exponentially (at rates of up to 30% every 6 months). Second, generally speaking, most previous external cache memory designs were limited by timing overhead because of nonoptimal clocking and package designs, not by the speed of available SRAM technology. Overhead times (the portion of the cache cycle time over and above the access time of the memory) of 10 to 15 ns or more were found to be typical on competing designs evaluated by the team, effectively cutting in half the potential clock frequency possible with the available SRAM technology. By focusing on improving these bottlenecks instead of adopting a radically different cache architecture the PCX-S design achieved both an industry-leading clock frequency and reduced cost of external SRAM components because of less aggressive access time requirements. The resulting design achieves a processor cycle time just 3 ns greater than the access time of the SRAMs employed.

Manufacturability. Time to market and manufacturability are also enhanced with an external primary cache design. The VLSI CPU chip is generally the most complex and expensive IC component in a system. Addition of large numbers of transistors to implement caches on this die can drive up costs and decrease manufacturing yields significantly. Also, fabrication technologies that are optimized for high-speed logic circuits are often less than optimal for implementing high-density memory circuits and vice versa.

Instruction Execution Pipeline

As a rule, competitive microprocessor designs use pipelined execution of instructions. The details (how each instruction is split between functions, number of stages, etc.) vary considerably between designs, but all have the same basic goal of exploiting parallelism to allow instructions to be issued and completed at a rate faster than they could be if unpipelined. The performance of different pipelines varies widely as a function of the efficiency of the implementation. The peak performance of a pipeline measures only the maximum rate at which instructions can be dispatched under ideal conditions, while the average or sustained performance includes nonideal events such as pipeline stalls because of code branches or hardware resource contention, measured while running real applications. Optimizing the latter was the goal for the PCX-S processor design.

Both a conventional RISC pipeline and a superscalar implementation were considered as viable options in the design. Superpipelining was another technique included in the study for comparison purposes, although it can be maintained that this approach is just an extension of conventional pipelining rather than a new pipeline implementation technique. Techniques such as branch prediction, data bypassing, cache streaming, and prefetching were first applied aggressively to the design to increase pipeline efficiency. Also considered in the design decision were the impacts of choices on clock speed, cost, and time to market. The design team chose a conventional pipeline implementation that includes a number of features to improve overall efficiency as the best suited for the performance, price, and schedule goals. The techniques used to increase the speed and efficiency of this pipeline are also designed to be fully applicable to future superscalar implementations.

Recent attempts to classify RISC processor designs based only on such features as their pipeline implementation miss many of the critical elements that determine the actual competitiveness of the designs. In reality, much more basic elements such as circuit design techniques, clock speeds, cache memories, main memory latency, and so on are far more critical in determining basic competitive attributes (cost, performance, time to market).

Future Growth Path

To maintain competitiveness in the high-performance arena, RISC designs must offer continual performance and cost improvements as the basic enabling technologies evolve. The industry trend for performance growth has historically been a doubling of high-end performance levels every 12 to 18 months. To maintain this level of improvement on a regular basis with a realistic level of investment in design resources and new technologies requires a significant amount of leverage between successive generations of designs.

In making the fundamental design trade-offs for the PCX-S processor, the impact of each decision on future scalability was an important criterion. Some of the more important aspects of the design that support scalability are:

- High-Frequency Circuit Designs. By using efficient circuit designs to achieve high speeds rather than pushing technology requirements to the limit, room is left to allow quick speed increases through technology scaling. The CPU design uses a technology featuring 1.0-μm drawn transistor channel lengths to achieve speeds in excess of 66 MHz. At this time, technologies using 0.8-μm geometries are widely available and 0.5-μm technologies are on the horizon. Algorithmic scaling of designs into faster fabrication technologies can allow significant speed improvements without requiring extensive redesign.

PA-RISC Performance Modeling and Simulation

The increasingly competitive computer market requires that we continue to provide the highest level of performance while offering competitive price/performance. Performance is traditionally measured with industry-standard benchmarks such as Linpack, the SPEC suites, and TPC-A. As important as benchmarks are, they are often too simplistic to reflect the customer's actual workload. To deliver superior performance to the customer it is no longer sufficient to base architectural and processor implementation decisions on industry-standard benchmarks alone.

Hewlett Packard has invested in the development of specialized hardware and software instrumentation to aid in the analysis of computer system behavior under actual customer workloads. With product offerings in both instruments and computers, HP is highly qualified in computer systems instrumentation. Today we have the capability to characterize workloads fully to any level of detail, from operating system call graphs to pipeline interlocks. Hardware instrumentation can analyze CPU, cache, memory, and I/O behavior nonintrusively. The resulting data from actual customer workloads is used to drive detailed CPU models. The models have proved extremely valuable in quantifying the potential performance gains from instruction level parallelism, specific processor implementation decisions, compiler scheduling algorithms, and operating system policy changes.

The design of the PA-RISC processor used in the HP 9000 Series 700 workstations was heavily influenced by information collected from customer workloads. The cache-tag emulator, which consists of specialized cache behavior monitoring hardware, was used to characterize the performance behavior of several alternative cache designs. Information obtained from instruction traces collected from customer workloads was used to guide numerous pipeline trade-offs. This information was used in analyzing branch prediction policies, instruction streaming during cache fill, stall on use, and instruction cache line prefetching. Instruction traces were also used to analyze several alternative TLB designs. Organization, replacement policy, page size, and software management policies were thoroughly analyzed. The final design incorporates a fully associative organization based on replacement policy that is an approximation of LRU (least recently used). Page size was increased to 4K bytes, improvements were made to reduce the number of cycles needed to update the TLB, and new software management policies were developed to improve overall performance. Use of actual workload data enabled the processor design team to improve performance dramatically while substantially reducing complexity.

This approach has helped to identify the set of design trade-offs that provide the best value for the customer. It has helped PA-RISC implementations steer clear of several pitfalls plaguing some processor designs. An example of the influence of instrumentation on design decisions is in the area of high-degree superscalar machines. While in theory these machines can provide substantive performance improvement, in practice this is often not achieved. Failure to consider actual customer workload behavior when considering superscalar designs can lead to exceptionally complex designs that yield little benefit to the customer. Worse still, the complexity of these designs often limits the clock rate and future performance scaling. The recently announced superscalar implementation of the PA-RISC architecture[1] was created only after careful analysis of actual customer workloads. Based on this analysis a minimal set of superscalar features was identified that provided the maximum benefit to the customer. Limiting superscalar capability to this minimal set also minimized complexity and development time. An additional benefit is that it allows a high clock rate, which is crucial to performance.

It is critically important that the performance analysis used in design decisions be accurate. This is guaranteed through continuous verification of performance projections. The projections are compared with actual measurement results. Any discrepancies cause careful review of the models to identify the source of the error and make the necessary corrections. Use of this process over the past five years has enabled us to predict the performance of multiuser systems consistently within 5%. For computational systems the accuracy has been somewhat better, generally falling in the 1%-to-2% range.

In the future, computing environments will become more complex. The adoption of client/server configurations and the widespread use of multiprocessors are challenges to instruments and models. To address these new environments we have been identifying customer workloads to instrument and are developing instrumentation to handle multiple systems. This will guarantee continued selection of design trade-offs that give the best value to the customer.

Reference

1. E. Delano, W. Walker, J. Yetter, and M. Forsyth, "A High-Speed Superscalar PA-RISC Processor," *COMPCON Spring '92 Digest of Technical Papers,* February 1992.

Richard G. Fowles
Project Manager
Systems Performance Laboratory

- Cache Memory Timing Techniques. Emphasis on improving the performance of packaging and off-chip cache timing paths also promotes future scalability. The 66-MHz version of the chip requires external SRAM cycle times of only 12 ns to operate over the specified environmental and operating conditions. Cycle times of 8 ns are available in off-the-shelf components today; these would allow much higher processor clock speeds. Furthermore, advanced multichip module packaging technology, which is not used in the present design, offers further opportunity to increase the speed of off-chip paths by reducing chip crossing delays and clock skews.
- Integration Levels. Adopting a multichip design that did not push the limits of chip density or die sizes enhanced both manufacturability and time to market. Future designs still have significant room for integration of additional features both to reduce cost and to increase performance. Because of the organization and partitioning of the CPU and floating-point coprocessor chips, it is a relatively straightforward process to integrate the two functions onto a single die, for instance.
- SRAM Technology Growth Path. The chip set was designed to support external cache memory sizes 4 to 8 times larger than those used in the present Series 700 systems without requiring redesign of any of the VLSI components. This enables future system designers to take advantage of the industry SRAM learning curve, which has resulted in exponential price reductions and a quadrupling of component densities every few years, in addition to regular speed improvements.

In addition, rapid advances in compiler technology, graphics hardware and algorithms, architectural features, and DRAM technology will also contribute to maintaining exponential system performance increases in the years to come. Many of these performance growth concepts have, in fact, been incorporated into an evolved version of the PCX-S CPU called the PA 7100.[4] This chip was first sampled less than one year after introduction of the first PCX-S-based systems.

Acknowledgments

The VLSI described in this article was designed and verified by a dedicated team of individuals in HP's Engineering Systems Lab. They are the true source of most of the contributions we have discussed. We are grateful for the opportunity to summarize their many contributions. The CPU design and verification team consisted of Paul Bodenstab, Duncan Weir, Tom Asprey, Mike Buckley, Eric DeLano, Mark Ludwig, Jeff Trull, Barry Arnold, Dan Halperin, John Wanek, Terry Blanchard, Harlan Hill, Bruce Long, Rose Melville, Joe Butler, Grace Chuang, Doug Josephson, Ed Ogle, and Steve Undy. The memory and system bus controller design and verification team consisted of Russ Brockmann, Paul Bodenstab, Duncan Weir, Grace Chuang, Jon Lotz, Bob Naas, Leon Sigal, Mike Buckley, Tony Lelm, Mark Ludwig, Jeff Trull, Bill Weiner, Mick Bass, Dan Halperin, Mike Jassowski, John Spencer, Dave Hollenbeck, Harlan Hill, Bill Hudson, Bruce Long, Steve Undy, Syrus Ziai, Tony Martinkus, and Richard Alley. The floating-point coprocessor design and verification team consisted of Rick Leubs, Anne Reynolds, Ray Gratias, Dan Halperin, Mike Jassowski, and Will Walker. We would also like to acknowledge Jeff Yetter and Craig Simpson for their management contributions to the PCX-S chipset. Finally, we wish to thank Charlie Kohlhardt, John Wheeler, Denny Georg, and Carl Snyder for their leadership and support during chip development.

References

1. D. Tanksalvala, et al, "A 90-MHz CMOS RISC CPU Designed for Sustained Performance," *ISSCC Digest of Technical Papers*, February 1990, pp. 52-53.
2. M. Forsyth, S. Mangelsdorf, E. Delano, C. Gleason, and D. Steiss, "A CMOS PA-RISC Processor for a New Family of Workstations," *IEEE COMPCON Spring '91 Digest of Technical Papers*, February 1991.
3. J. Lotz, B. Miller, J. Lamb, M. Forsyth, and T. Hotchkiss, "A CMOS RISC CPU Designed for Sustained High Performance on Large Applications," *IEEE Journal of Solid-State Circuits*, October 1990, pp. 1190-1198.
4. E. Delano, W. Walker, J. Yetter, and M. Forsyth, "A High-Speed Superscalar PA-RISC Processor," *COMPCON Spring '92 Digest of Technical Papers*, February 1992.

ECL Clocks for High-Performance RISC Workstations

In the HP 9000 Series 700 workstations, clock signals are distributed using differential ECL circuits, and the VLSI chips have CMOS inputs operating at ECL levels. Critical clock delay signals are routed on 50-ohm striplines on printed circuit board inner layers.

by Frank J. Lettang

In synchronous computers the clock system generates the rhythm that makes everything work, so clock performance plays an important role in determining the maximum operating frequency of a high-performance workstation. Clock design and distribution are important system design issues that separate semifast computers from extremely fast computers.

To make an effective clock system it is important to minimize both clock jitter and skew since both effects reduce the time available for the computer to do useful work between each clock tick. Component variations, electrical noise, temperature, and supply voltage changes all combine to increase the amount of jitter and skew found in a clock system. The clock system design must minimize these problems, not only in the clock circuits themselves but also in the components that use the clocks.

In HP 9000 Series 700 workstations, high-speed clock signals are distributed using ECL logic because of its superior timing characteristics and faster rise and fall times. To improve the performance, nearly all the ECL signals are routed differentially from the outputs of low-skew clock distribution ICs. Using both the signal and its complement to drive the differential inputs of an ECL device greatly reduces the effects of differences between the rise and fall times. CMOS differential inputs operating at ECL thresholds are used on all the high-speed VLSI chips. This minimizes on-chip delay and skew by making the conversion to internal CMOS levels more efficient. While all of this may seem a bit expensive,

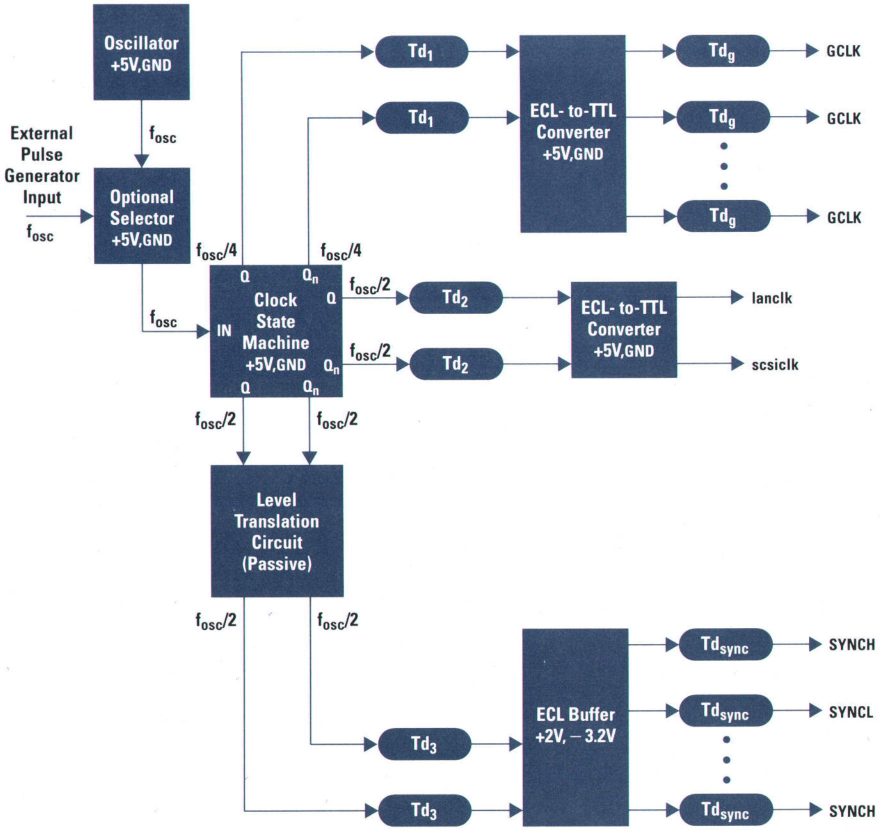


Fig. 1. HP 9000 Series 700 workstation clock system block diagram. All active clock components are on the CPU board except the ECL-to-TTL converter that drives lanclk and scsiclk.

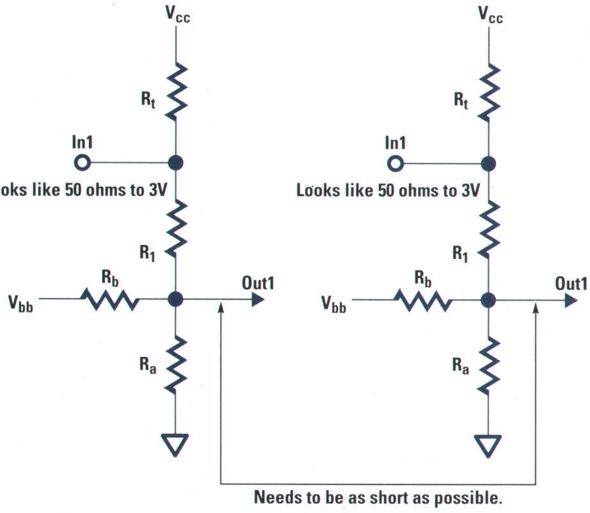


Fig. 2. Passive differential level translation circuit.

the performance of the clock system affects the performance of all of the devices that connect to it, and spending a little more money on the clock system saves a lot of money in other areas.

Fig. 1 is a block diagram of the Series 700 clock system. The system starts with an ECL oscillator running at twice the processor frequency. This oscillator drives the clock input of an ECL finite state machine that generates differential ECL clock signals at the processor frequency and half the processor frequency. These signals need a specific phase relationship. The state machine always reverts to the proper transition pattern without the need for a reset signal. From the state machine, differential ECL signals drive clock distribution chips and ECL-to-TTL converters.

The state machine is implemented using a number of ECL high-speed edge-triggered flip-flops. Since all the flip-flops are on the same die, skew and jitter are minimized. The oscillator has a frequency stability of 100 ppm, which contributes only about 2 ps to the clock system jitter. Oscillators with good frequency stability are essential to designing high-performance clock systems. As clock speeds increase beyond that of the current Series 700 workstations, providing good frequency references will become an even more important challenge. For the current clock system the maximum measured combined jitter and skew is about 75 ps and is largely the result of noise and component behavior.

A differential ECL level translation circuit is needed between the clock state machine, which uses ECL referenced to 3 volts, and the clock distribution chip used by the VLSI chips, which is referenced to 0 volts. This eight-resistor network (Fig. 2) properly terminates the complementary ECL signals at the characteristic impedance of the striplines that carry them (50 ohms) while generating the input signal needed by the ECL clock distribution IC. Since the output impedance of this network is relatively high compared to what can be achieved on the printed circuit board, the passive network is located physically as close as possible to the clock distribution chip.

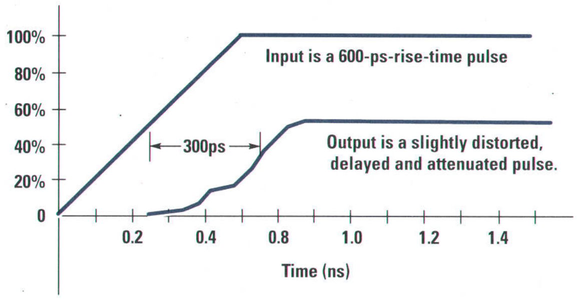


Fig. 3. Time-domain response of the passive differential level translation circuit.

Using an HP 54121T time-domain network analyzer, the response of this circuit to input pulses is easily verified. The response is shown in Fig. 3. The printed circuit board implementation generates an edge delay of about 298 ps and degrades the 700-ps ECL rise time slightly. This is largely the result of the parasitic capacitance and inductance present in any real implementation of a level translation network. It is always useful to verify designs of this type with actual measurements and check for unexpected parasitic effects.

The high-speed system bus uses TTL clocks running at half the processor frequency. These clocks are generated by an ECL-to-TTL converter that doubles as a TTL clock distribution chip. A differential ECL delay line, implemented as a pair of striplines, is placed between the state machine and this converter to optimize setup and hold times on the system bus. ECL signals are ideal for realizing highly accurate signal delays because of their crisp edges and high-quality terminations. No level translation is required on this path since the converter expects 3-volt-referenced ECL and that is what the clock state machine generates.

Clock signal quality and noise immunity are maximized by routing critical clock delay lines on what are nominally 50-ohm stripline geometries implemented on printed circuit board inner layers. The quality of these delay lines is easily verified using the HP 54121T as a time-domain reflectometer. Fig. 4 shows a typical measurement.

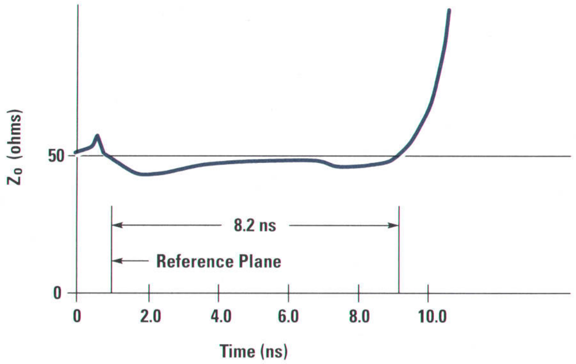


Fig. 4. TDR measurement of an unterminated 4.1-ns clock system delay line.

Conclusion

By carefully engineering the Series 700 clock system to minimize undesirable jitter and skew, valuable cycle time is used as efficiently as possible. Additionally, the clock system implements controlled delays designed to optimize performance of the system bus. Clock design directly translates into product performance, so spending extra design effort in this area is extremely cost-effective.

Acknowledgments

Many talented individuals contributed to the HP 9000 Series 700 workstations. I would like to thank the following people for their key roles in the development of the Model 720/730/750 clock systems: Rob Horning for his technical expertise and advice and Sharon Ebner, Ed Kean, and Robin Stevens for their help and support.

HP 9000 Series 700 Input/Output Subsystem

Integrated on a single 8.5-by-11-inch I/O board is hardware support for the SCSI, the Centronics parallel printer interface, two RS-232 ports, the IEEE 802.3 LAN, the HP-HIL, four audio tone generators, and a real-time clock. An application-specific IC serves as I/O subsystem controller.

by Daniel Li and Audrey B. Gore

In today's environment of ever-increasing CPU performance, it is critical that I/O subsystem performance keep up with CPU performance. If I/O subsystem performance cannot keep up with CPU performance, the system will become I/O bound and will not benefit from increased CPU performance. The goal in designing the I/O system for the HP 9000 Series 700 workstations was to design a balanced high-performance system with many built-in features and yet still keep the system cost low.

To increase performance, the core I/O subsystem is attached directly to the high-bandwidth, pipelined system bus. The high-speed I/O devices, such as the SCSI (Small Computer System Interface) and the Ethernet local area network (LAN), perform DMA (direct memory access) data transfers to and from the system memory with very low latency. This not only greatly reduces the chance for LAN and SCSI controllers to overrun their internal buffers, but also minimizes the use of available system bus bandwidth and frees bandwidth for graphic devices and I/O expansion slots.

A low-cost CMOS ASIC (application-specific integrated circuit) chip called the I/O controller implements the logic that controls the interface between the I/O subsystem and the system bus, thereby minimizing the need for interface logic.

I/O System Features

In the HP 9000 Series 700 workstations, a set of I/O functionality is integrated on an 8.5-by-11-inch system I/O board. The following is a list of the built-in I/O system features:

- SCSI with DMA scatter/gather capability
- Parallel interface with DMA capability (bidirectional with HP ScanJet support)
- Two high-performance, asynchronous RS-232 ports
- Ethernet LAN with DMA capability
- HP Human Interface Loop (HP-HIL)
- Four audio tone generators (internal and external capabilities)
- Two 128K × 8-bit ROMs containing self-test, boot, console handler, CPU, and I/O firmware code
- Real-time clock with lithium battery backup
- 8K × 8-bit EEPROM nonvolatile memory.

I/O Subsystem Overview

Fig. 1 is a block diagram of the I/O subsystem. Inside the I/O subsystem there are two buses: a local data bus and an address bus. All functional blocks are located between these buses and use some portion of the local data bus. Depending on its specific requirements, a given block may or may not use some portion of the address bus.

The core I/O subsystem is attached to the system bus. Data communication passes through a 32-bit bidirectional tristate register. I/O addresses pass through a 30-bit bidirectional tristate register, making the I/O subsystem capable of performing master DMA operations.

If necessary, byte addressing during DMA operations is done to get into word alignment or to finish a transfer that does not end on a word boundary. The I/O controller chip automatically handles word alignment.

A set of bidirectional tristate buffers (74245s) attached to the local data bus and to the system bus interface data registers makes word assembling and data byte disassembling possible.

Bus Controller Interface

Except for the interrupt request signal, the signals needed by the core I/O subsystem to interface to the memory and system bus controller chip are all defined in the system bus specification. The interrupt is asserted by the I/O controller chip on behalf of the devices inside the core I/O subsystem. The interrupt is deasserted after a CPU read to the I/O controller chip's interrupt register.

Direct Memory Access (DMA)

There are three bus masters that use DMA on the I/O subsystem board: the LAN, the parallel interface, and the SCSI. Inside the NCR53C700 SCSI controller, there is a bus master DMA device which is capable of moving data between disk and system memory at the rate of 27.7 Mbytes/s. This assumes the NCR chip is running at 33 MHz, has a burst size of 24 bytes, and has an arbitration overhead of 13 system bus clock cycles. The Intel 82596 LAN controller also has a built-in high-performance DMA controller. Inside the I/O controller chip, a 32-byte FIFO register and a DMA channel support bidirectional parallel printer interface applications, such as the HP ScanJet.

The I/O controller chip uses the system bus request and bus grant signals on the system bus to gain access to the bus for each device. If multiple bus requests occur simultaneously,

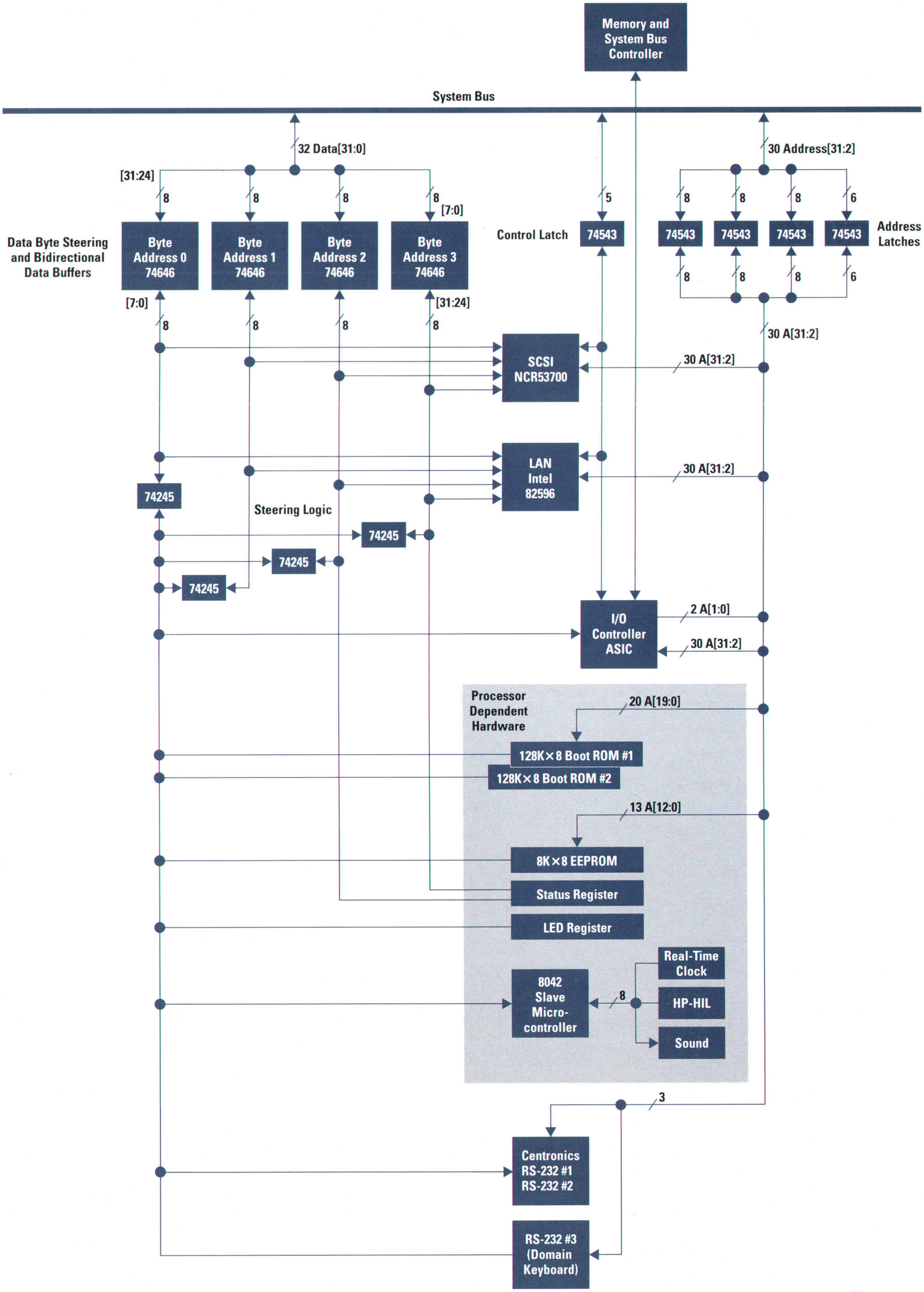


Fig. 1. HP 9000 Series 700 I/O subsystem block diagram.

the I/O controller chip will arbitrate access for one bus cycle, allowing each requesting device to master the bus according to its priority.

SCSI

The SCSI (Small Computer System Interface) is a system-level interface bus used to connect disk drives, tape drives, and other I/O devices to a computer system. Numerous workstations today support this bus standard, and it is becoming the de facto disk interface standard.

The Series 700 I/O subsystem will support the SCSI II specification. Currently, it supports the 8-bit data bus, running single-ended at 5 Mbytes/s.

The NCR53C700 intelligent SCSI controller chip is used in the core I/O subsystem. On the host bus side, it has an on-chip 32-bit DMA engine and a "script processor," which fetches its own commands and performs SCSI transactions with minimal host processor intervention. A small portion of the I/O controller chip is used to implement the logic that controls the interface between the NCR53C700 and the system bus.

A concern about the SCSI compared to device-level interfaces is the amount of latency the SCSI control logic adds to the total subsystem overhead. The Series 700 SCSI subsystem is designed to keep this overhead to a minimum. The script processor inside the NCR53C700 minimizes SCSI I/O start latency; it takes only 500 ns to begin, compared to 2 to 8 ms for a traditional SCSI controller. The NCR53C700 can make decisions based on phase changes on the SCSI bus and compare specific data values. This minimizes the number of interrupts to the processor, which may take more than several hundred microseconds to execute and can be a large source of performance loss.

In the HP-UX* operating system, a disk data transfer may be broken up and data buffers may be scattered throughout system memory. The latency to reinstruct the DMA operation can result in a missed disk revolution. The performance degradation resulting from this data scattering is minimized by the scatter/gather feature of the NCR53C700 chip.

The combination of the fast system bus transfer rate, the fast SCSI bus transfer rate, and the efficient architecture of the NCR53C700 chip enables us to achieve a high disk I/O transfer rate without the need to place a large private buffer between the SCSI controller chip and the system bus interface. This not only lowers the system cost but also avoids the complexity and latency of managing the buffer, thus maximizing the disk I/O throughput.

Local Area Network (LAN)

The Series 700 workstations implement a built-in LAN that conforms to the IEEE 802.3/Ethernet standard. The LAN circuitry consists of the Intel 82596DX-82C501AD chip set, plus a transceiver chip and associated circuitry. The Intel 82596DX is an intelligent, high-performance 32-bit LAN controller. The 82C501AD device provides the electrical interface to the transceiver cable (AUI or built-in Cheapernet MAU), generates a 10-MHz transmit clock for the LAN controller, and performs Manchester encoding and decoding of the transmitted and received frames.

The 82596DX has large on-chip FIFO buffers, 128 bytes for receive and 64 for transmit. It also provides a four-channel DMA controller to communicate directly with the system memory via the high-performance system bus interface. The low memory access latency and the large on-chip FIFO practically eliminate overrun and underrun without using an external FIFO or dedicated packet buffer memory.

The 82596DX bus interface is optimized for the Intel386 microprocessor bus. The similarity between the system bus and the Intel386 bus made the control circuitry to interface the two extremely simple; for the LAN-specific portion of the I/O controller chip, the total gate count is less than 100. Unlike some older-generation controllers, the 82596DX system clock rate is asynchronous to the 10-MHz Ethernet clock so that the controller, the I/O controller chip, and the system bus all run at the same frequency. For DMA operations, the I/O controller chip arbitrates system bus access on the 82596DX's behalf, manages the address valid/ready handshake, and controls the address and data buffers. The I/O controller chip also controls access to the 82596DX's CPU port.

The four-channel DMA controller manages memory structures automatically using command chaining and bidirectional data chaining. This allows autonomous block data transfers and greatly reduces the CPU overhead. The four channels are: CU (transmit header), TXD (transmit data), RU (receive header), and RXD (receive data).

A permanent copy of the LAN station nodal address is kept in location 0 of the nonvolatile RAM (EEPROM). The on-board processor dependent hardware status register has three bits for LAN connector status: one for MAU power/fuse and two for ThinLan/ThickLan selection; it also has three bits for the SPU ID.

The hardware design is IEEE 802.3 compliant and Ethernet revision 2 compatible. A removable jumper selects either the AUI or BNC connectors. The LAN heartbeat is indicated by a front-panel LED to facilitate diagnostics.

Bidirectional Printer Interface

A majority of computers communicate with output devices such as printers and plotters via the Centronics parallel interface. Centronics is an 8-bit parallel, synchronous interface with additional control signals from the host computer and status signals from the peripheral.

The Centronics interface was defined many years ago. With the passage of time, the demand for higher transfer rates motivated some companies to modify the Centronics standard so that it transfers data faster in a stream mode without the overhead of a shared handshake for each data byte transferred. Some products' Centronics interfaces do not have overlapping strobe and busy handshake signals. Yet another variation of the Centronics interface is the bidirectional HP ScanJet interface. In the output mode, this interface is compatible with the Centronics standard. However, in the input mode, the peripheral becomes the master and the busy and strobe signals change direction.

The Series 700 parallel interface is designed to support all these variations of the Centronics interface. Additional functionality is implemented by special logic built into the I/O controller chip, and the Western Digital WD16C552 chip is used at the parallel port as the device driver/receiver interface.

The Series 700 I/O subsystem supports automatic hardware handshaking at the parallel interface, which removes the burden from software and improves performance. However, the software is capable of controlling all handshakes directly.

The Series 700 hardware also adds DMA capability to the parallel interface to increase the performance. Inside the I/O controller chip, a parallel-port DMA controller and a FIFO buffer are implemented. The FIFO buffer supports 32-byte inbound and 32-byte outbound data transfers. The hardware is capable of a 380-kbyte/s data transfer rate.

Serial Channel Communications

There are two RS-232-C serial ports in the I/O subsystem. They are fully compatible with the National NS16550A chip. Host communication to the serial ports and the status and control registers is done through the I/O controller chip. Each serial port uses a 16-byte inbound and a 16-byte outbound FIFO data buffer to increase efficiency.

The interface supports a 19.2-kbaud inbound data transfer rate with no data loss, using only software XON/XOFF flow control. Up to 230.4 kbaud inbound is supported with no data loss using hardware flow control. Hardware flow control is implemented by additional logic and controls the RTS line to the peripheral, thus preventing data overrun errors in the input FIFO buffer or input holding register. This feature is intended for use with high-speed serial devices that are capable of quickly suspending serial data transfers to the host when the RTS line is dropped by the host interface controller hardware.

Processor Dependent Hardware

The processor dependent hardware includes the boot ROM, EEPROM nonvolatile memory, status switches, and status LEDs, as well as the 8042 slave subsystem, which consists of the real-time clock, audio generator, and HP-HIL.

Two sockets are available for boot ROM. Initially, each socket can hold a 128K × 8-bit ROM, but the sockets are wired so that they are also capable of holding 512K-byte parts if necessary. Thus, there are 256K bytes of boot ROM available, expandable to a maximum of 1M bytes.

The I/O subsystem has an 8K × 8-bit EEPROM which may be used for storing system configuration status and any other alterable, nonvolatile information. The manufacturer of the EEPROM guarantees reliability of at least 10 years with a maximum total number of write cycles of 10,000 for any given byte.

The 8042 subsystem is taken from the HP Apollo Series 400 workstations and is composed of several devices: battery-backed real-time clock, system timers, user timers, audio generator, and HP Human Interface Loop (HP-HIL). These devices are controlled via an Intel 8042 slave microprocessor which acts a server for these devices. Access to the devices is through the 8042 command/data protocol under control of the I/O controller chip. The HP-HIL supports eight input devices (keyboard, mouse, tablet, etc.) with a single system connector.

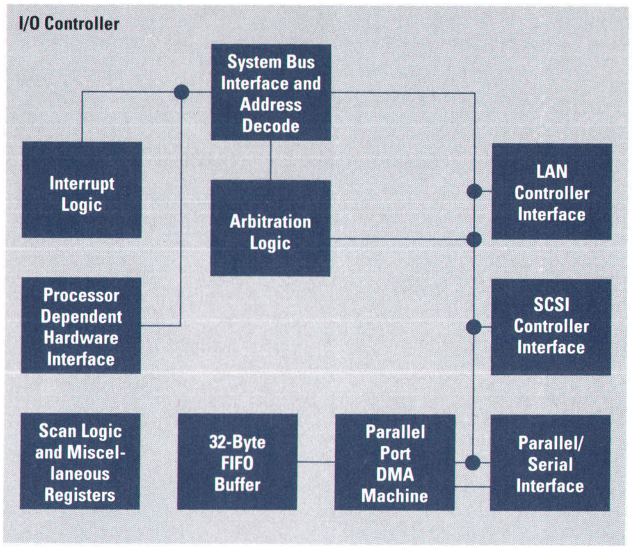

I/O Controller
System Bus Interface and Address Decode
Interrupt Logic
Arbitration Logic
LAN Controller Interface
Processor Dependent Hardware Interface
SCSI Controller Interface
Scan Logic and Miscellaneous Registers
32-Byte FIFO Buffer
Parallel Port DMA Machine
Parallel/ Serial Interface

Fig. 2. I/O controller ASIC chip block diagram.

Porch Board

Since there are so many I/O connectors, a double-height bulkhead is needed to hold them all. The porch board is the means by which the signals get to the top level of the bulkhead. An 80-pin connector carries signals from the system board up to the porch board. The I/O connectors on the system I/O board are SCSI II, LAN AUI, and LAN BNC. The I/O connectors on the porch board are HP-HIL, audio, Centronics, and the two RS-232 ports. Other parts on the porch board are for ESD protection and EMI prevention.

I/O Controller Chip

The I/O controller chip is a 15,000-gate ASIC controller for the I/O subsystem. It interfaces with the memory and system bus controller chip and the system bus. Besides control functions, it contains registers for subsystem interrupts, a DMA channel for the parallel printer interface scan path logic, some parallel printer interface related logic including a 32-byte bidirectional FIFO buffer, system bus address decoders, and other miscellaneous registers. The I/O controller chip is housed in a 160-pin PQFP (plastic quad flat package) with a large number of power and ground pins to minimize simultaneous switching noise. Two pins on the I/O controller chip are designated for scan-chain testing. The I/O controller chip also provides a tristate test mode, which facilitates board production testing. In the tristate test mode, all of the outputs and bidirectional pins of the I/O controller chip are tristated (in a high-impedance state, instead of being driven high or low). This test mode helps isolate the I/O controller chip while testing other parts on the board.

Fig. 2 shows the I/O controller chip block diagram. The I/O controller chip is the heart of the I/O subsystem. Its main functions are I/O subsystem address decoding, bus arbitration, interrupt control, peripheral device handshaking, and data flow control. The I/O controller chip interfaces with the memory and system bus controller chip, the system I/O controller, and the I/O subsystem devices. The I/O controller chip is designed to work at frequencies from 25 to 33 MHz with no modifications. It provides the following resources and functions:

- System bus interface and arbitration
- Address and data bus transceiver control
- Western Digital serial/parallel chip control
- DMA channel for parallel printer interface
- Intel 82596DX LAN chip control
- NCR53C700 SCSI chip control
- Intel 8042 control (HP-HIL, audio, real-time clock)
- Processor dependent hardware control (ROM, EEPROM, status registers, LEDs)
- Interrupt registers.

Arbitration. The need for arbitration arises because the address and data buses for SCSI, LAN, and the I/O controller chip are multiplexed together on the I/O system board. The address buses of these devices are connected together directly. For the data buses, the I/O controller chip's 8-bit data bus first goes through the steering logic (74245s) and then connects with the SCSI and LAN data buses. This 8-bit data bus is also multiplexed with the Western Digital chip (WD16C552) and other I/O devices.

System Bus Interface. The system bus interface block of the I/O controller chip combines the graphics interface and byte packing functions. For master (outgoing) transactions, the system bus interface is responsible for enabling and clocking the external address and data buffers and the I/O controller chip address out drivers, and for managing the system bus address valid/ready protocol. It also handles error conditions.

For slave (incoming) transactions, the system bus interface block detects the start of transactions to the core I/O system and generates the external signal for latching addresses in the external buffers. This is one of the most critical timing paths in the I/O controller chip and the core I/O system. The system bus protocol and tight timing make detecting and latching new valid core I/O addresses difficult. Furthermore, the address must be held steady for the length of the transaction and released in time to look at new addresses. The system bus interface decodes the upper eight bits of the incoming address to determine if the current transaction is to the core I/O system. The system bus interface also controls byte packing in the core-I/O-to-system-bus direction and byte unpacking in the system-bus-to-core-I/O direction for the 8-bit devices in the core I/O system.

Address Decoder. The address decoder serves two main functions: selecting slave units by decoding incoming system-bus addresses, and synthesizing address bits 1 and 0 for devices that need them.

In general, only as much decoding as necessary is done to determine which unit is being addressed. Consequently, there is much aliasing. The upper eight address bits do not come into the address decoder. These are used instead by the system bus state machine, which generates a master select enable selecten. To prevent select signals from being generated from spurious or invalid addresses, all the decodes are ANDed with the selecten signal. Since the address is latched externally, it is guaranteed valid and stable for the entire time selecten is asserted.

Interrupt. Inside the I/O controller chip there are an interrupt request register (IRR) and an interrupt mask register (IMR) similar to the interrupt structure of the PA-RISC architecture. An interrupt pending register (IPR) is also provided. All of these registers appear to be 32 bits wide and are accessed as such. However, only the 15 least-significant bits are implemented for each register. The remaining bits are not affected by writes and are always read as zeros.

The possible sources of the defined interrupts are:

- NMI from EISA
- 8042 general interrupts
- 8042 high-priority interrupts
- Reserved
- Reserved
- WD16C552 SIO 1
- WD16C552 SIO 2
- WD16C552 parallel printer interface
- LAN
- SCSI
- EISA
- Graphics1
- Graphics2
- SIO
- Domain keyboard.

The interrupt pending register (IPR) is used to latch incoming interrupts and indicate that they are pending. The external interrupts are synchronized in the IPR and an active edge on any synchronized signal causes the corresponding IPR bit to be set to 1.

The interrupt mask register (IMR) is a read/write register used to mask pending interrupts. A 1 in an IMR bit enables the corresponding pending interrupt to create an interrupt request.

External devices must assert interrupts for just over two CPU clock cycles (one I/O subsystem cycle) to be synchronized and detected. The interrupt must also deassert for at least two CPU clock cycles (one I/O subsystem cycle) for the next assertion to be recognized.

SCSI. Because of timing and loading constraints on the system bus, part of the I/O controller chip is used to implement the control logic between the NCR53C700 SCSI controller and the system bus. The SCSI interface block primarily handles the control signals while the system bus interface controls the address and data buffers between the system bus and the NCR chip.

For the address and data path, the address buffers (74543s) are in transparent mode for SCSI master transactions and in latch mode for slave transactions. The data buffers (74646s) are in transparent mode for master and slave write transactions and in latch mode for master and slave read transactions.

LAN. Part of the I/O controller chip is also used to implement the control logic between the Intel 82596DX LAN controller and the system bus. The LAN subblock controls the Intel 82596DX. The slave subblock controls slave operations (port and channel attention) and master subblock interfaces to the system bus on master operations.

A slave mode transaction is one that is initiated by the processor. The three slave mode transactions to the 82596DX are channel attention, port select, and hard reset.

When the 82596DX wants the bus for DMA master transactions, it asserts HOLD. The I/O controller chip LIOARB block arbitrates among the core I/O masters (LAN, SCSI, and Centronics DMA) and requests the system bus. The I/O controller chip then asserts HLDA to the 82596DX and lets the 82596DX keep the bus as long as it wants to (until HOLD is deasserted). The I/O controller chip also enables the address buffers from the core I/O system to the system bus.

Parallel Port and DMA. The Western Digital WD16C552 chip is used at the parallel port as the device driver/receiver interface. The Centronics compatible features and the additional functionality including that required by the HP ScanJet products are controlled through the I/O controller chip.

The parallel port's DMA controller inside the I/O controller chip is really a bidirectional data accumulator that masters transactions on the system bus. Its main purpose is to better match the relatively low-bandwidth parallel port to the high-performance system bus and give the port direct access to memory without affecting CPU performance very much. It buffers data for both the read and write functions of the built-in parallel port. It collects data from the system bus and sends it to the parallel port and vice versa. The system bus connection is a 32-bit-wide bus and the parallel port connection is 8 bits wide, so in addition to dealing with the high-speed system bus interface the DMA controller also does byte packing.

I/O Verification

The I/O verification strategy was divided into two parts; a stand-alone I/O system test using a tool called the bus exerciser, and a complete system test using the actual CPU, memory and system bus controller chip, and memory models. The stand-alone verification strategy will be discussed here. The full-system verification strategy is covered in the article on page 34.

The bus exerciser is a tool that verifies the correctness of the I/O subsystem design in the behavioral simulation environment. The bus exerciser is written in the MADL description language (see "Simulation Toolset," page 36). The bus exerciser is a kind of substitute for the CPU and the memory and system bus controller chip in that it executes test programs and generates corresponding transactions on the system bus to stimulate the I/O subsystem and check the correctness of the response from the I/O subsystem.

There were three main reasons for using the bus exerciser to verify the I/O subsystem design. First, the CPU, memory and system bus controller chip, and memory models were not available in the early phases of the I/O controller chip design. Using the bus exerciser eliminates the need for stable working models of the CPU and the memory and system bus controller chip before testing the rather independent I/O subsystem. Second, the bus exerciser models the CPU, the memory and system bus controller chip, the EISA card, and the memory subsystem in a simplified fashion. Thus simulation in the bus exerciser environment is about ten times faster than simulation with the CPU and memory and system bus controller chip models. Third, when running with the bus exerciser, we do not see cache misses and TLB misses. All the DMA data setup and result checking are done through the bus exerciser's virtual hardware in one clock cycle rather than using the CPU model running PA-RISC instructions to do data setup and result checking. As a result, we can issue much more intense and simultaneous I/O activities on the system bus, which helps stress the I/O subsystem.

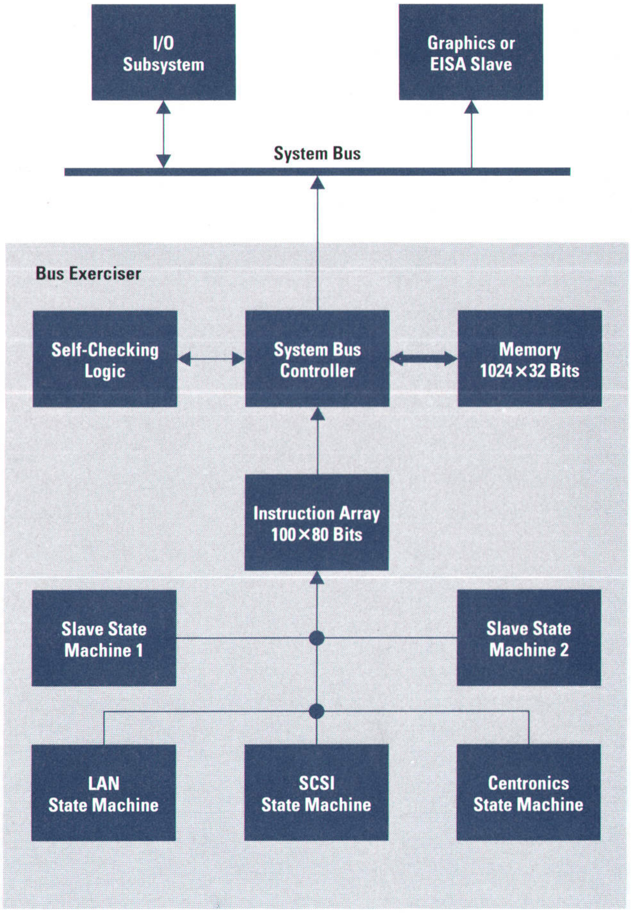


Fig. 3. Block diagram of the bus exerciser used for stand-alone I/O subsystem verification.

Fig. 3 is a block diagram of the bus exerciser. There are five main sections:

- Instruction array (100 instructions deep)
- Memory (4K bytes)
- System bus controller
- Self-checking logic
- Stress test state machines.

Instruction Array. To simplify the design, the bus exerciser implements an array that holds up to 100 instructions. The instruction array is initialized through guide vectors before the simulation begins. The bus exerciser can execute 1-, 2-, or 4-byte reads and 1-, 2-, or 4-byte pipelined or nonpipelined writes.

Each instruction in the instruction array is 80 bits wide. A simple programming language was developed to simplify test development. The program contains several fields. The first field specifies whether the operation is a read or a write. The second field specifies the address from which to read or to which to write. The next field specifies the data expected on a read or the data to write on a write operation. A fourth field might be used on a write operation to specify pipelined mode. An example of the programming language is as follows (the lines beginning with # are comments):

```
##### INTERRUPT TEST (int.inst1)
# Read interrupt pending register
r f0800008 0
# Read interrupt mask register
r f0800004 0
# enable interrupt mask
w f0800004 ffffffff p
# master clear
wb f082f000 0
# read interrupt mask reg
r f0800004 0
```

This programming language is translated to guide vectors using a simple HP-UX shell script. The resulting file contains 80 bits of information:

Bits 31:0 store the data information and bits 63:32 store the address information. Bits 79:76 are the opcode, such as read word or write byte. Bits 75:72 give pipeline information for the write operations. Bits 71:68 give information on whether or not to compare the read data. On a read instruction, if the expected data is known, then that data can be included in the data field (bits 31:0).

The table below further describes the instruction bits:

inst[79:76]

0000 : read word
0001 : write word
0010 : read byte
0011 : write byte
0100 : read 2 bytes
0101 : write 2 bytes

inst[75:72]

0000 : no-pipeline write
0001 : pipeline write

inst[71:68]

0000 : no-read data compare
0001 : read data compare

Listed below is an example of the first five entries of an instruction array corresponding to the sample program above.

```
input busex/inst[0] 0 %h0010f080000800000000
input busex/inst[1] 0 %h0010f080000400000000
input busex/inst[2] 0 %h1100f0800004ffffffff
input busex/inst[3] 0 %h3100f082f00000000000
input busex/inst[4] 0 %h0010f080000400000000
```

Memory. Inside the bus exerciser model is a memory array of 1024 entries by 32 bits. This is provided mainly because the SCSI and LAN models need memory to perform their DMA operations. The memory block shared between the LAN and the CPU is also hardwired into the memory model to minimize the tedious LAN DMA setup.

System Bus Controller. The system bus controller is the "brain" of the bus exerciser. The bus controller is implemented in a large, complex state machine. The bus controller fetches and executes instructions from the instruction array according to the system bus protocol. The bus exerciser is the bus master and the I/O controller chip is the slave while instructions are being fetched and executed. Instructions are fetched and executed until the instruction array is empty or until an I/O device requests the bus. When an I/O device requests the bus, the bus exerciser becomes the slave and the I/O controller chip is the bus master.

Listed below are the system bus functions supported by the system bus controller. Some functions were described in the "Instruction Array" section above. In this context, "host" is the bus exerciser and "guest" is the I/O system controller.

- One-byte, two-byte, or four-byte host-generated pipelined write
- One-byte, two-byte, or four-byte host-generated nonpipelined read/write
- One-byte, two-byte, or four-byte guest-generated pipelined read/write
- One-byte, two-byte, or four-byte guest-generated nonpipelined read/write
- Interrupt handling
- Error generation and error handling
- Bus arbitration.

Self-Checking Logic

Extensive self-checking capability was added to the bus exerciser model to minimize human intervention. For instance, the test program can specify the expected data for each read instruction. Upon completion of the read transaction, the bus exerciser will check the returned data with the expected data and raise an error flag if an inconsistency occurs.

For LAN, SCSI, and Centronics DMA operation, the bus exerciser will remember the data transferred from the memory to the I/O devices and later perform self-checking while the previously transferred data is looped back from the I/O devices to memory.

Stress Mode. Normally a test will be written in the bus exerciser programming language, converted to guide vectors, and run on the I/O system model. The tests are relatively short since the address queue can only hold up to 100 instructions. These short tests are usually used to test a particular device, such as the HP-HIL, LAN, or SCSI.

However, situations arise in which we would like to use the bus exerciser to generate I/O events randomly to stress the I/O subsystem over a long period of time. The bus exerciser uses stress mode to run randomly generated programs that can run for any specified length of time. The goal is for the bus exerciser to create different corner cases randomly to stress the I/O subsystem, and to perform the necessary self-checking without human intervention.

Five state machines were added to the bus exerciser model for the stress tests. Three of the state machines are DMA state machines for the LAN, SCSI, and Centronics devices. The state machines contain blocks of code that can be put into the instruction queue and executed by the bus exerciser. The blocks of code tell the DMA devices in the I/O subsystem to do a DMA read or write. In the case of the SCSI, for example, the bus exerciser's SCSI state machine can send a block of code to the instruction queue which, when executed, causes the SCSI device to perform a DMA read from memory. Once the SCSI device finishes the DMA read, it sends an interrupt back to the bus exerciser. Next, the bus exerciser's SCSI state machine sends a block of code to the instruction queue, which causes the SCSI device to perform a DMA write back to memory. Once the SCSI device finishes

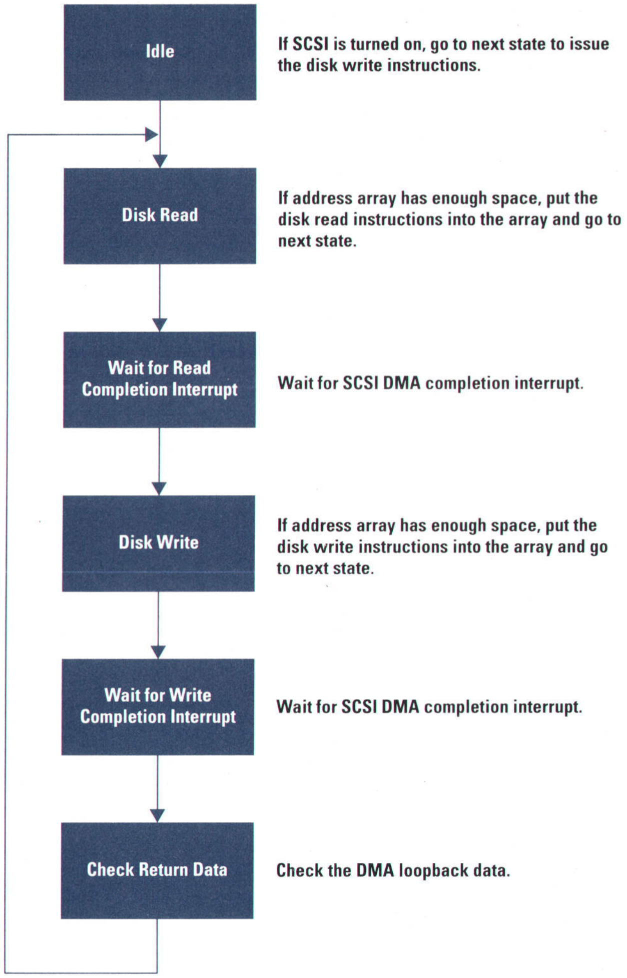

Idle
If SCSI is turned on, go to next state to issue the disk write instructions.
Disk Read
If address array has enough space, put the disk read instructions into the array and go to next state.
Wait for Read Completion Interrupt
Wait for SCSI DMA completion interrupt.
Disk Write
If address array has enough space, put the disk write instructions into the array and go to next state.
Wait for Write Completion Interrupt
Wait for SCSI DMA completion interrupt.
Check Return Data
Check the DMA loopback data.

Fig. 4. Operation of the bus exerciser's SCSI state machine.

the DMA write, it sends another interrupt back to the bus exerciser. Once the round trip DMA has been completed, the bus exerciser's SCSI state machine compares the data read from memory with the data written back to memory. If the comparison is not successful, an error is flagged. This process is repeated in a loop and the program can run for any length of time. During each loop, the DMA state machine gets its DMA transfer sizes and starting addresses randomly from a range of selections. An example of how the bus exerciser's SCSI state machine operates is illustrated in Fig. 4. The LAN and Centronics state machines behave in a similar fashion and all three state machines operate simultaneously but asynchronously. This results in a stressful simulation environment.

In addition to the three DMA state machines there are two other state machines that make the system even busier. These state machines also send blocks of code to the instruction queue, but the instructions access non-DMA devices. The bus exerciser issues a highly mixed pattern of instructions to the I/O subsystem by using this combination of three DMA state machines and two non-DMA state machines.

Although the bus exerciser was used as a debugging tool, it in no way replaced full system verification. It was still necessary to simulate operation with the CPU and the memory and system bus controller chip to catch system-level bugs. However, more than 95% of the I/O controller chip logic bugs were caught in the bus exerciser environment before integration with the CPU and memory and system bus controller chip models. The result was a clean I/O controller chip that worked the first time and required no chip turnarounds.

Acknowledgments

The authors wish to acknowledge the efforts of other I/O subsystem design team members: Wayne Ashby, Ping Hao, Danny Lu, Rob Snyder, Frank Lettang, and project manager Andy De Baets. Additional thanks to Alan Wiemann and his verification team members: Robert Lin, Steve LaMar, Ali Ahi, and Greg Burroughs.

Design Verification of the HP 9000 Series 700 PA-RISC Workstations

First a high-level system model was simulated and compared with a reference machine running both HP standard and pseudorandom test programs. Then the same tests were run on hardware prototypes. All chips were able to boot the operating system on first silicon.

by Ali M. Ahi, Gregory D. Burroughs, Audrey B. Gore, Steve W. LaMar, Chi-Yen R. Lin, and Alan L. Wiemann

The HP 9000 Series 700 workstations are based on a set of new custom VLSI chips.[1,2] To bring these products to market as quickly as possible, it was essential that these new chips function correctly the first time. System verification through simulation and prototype testing addressed this objective. This work proceeded in two phases. Phase 1 consisted of simulating a high-level behavioral model of the Series 700 system running a variety of test programs. Phase 2 consisted of running similar tests on initial prototypes of the system.

In addition to simulation and verification of the system, each new VLSI chip was subjected to stand-alone testing. A behavioral model of the chip was tested with vectors from vector generators without regard to the other chips in the system. Both test efforts were essential to achieving the goal of fully functional first silicon. This stand-alone testing was done by the chip design groups and that work is not described here.

The objectives for phase 1 of the system verification were twofold: (1) to demonstrate the correctness of the design according to HP standards, and (2) to find any functional bugs. Meeting the first objective meant running and passing the HP standard test programs (see "HP Standard PA-RISC Test Programs," page 35). Our strategy for achieving the second objective consisted of using whatever means were necessary to discover functional bugs in the behavioral models of the VLSI chips.

The objective for phase 2 of the system verification was to find any functional bugs that were not detected during phase 1. The phase 2 testing was focused largely on the new VLSI, but also served to discover bugs in other components in the system, including vendor-supplied components. Most of the test software developed for phase 1 testing was intended for eventual use in phase 2, but underwent substantial modification to run on actual hardware. System debugging software required that pseudorandom tests be modified to cooperate with the debugger in control of the hardware. Device drivers developed for the simulation models had to be modified or rewritten completely to work with actual I/O hardware.

Because of the considerable overhead of simulating an individual chip in a system, simple functional bugs confined to a single chip (error in branch prediction, floating-point arithmetic error, etc.) are best found in stand-alone testing, where the chip is surrounded by only enough support logic to allow it to function normally. The system model is primarily used to discover bugs that only show up when two or more of the chips are working together. Nonetheless, several bugs were found in system simulation that were confined

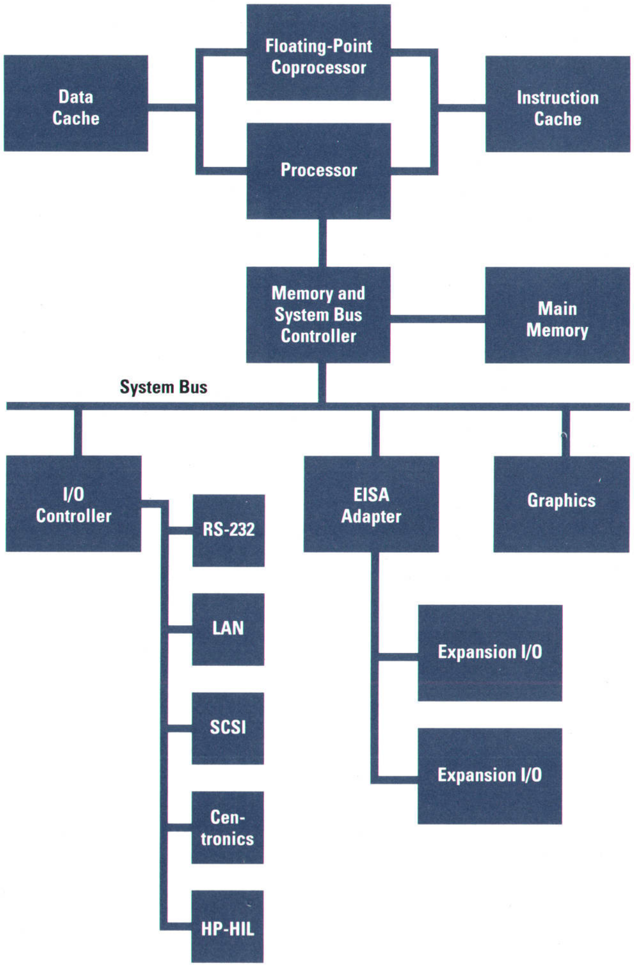


Fig. 1. HP 9000 Series 700 system block diagram.

HP Standard PA-RISC Test Programs

The PA-RISC verification process involves running a set of HP standard test programs on the model of the implementation, and creating and running code specific to the system. The standard code is run to ensure that the implementation minimally conforms to the architecture and to ensure that defects in previous implementations are not found in the current implementation. For our implementation, some of this standard code needed modification to bring it up to date with the current version of the PA-RISC architecture.

AVPs

During the verification of PA-RISC hardware a de facto standard set of test code is used to confirm that the implementation minimally conforms to the PA-RISC architecture. This set of test code is known as the architectural verification programs (AVPs). These AVPs are divided into two categories: processor and floating-point. One of the first major milestones reached during verification was the behavioral model's successful completion of all of the processor and floating-point AVP tests.

IDCs

Another group of de facto standard tests used during PA-RISC hardware verification is the implementation dependent codes (IDCs). This group of tests is composed of several small pieces of code written during the verification of previous PA-RISC implementations. This code usually stresses a characteristic (pipeline interlock, bypass, etc.) or exhibits a defect found in a previous implementation. These tests are run to ensure that the implementation being verified does not contain defects found in past implementations. However, the characteristics that some of these tests were written to stress in previous implementations will not necessarily be exercised in the implementation currently being verified.

Implementation-Specific Test Code

For our verification task, we developed our own set of tests to stress the memory and I/O controller portions of the system. Operations performed by these tests typically included simple I/O register reads and writes, small DMA transfers, or error condition creation. These tests typically did not create much activity on the system bus, but ensured that the basic functionality of individual pieces of the I/O system was working correctly. Once the basic functionality was verified, the more complex and stressful testing with the BPS I/O tests (see accompanying article) could proceed.

within an individual chip. The point is that system simulation is not a substitute for stand-alone testing, but rather supplements it.

Scope of the Problem

The system verification included behavioral models for the following chips and boards: the processor chip, the floating-point coprocessor chip, the memory/system bus controller chip, the memory/system bus interface chip, the I/O controller chip, the processor board, and the system I/O board (see block diagram, Fig. 1).

Our search for functional bugs concentrated on these chips and boards. The complete system model also included behavioral models for cache RAM chips, memory arrays, I/O devices, and clock circuits. It was not our specific intent to discover bugs in these models, but many were found as a consequence of our testing.

Challenges

A challenge was presented by several enhancements included in the advanced PA-RISC 1.1 architecture of the new workstation. This new revision of the architecture required substantial changes to existing test programs and tools. Some of the new features that required verification included block TLB (translation lookahead buffer) entries, new registers to reduce CPI (cycles per instruction) on system interrupts, and new floating-point instructions.

Another challenge was presented by the geographical distribution of people. Some of the chip design groups were located in Colorado, while other design groups and the verification group were in California. This meant that we had to rely extensively on HP's networking capabilities to exchange models, test files, simulation results, and bug reports. Extensive use of telephone and videoconference facilities was necessary in addition to frequent travel between locations. In spite of the considerable barriers presented by geographical separation, a high degree of teamwork and cooperation was achieved.

Methodology

The strategy for accomplishing the objectives consisted of first running and passing the HP standard test programs on the system model. These include tests called architectural verification programs (AVPs) and tests called implementation dependent code (IDCs). These tests address only processor and floating-point functionality and some of the memory access functionality. This was followed by additional testing using a pseudorandom test technique (BPS, see below), augmented by individual tests to check particular test cases. Testing of the I/O chips required the creation of many hand-written tests (assembly language programs) which were run and passed before continuing with more pseudorandom I/O testing.

Completion Criteria

Some measure of completeness was needed both to quantify our objectives and to enable us to measure progress towards achieving the objectives. This work had two basic completion criteria: (1) pass all AVPs, IDCs, and newly created I/O tests, and (2) run some fixed amount of new pseudorandom tests without a bug-caused failure in any model. The second criterion is analogous to the 24-hour CHO (continuous hours of operation) metric used for operating system testing. For phase 1, the criterion was to run new tests on all available simulation engines for at least one week. For phase 2, the criterion was to run new tests on a prototype for 48 hours. The one-week and 48-hour time limits are somewhat arbitrary, as is the 24-hour limit for operating system testing.

Reference and Target Machines

Self-checking verification programs, such as the PA-RISC AVPs (see below), have built-in routines that check for correct program execution. Self-checking programs have many strengths, but suffer from one fundamental weakness: it is difficult to write a self-checking test for an unforeseen failure mode. Our system verification process used a reference machine verification methodology to supplement the existing self-checking verification programs. In our reference machine methodology, a Series 700 simulator was developed from the PA-RISC simulator (see below). Each verification program was run against the Series 700 simulator and the behavioral model. The test was considered run successfully if a particular subset of the machine state of the model agreed with that of the simulator at the end of the two program executions.

Simulation Toolset

HP's ChipBuster toolset and other tools were used to create and run tests on the Series 700 system model.

ChipBuster is a set of tools used to create models, run simulations, do schematic capture, generate simulation vectors, run simulations and do artwork layout design.

MADL is the modeling language in the ChipBuster toolset that was used for many of the models for the Series 700.

Guide is the environment in ChipBuster that is used to compile and then run simulations on the MADL models.

Piglet is a schematic capture system that is part of the ChipBuster toolset. The captured schematics were then converted to structural models and incorporated into the system model for the Series 700.

BEST is one of the simulators that is part of the ChipBuster toolset. It runs under the Guide environment. BEST is short for BEhavioral, functional Specified Timing simulator.

The Design Capture System (DCS) is another schematic capture system that was used for capturing the board level schematics on the Series 700. DCS is not in the ChipBuster toolset, so a program was written to convert DCS output to structural MADL.

The Design Database Language (DDL) is a programming language that runs under DCS and provides a way to access the DCS database. A DDL script was used to access the DCS database and turn the schematics into structural MADL models.

PLADO is a modeling language that describes the behavior of PLAs (programmable logic arrays).

We used two techniques to compare state. In the simple case, a test was run on both the Series 700 simulator and the behavioral model, the final state was dumped to a file in both cases, and the states of the simulator and the model were compared. In the more complicated case, the program was first run on the simulator, an execution trace and the final machine state were saved and incorporated into a self-checking program, and the self-checking program was run on the behavioral model.

PA-RISC Simulator

The PA-RISC simulator is a functional level instruction set simulator for the PA-RISC architecture.[3] The simulator runs programs at a rate of thousands of instructions per second, and is a much more pleasant environment for developing and debugging verification test source code than the behavioral model. The simulator is a correct implementation of the PA-RISC architecture, so it can serve as a reference machine for processor-oriented programs. To support verification of PA-RISC 1.1 processors, we made the following enhancements to the PA-RISC simulator:

- Upgrade to PA-RISC 1.1
  - Additional floating-point registers and instructions
  - Cache extensions
  - Page size increase from 2K to 4K bytes
  - Block TLB entries
  - Shadow registers
- Support for additional processor organizations
  - Virtual cache and arbitrary cache indexes
  - Additional I/O modeling capabilities
- Additional hooks for scheduled interrupts and I/O.

Simulation Tools

We used the HP ChipBuster toolset to do system simulation (see "Simulation Toolset," at left), specifically the BEST simulator and the Guide environment (BEST and Guide are parts of the ChipBuster toolset). However, the various parts of the system were modeled in different ways, and each part had to be incorporated into one system model that could run under Guide and BEST. The system simulation model came from a variety of sources, and we used a number of tools to create the model and to run simulations.

Behavioral models for most of the VLSI chips were written in the ChipBuster MADL modeling language. Their structural connectivity was added using Piglet, a schematic editing tool contained within ChipBuster. The VLSI chips modeled in this fashion included the processor chip, the floating-point coprocessor chip, and the memory and system bus controller chip. The VLSI chip models were connected by a top-level structural model, which was created from a schematic of the CPU board captured by DCS—HP's Design Capture System. DCS is not part of ChipBuster, but the DCS database can be accessed using a programming language called DDL, which runs under DCS. We used a DDL script to change the DCS schematic to structural MADL. The VLSI models were given to us by the chip designers, but models for other parts of the CPU board were created specifically to support design verification. These parts included the instruction and data caches, the clock circuitry, an EISA model, and a graphics model.

The I/O controller chip and the I/O system were modeled differently from the rest of the system. The state diagrams for the I/O controller were converted to state equations using the PLADO tool, a modeling language that describes the behavior of PLAs (programmable logic arrays). The state equations were converted to MADL structural and behavioral models. The rest of the I/O system was modeled using a combination of behavioral and structural models, and more PLADO models for the various I/O devices such as the SCSI chip and the LAN chip. Once the models were converted to MADL they were attached to the rest of the system model by hand-editing the top-level structural MADL model that was generated from DCS for the CPU board. The editing was done to include the top-level I/O signals, which were attached to the CPU board via the system bus connector.

We had a variety of hardware resources on which to run system simulations. The verification group had six HP 9000 Model 835 systems and three Model 855s for system simulation. One of the Model 835s contained the system model and was not used for simulation, but was used for bringing up new models and as an NFS (Network File System) server. The other machines were NFS-mounted to the discs containing the model.

Simulations were managed by means of a queuing system which we developed.

Pseudorandom Test Technology

Unlike other test suites which are written entirely by hand, the pseudorandom test programs used for Series 700 system verification were created by a very powerful test generation tool called the Busy-system Program Synthesizer, or BPS. A test is a collection of rules, which are consumed by the program generator to create random programs. The rules define the kind of instructions allowed in a test. The program generator creates programs that contain instructions defined by the rules in random order. One set of rules can yield many different programs. BPS generates virtually countless test programs and allows us to exercise the system exhaustively.

The verification process is an iterative process (see Fig. 2). It starts with using the program generation tool BPS to create a test program from a collection of rules. This test program is then run on the target hardware model (or prototype) and on the reference machine, that is, the PA-RISC simulator. The reference machine verification techniques described under "Methodology" above are applied. When differences between the target and reference machines occur, the execution traces are compared and the potential design flaws are identified. When there are no differences between the target and reference machines, the test is discarded. This process is repeated until all functional bugs have been found.

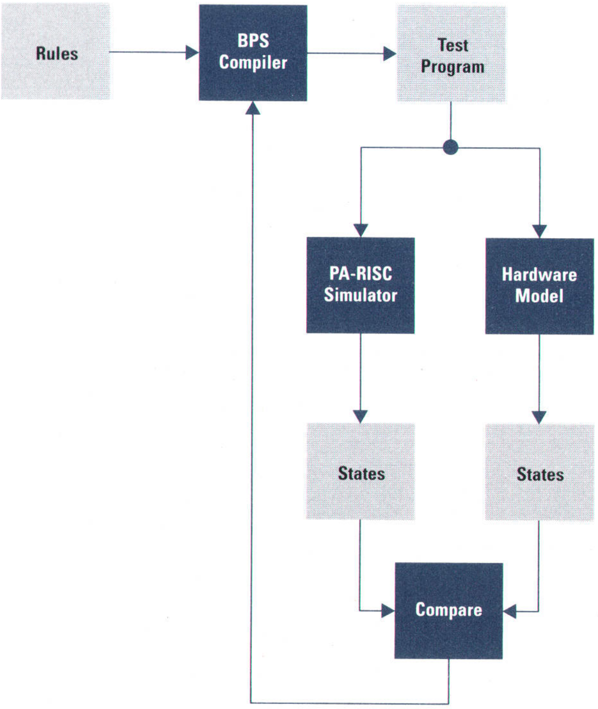


Fig. 2. Verification process.

Why BPS?

BPS can reduce the time it takes to develop test software. Traditionally, a test was written specifically for every system function or group of functions under test. Test program development was a very demanding part of the verification process. The extent to which verification could be performed before the design release was limited by the amount of test software available. As the complexity of VLSI increases and new-product development time decreases, the need for a better and faster way to develop test programs grows. BPS fills this need.

BPS also breaks coding habits. People tend to develop habits after doing certain things many times. Once a habit is developed, code segments written for similar tasks tend to follow the same coding style. Programming habits limit the variety of operations that are exercised on a target system. BPS avoids this problem.

Thirdly, BPS can stress a target system as well as real-world software. At the software level, there is a boundary between system code, user code, and firmware code. At the hardware level, that distinction disappears. The code from all three sources is intermingled in front of the CPU. The system should be able to perform any combination of instructions under any system conditions. Software that can model the mixture of operations coming from all those three code sources is needed, especially during the development of new hardware, when it is not possible to run the real operating system and user software. BPS satisfies this requirement.

Finally, debugging may be easier for BPS than it is for other test software or for a real operating system. Linear code is easier for a human to follow than repetitive code. (Handwritten code that has loops can be made easier to debug by unfolding the loops.) During the turn-on phase of new hardware, the effort to debug the hardware using the real operating system is tremendous because of the amount and the complexity of the operating system code. BPS generates nonlooping (but branchy) code that stresses the system just as much as the real software.

Basic BPS Concepts

Programs generated by BPS have four basic components:

- A system initialization routine
- A page of random instructions
- One or more pages of random data
- A group of hand-coded routines.

The random instructions are created based on the rules given to BPS. The random instruction page is the only part that varies among different programs generated by BPS. The data in the random data pages is arbitrary. Random data is pregenerated. The same random data is included in every test program. The hand-coded routines handle various trap conditions. System initialization sets the system to certain states for each test.

BPS tests are used for verification of three major system functional units: the processor, the floating-point coprocessor, and the I/O subsystem. The I/O tests ultimately verify the entire system.

Processor and Floating-Point Coprocessor Tests

A typical BPS CPU/FP program has four sections (see Fig. 3): initialization code, random instructions, random data, and trap handlers.

At the beginning of a program, the initialization code sets up the system resources (control registers, general registers, floating-point registers, translation lookaside buffers or TLBs, etc.) to certain states. The initialization code runs in real mode. It turns on the virtual mode before execution

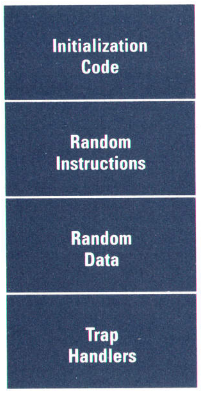


Fig. 3. Structure of a BPS test.

switches to the random code. The necessary setup for virtual addressing mode includes TLB entries, address queues, the processor status word, and so on. The switch from real to virtual mode takes place on the last instruction in the initialization code. Once in virtual mode, execution of instructions on the random code page continues indefinitely until a recovery counter trap occurs.

BPS generates only one random code page for every program. All virtual instruction pages are mapped to that same code page. The offset part of a virtual address is the offset of a corresponding physical page. When a virtual address goes across a page boundary as a result of sequential execution, an ITLB (instruction TLB) miss trap occurs. The trap handler maps the new virtual address to the same random code page, and execution falls back to the top of this page. When a virtual address goes to another unmapped virtual page as the result of executing a branch, again the new virtual address is mapped to the same random code page by the trap handler. A virtual address may point to anywhere in the virtual space, but the mapped physical address never goes out of the random code page. The program is stopped by the recovery counter trap after some desired number of instructions are executed on the random code page.

The strategy used for random instructions is also used by the DTLB (data TLB) trap handler for random data. To verify the TLB protection function, random access rights are used when inserting a new TLB entry. Thus, a TLB miss trap is often followed by a TLB protection trap. To recover from the protection trap, the protection trap handlers supply good TLB entries and good PIDs (process IDs). The illegal instruction trap handler sets the IPSW to nullify the illegal or undefined instructions. The assist exception trap handler retries the exception instructions. The recovery counter trap handler stops a test and also invokes a self-checking routine for floating-point tests.

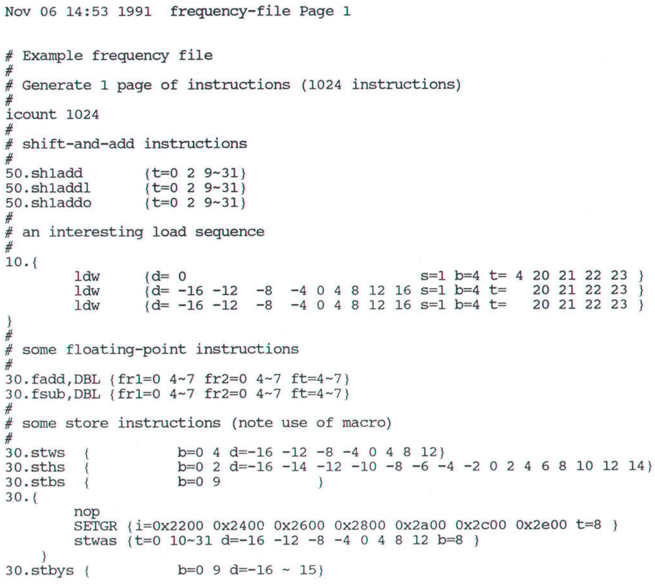
Target versus Reference. A test is passed if, when the test stops, the contents of the general registers of the target machine match those of the reference machine. Although the final state of the general registers may seem to represent only the operations near the end of a test, in many cases when the target machine operations begin to depart from the reference machine operations, the differences are propagated all the way to the end of the test.

In this project, the floating-point registers were not visible on the target machine. Instead, the test software stored all floating-point registers into memory and compared them with reference values stored in another memory area. The reference values were created during test program generation and were included as part of the test program.

Frequency Files. BPS generates random codes according to user-specified rules. The file that holds these rules is called a frequency file (see Fig. 4). The rules can be general or specific. A rule describes what PA-RISC instructions are to be generated in the random code. This may be a specific instruction, any instruction in a particular group, or a specific instruction sequence.

Instructions described in a frequency file will appear many times in random order in the code generated by BPS. The number of occurrences is controlled by two things: the frequency and the total number of instructions to be generated. The frequency for each user-specified rule is a relative weight assigned by the test developer. An instruction described by a rule with twice the weight of others will appear twice as often in the resulting code. The statement icount followed by a number in a frequency file describes the total number of instructions to be created for a test program.

Rules for instructions that perform similar hardware operations are grouped together to form basic tests. For example, ALU is a basic test category. The frequency file describes only ALU (arithmetic logic unit) instructions. The random code generated from that frequency file contains all of the ALU instructions described in that file in random order. Other basic tests include CPU memory load, CPU memory store, unconditional branch, conditional branch, floating-point operations, floating-point memory operations, and so on.



```
Nov 06 14:53 1991  frequency-file Page 1


# Example frequency file
#
# Generate 1 page of instructions (1024 instructions)
#
icount 1024
#
# shift-and-add instructions
#
50.sh1add       {t=0 2 9~31}
50.sh1addl      {t=0 2 9~31}
50.sh1addo      {t=0 2 9~31}
#
# an interesting load sequence
#
10.{
        ldw     {d= 0                            s=1 b=4 t= 4 20 21 22 23 }
        ldw     {d= -16 -12  -8  -4 0 4 8 12 16 s=1 b=4 t=   20 21 22 23 }
        ldw     {d= -16 -12  -8  -4 0 4 8 12 16 s=1 b=4 t=   20 21 22 23 }
}
#
# some floating-point instructions
#
30.fadd,DBL {fr1=0 4~7 fr2=0 4~7 ft=4~7}
30.fsub,DBL {fr1=0 4~7 fr2=0 4~7 ft=4~7}
#
# some store instructions (note use of macro)
#
30.stws  {          b=0 4 d=-16 -12 -8 -4 0 4 8 12}
30.sths  {          b=0 2 d=-16 -14 -12 -10 -8 -6 -4 -2 0 2 4 6 8 10 12 14}
30.stbs  {          b=0 9              }
30.{
        nop
        SETGR {i=0x2200 0x2400 0x2600 0x2800 0x2a00 0x2c00 0x2e00 t=8 }
        stwas {t=0 10~31 d=-16 -12 -8 -4 0 4 8 12 b=8 }
    }
30.stbys {          b=0 9 d=-16 ~ 15}
```

Fig. 4. An example of a frequency file.

Debugging Tools

In phase 1 of the HP 9000 Series 700 system verification, most of the debugging was done using the execution history files of the model to compare the model with the reference machine. A depiper block was implemented inside the processor behavioral model. It collected the addresses and data associated with the various stages of an instruction's execution and wrote them at the instruction's last pipe stage. Traces of signal values were available, taken either at the pin boundary or at internal nodes of the system components. A custom difference locator was used to compare the flow of the test programs on an instruction-by-instruction basis (see Fig. 1).

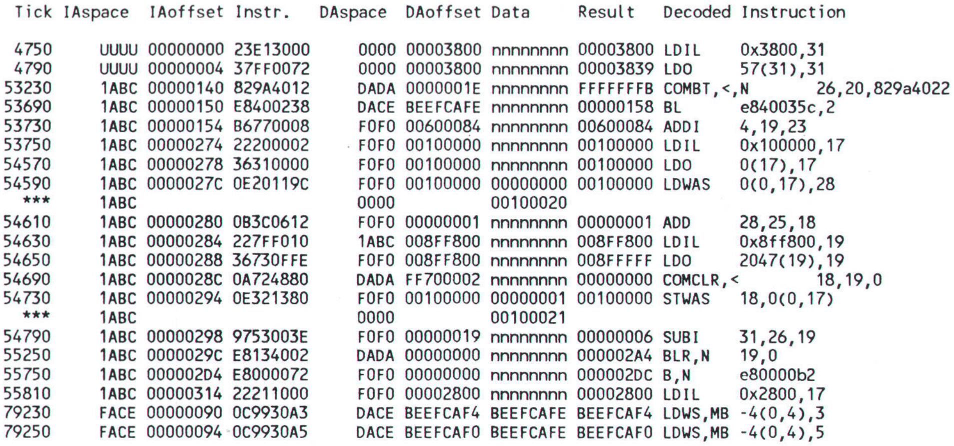
In phase 2 the remote debugger RDB was used for running the test programs on the prototype. The HP 16500 logic analyzer was used to trace system bus transactions and the HP AWAX toolset was used for cache and Pbus activities. RDB, the remote debugger, is a program that downloads an executable file from an HP 9000 Series 300 computer into the memory of a prototype. Its interface and control flow are similar to that of the PA-RISC simulator.[1] AWAX is an HP internal processor monitor toolset for the PCX chipset family, to which the Series 700 chips belong. Functionally, AWAX consists of five parts: data capture board, depiper, state analyzer, performance analyzer, and controller.

The bus monitor is a program, used during phase 2, that takes a trace of the system bus and provides a statistical analysis of the activity and utilization of the bus (see Fig. 2).

Init_tlb, init_cache, and similar programs initialize the TLB and registers in the CPU. This is done to bring the model to a particular state without having to go through the whole simulation sequence again.

Reference

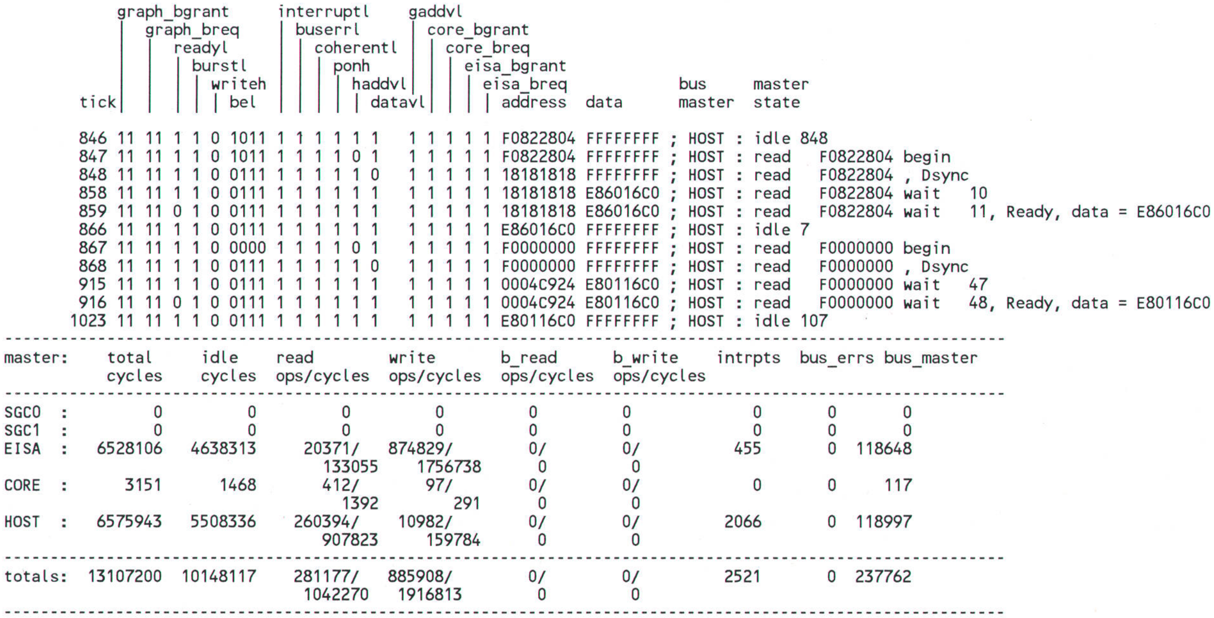
1. D Magenheimer, "Remote Debugger," *Hewlett-Packard Journal*, Vol. 37, no. 8, August 1986, p. 43.



```
 Tick IAspace  IAoffset Instr.   DAspace DAoffset Data     Result   Decoded Instruction

 4750     UUUU 00000000 23E13000    0000 00003800 nnnnnnnn 00003800 LDIL    0x3800,31
 4790     UUUU 00000004 37FF0072    0000 00003800 nnnnnnnn 00003839 LDO     57(31),31
53230     1ABC 00000140 829A4012    DADA 0000001E nnnnnnnn FFFFFFFB COMBT,<,N         26,20,829a4022
53690     1ABC 00000150 E8400238    DACE BEEFCAFE nnnnnnnn 00000158 BL      e840035c,2
53730     1ABC 00000154 B6770008    F0F0 00600084 nnnnnnnn 00600084 ADDI    4,19,23
53750     1ABC 00000274 22200002    F0F0 00100000 nnnnnnnn 00100000 LDIL    0x100000,17
54570     1ABC 00000278 36310000    F0F0 00100000 nnnnnnnn 00100000 LDO     0(17),17
54590     1ABC 0000027C 0E20119C    F0F0 00100000 00000000 00100000 LDWAS   0(0,17),28
  ***     1ABC                      0000          00100020
54610     1ABC 00000280 0B3C0612    F0F0 00000001 nnnnnnnn 00000001 ADD     28,25,18
54630     1ABC 00000284 227FF010    1ABC 008FF800 nnnnnnnn 008FF800 LDIL    0x8ff800,19
54650     1ABC 00000288 36730FFE    F0F0 008FF800 nnnnnnnn 008FFFFF LDO     2047(19),19
54690     1ABC 0000028C 0A724880    DADA FF700002 nnnnnnnn 00000000 COMCLR,<          18,19,0
54730     1ABC 00000294 0E321380    F0F0 00100000 00000001 00100000 STWAS   18,0(0,17)
  ***     1ABC                      0000          00100021
54790     1ABC 00000298 9753003E    F0F0 00000019 nnnnnnnn 00000006 SUBI    31,26,19
55250     1ABC 0000029C E8134002    DADA 00000000 nnnnnnnn 000002A4 BLR,N   19,0
55750     1ABC 000002D4 E8000072    F0F0 00000000 nnnnnnnn 000002DC B,N     e80000b2
55810     1ABC 00000314 22211000    F0F0 00002800 nnnnnnnn 00002800 LDIL    0x2800,17
79230     FACE 00000090 0C9930A3    DACE BEEFCAF4 BEEFCAFE BEEFCAF4 LDWS,MB -4(0,4),3
79250     FACE 00000094 0C9930A5    DACE BEEFCAF0 BEEFCAFE BEEFCAF0 LDWS,MB -4(0,4),5
```

Fig. 1. Difference locator output.



```
         graph_bgrant      interruptl    gaddvl
         | graph_breq      | buserrl     | core_bgrant
         |  | readyl       |  | coherentl|  | core_breq
         |  |  | burstl    |  |  | ponh  |  |  | eisa_bgrant
         |  |  |  | writeh |  |  |  | haddvl|  |  | eisa_breq             bus      master
     tick|  |  |  |  | bel |  |  |  |  | datavl|  |  | address  data      master   state

      846 11 11 1 1 0 1011 1 1 1 1 1 1   1 1 1 1 1 F0822804 FFFFFFFF ; HOST : idle 848
      847 11 11 1 1 0 1011 1 1 1 1 0 1   1 1 1 1 1 F0822804 FFFFFFFF ; HOST : read   F0822804 begin
      848 11 11 1 1 0 0111 1 1 1 1 1 0   1 1 1 1 1 18181818 FFFFFFFF ; HOST : read   F0822804 , Dsync
      858 11 11 1 1 0 0111 1 1 1 1 1 1   1 1 1 1 1 18181818 E86016C0 ; HOST : read   F0822804 wait   10
      859 11 11 0 1 0 0111 1 1 1 1 1 1   1 1 1 1 1 18181818 E86016C0 ; HOST : read   F0822804 wait   11, Ready, data = E86016C0
      866 11 11 1 1 0 0111 1 1 1 1 1 1   1 1 1 1 1 E86016C0 FFFFFFFF ; HOST : idle 7
      867 11 11 1 1 0 0000 1 1 1 1 0 1   1 1 1 1 1 F0000000 FFFFFFFF ; HOST : read   F0000000 begin
      868 11 11 1 1 0 0111 1 1 1 1 1 0   1 1 1 1 1 F0000000 FFFFFFFF ; HOST : read   F0000000 , Dsync
      915 11 11 1 1 0 0111 1 1 1 1 1 1   1 1 1 1 1 0004C924 E80116C0 ; HOST : read   F0000000 wait   47
      916 11 11 0 1 0 0111 1 1 1 1 1 1   1 1 1 1 1 0004C924 E80116C0 ; HOST : read   F0000000 wait   48, Ready, data = E80116C0
     1023 11 11 1 1 0 0111 1 1 1 1 1 1   1 1 1 1 1 E80116C0 FFFFFFFF ; HOST : idle 107
-----------------------------------------------------------------------------------------------------------
master:    total     idle    read        write       b_read      b_write     intrpts  bus_errs bus_master
           cycles    cycles  ops/cycles  ops/cycles  ops/cycles  ops/cycles
-----------------------------------------------------------------------------------------------------------
SGC0  :         0         0           0           0           0           0         0      0        0
SGC1  :         0         0           0           0           0           0         0      0        0
EISA  :   6528106   4638313     20371/    874829/          0/          0/         455      0   118648
                               133055     1756738          0           0
CORE  :      3151      1468       412/        97/          0/          0/           0      0      117
                                 1392         291          0           0
HOST  :   6575943   5508336    260394/     10982/          0/          0/        2066      0   118997
                               907823      159784          0           0
-----------------------------------------------------------------------------------------------------------
totals:  13107200  10148117    281177/    885908/          0/          0/        2521      0   237762
                              1042270    1916813           0           0
-----------------------------------------------------------------------------------------------------------
```

Fig. 2. Bus monitor output.

In most cases, creating a new test category is a matter of writing new frequency files or combining existing ones. For example, CPU memory load and store can be combined, as can CPU and floating-point memory load and store.

Frequency file development is a peculiar form of programming. These files must be designed with care. A rather strict register convention was adopted to allow for address alignment, procedure calls, floating-point data subsets, and other requirements.

In summary, although the frequency files remain the same for a given test category, the number of test programs that can be generated is very large, and it is easy to make new test categories. As a result, the number of test programs available can grow both horizontally and vertically.

Instrgen. Fig. 5 shows the BPS test generation process. The core part of the BPS test generator is the program instrgen, which takes frequency files from a test directory and generates the random instructions. The BPS test generator combines the random code generated by instrgen with the random data and hand-coded routines supplied to it. The test code is in assembly language. It is compiled and run on the target machine.

Instrgen is independent of the PA-RISC implementation. The only thing instrgen knows about PA-RISC is the mnemonic format of the PA-RISC instruction set. It doesn't know register dependences, CPU pipelines, or the effects of instruction execution. Compared with other random code generator programs, the code generation algorithm is simpler.

Instrgen allows the test designer to describe a PA-RISC instruction using either the instruction mnemonic or the instruction type. Instruction types are defined by type strings. Every PA-RISC instruction can be described by a major type followed by zero or more subtypes. For example, the major type for ldwas is memory operation. The subtypes are load operation, word access, short displacement, and absolute. If only memory operation is specified, the instructions described include all memory load and store instructions. If the type memory operation is followed by the subtypes load operation and word access, the instructions described include only ldws, ldwas, ldwx, ldwax, fldws, and fldwx.

One can also tell instrgen what values are valid for fields of an instruction or instruction type. Fields are named using the same field mnemonics that are used in the processor handbook. A field can receive a single value, a list of values, or a range of values. Instrgen randomly picks a value from the list or range for every instruction. Random values are assigned to fields that have no specified value, list, or range.

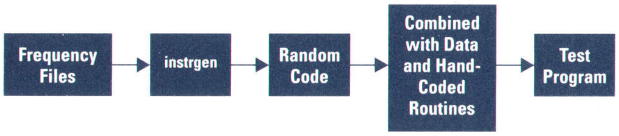


Fig. 5. BPS test generation process.

Floating-Point. Floating-point tests are classified similarly to CPU tests, that is, as memory operations, floating-point operations (flops), and combinations of memory operations and flops. The frequency files for the basic floating-point tests can be combined with CPU tests to form new tests.

BPS floating-point tests were designed for testing the protocol between the CPU and the floating-point coprocessor, not for checking the correctness of the floating-point arithmetic operations. Data and register use were carefully chosen to avoid NaNs (not a number). Exceptions occurred mostly during type conversion. The trap handler retries all excepted instructions. All floating-point instructions are included in this test suite.

Input/Output Subsystem Tests

The BPS I/O subsystem tests used in this project focused on the functional verification of the HP VLSI, that is, the I/O controller and memory/system bus controller chips, not the off-the-shelf I/O controller chips or the peripheral devices. The purpose was to simulate heavy system bus transactions along with CPU-to-memory-controller transactions. Using various I/O transactions, a test exercises the major functions of the I/O subsystem, that is, system bus arbitration, data transfer, and interrupt delivery. Typical I/O transactions are DMA transactions for the SCSI, LAN, EISA, and Centronics interfaces, character transactions for the RS-232 and HP-HIL interfaces, and special graphics transactions for graphics.

Like a BPS CPU/FP test, a BPS I/O test has four basic parts: initialization code, random code, random data, and trap handlers. In addition, a BPS I/O test has device drivers, I/O interrupt service routines, and an I/O request server. The random code section of an I/O test contains a number of procedure calls to the I/O request server. Every call represents one I/O request. The arguments passed to the I/O request server are:

- Device type
- I/O transaction length
- Device options.

The I/O request server checks the device status and starts the I/O transaction.

Round Trips. BPS I/O tests check the correctness of every I/O transaction. Every I/O transaction from memory to a device is always followed by an I/O transaction in the opposite direction. Data transferred to devices is kept in a dedicated memory area. Another memory area is reserved for data transferred back from I/O devices. At the end of an I/O round trip, the data in these memory areas is compared. Unlike processor tests, the recovery counter does not determine the length of I/O tests. Instead, a test stops when an error is encountered or after a specified number of I/O round trips.

Queues and Interrupt Services. A major goal in testing the I/O subsystem is to keep it busy. The software issues multiple I/O transactions, one for every device. The I/O subsystem is programmed to signal the completion of an I/O transaction through an external CPU interrupt. When an I/O transaction is completed, a new transaction is started immediately. A request queue is maintained for every device for continuously generating I/O transactions.

Metrics

Throughout the course of this project a number of state variables (metrics as a function of time) were collected. By logging every simulation event (start, stop, pass/fail status) to a file, we were able to determine a number of progress-related variables automatically. State variables were measured weekly and reported monthly. The primary function of the weekly metrics was to provide some accurate gauge of project progress. A number of variables were measured automatically from simulation logs, primarily indicating progress in passing test suites on models. Others indicated the size and speed of simulations and the effectiveness of the various test suites. A record was kept of how many bugs were discovered by each suite of tests. The results are shown in Fig. 1.

In general, test suites that were run first had the best opportunity to find the most bugs. For example, the test suite labeled "Hand", that is, the hand-coded tests, was run before attempting the more complex BPS tests. Consequently, that suite of tests had an opportunity to find the greater number of bugs, while the pseudorandom BPS tests found the more subtle bugs. We found that the pseudorandom test technique generally found more bugs with less effort.

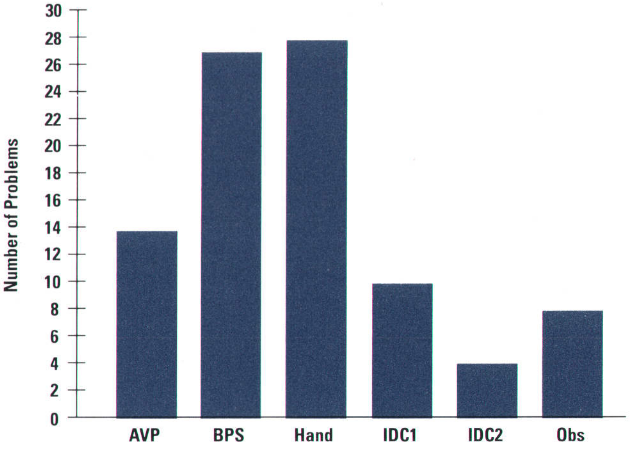


Fig. 1. Number of bugs found by each test class.

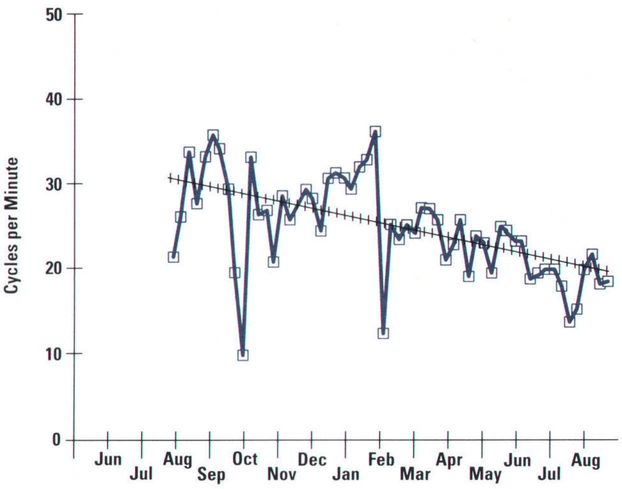


Fig. 2. Simulation speed declined as more components were added to the model.

Another metric was the average simulation speed. This was used on a regular basis to estimate the number of simulation servers needed to complete a given simulation objective as well as the expected length of time needed to simulate a test. Actual simulation speed as measured on a weekly basis as shown in Fig. 2.

Simulation speed declined as more and more components were added to the model. The actual simulation capacity varied from week to week as machines were added to and removed from the pool of available servers.

These metrics were extremely useful for managing the aggressive schedule of the Series 700 system verification and provided a basis for planning future developments.

The life of an I/O transaction begins in the random code section. A randomly generated call to the I/O request service is passed in an I/O transaction request. The transaction is started if the device is idle. Otherwise, it is queued. When the device finishes the current request, the request at the top of the queue is executed. A request represents a round-trip transaction and therefore two interrupts, one for the transaction from memory to device, and one for the transaction from device to memory. After the data transferred back from the device is checked, the current request is complete and the request on the top of the queue is started.

The I/O request server maintains the request queues. The I/O drivers set up the devices and start the transaction. The external interrupt handler dispatches the interrupts to the interrupt service routines. The interrupt service routines check the source and target data and keep the devices running.

With interrupt signaling and request queuing, BPS I/O tests can keep multiple I/O transactions active at the same time. The arbitration logic in the I/O controller and memory/system bus controller are exercised for timesharing the system bus for various I/O transactions. While I/O transactions are progressing, the CPU is available for running the random code. As a result, more requests are queued. How busy the system bus can be depends on how fast the I/O queues grow and how fast the interrupt can be served.

Gremlins. There is no reason why I/O tests cannot be combined with CPU/FP tests. We did exactly that, keeping both I/O and the CPU busy all the time. In addition, a number of gremlins were added in the tests—for example:

- Double-bit error
- Memory-controller-sourced low-priority machine check
- Memory refresh rate change
- Bus arbitration change
- Random read and write of I/O registers
- Random flush of the I-cache
- High-priority machine check during DMA.

Phase 1 versus Phase 2. Because of the complexity of the I/O subsystem components (LAN, SCSI, EISA, graphics, etc.) and the lack of exact models compatible with our simulation environment, a simplified behavioral model was written for each component. These models followed the interface protocol, but did not contain the full functionality or represent the timing characteristics of every component.

The timing behavior of these I/O component modules, unlike the actual chip, was close to that of the CPU. Thus, by the time CPU would go through scheduling a transaction (e.g., DMA), initiating it, and returning from the scheduling code, the transaction would almost be completed and the completion interrupt would soon be received.

For this reason, the BPS I/O phase 1 tests were run in burst mode. Burst mode means that a number of I/O activities are scheduled in a queue and are all started at the same time. This guarantees a greater degree of randomness and concurrency in the I/O activities.

BPS I/O phase 2 was different, since we were able to schedule I/O transactions one at a time and perform other scheduling or interrupt services before the transaction was completed. This was because of the overhead and latency of the actual system components.

Another major difference between the BPS I/O phase 1 and phase 2 tests was the number of I/O transactions requested. Phase 1 programs ran at about one cycle every two seconds in the simulation environment, while phase 2 programs, being run on the real hardware, ran eighty million times faster. Thus there were many thousands of times more I/O tests in phase 2 than in phase 1.

Measurements. During BPS testing we took some measurements of the number of reads and writes on the system bus connection. This gave us a measure of how busy the I/O subsystem was during testing. On the average, for each sample of 1024 cycles, there were 20.58 reads and 8.74 writes, adding up to 29.32 reads and writes.

For comparison, similar measurements were taken on the system running the HP-UX operating system. On the average, there was less than one read or write per sample. Two processes were then added: nfsperf between an HP 9000 Series 700 and a Series 300 and the X11perf benchmark on the Series 700. The average numbers of reads for nfsperf and X11perf were 14 and 13 respectively, and the average numbers of writes were 9 and 12, for totals of 23 and 25 reads and writes per 1024 samples. Thus the BPS testing did 17% to 28% more reads and writes than the HP-UX operating system running nfsperf and X11perf.

Results

As a result of the combined efforts of the system verification team and the chip development teams, first silicon on all parts had no critical bugs. The system was able to boot the HP-UX operating system on first silicon with no workarounds. This was a new record for HP.

There were also a number of useful by-products from this system simulation work. The architectural simulator was upgraded to PA-RISC 1.1 and extensions were added to model external interruptions and additional I/O devices. The pseudorandom program generator BPS was upgraded to PA-RISC 1.1 and extended to handle asynchronous I/O. The AVPs and IDCs were also upgraded to PA-RISC 1.1 where needed. A significant number of testing utilities were developed which can be used by other similar efforts. One other tool, a pipeline performance simulator called pipes, was developed somewhat independently of the simulation effort and was furnished to the optimizer developers for tuning their design to the Series 700.

PA 7100 Chip

The design teams responsible for the microprocessor chips in the Series 700 were also responsible for the recently announced PA 7100 processor chip.[4] This chip is significantly more complex in that it is a second-degree superscalar machine and incorporates on-chip floating-point circuitry. It also contains numerous other features designed to increase performance.

The same techniques were used for design verification of this new chip, except that BPS was used to test the design *before* the standard test programs were used. The reference machine was also enhanced so that it would automatically identify the location in the execution of the pseudorandom tests where an error was likely to have occurred in the design under test.

The results obtained with the PA 7100 were similar to those obtained with the original chips. The HP-UX and MPE/iX operating systems were booted on first silicon.

Acknowledgments

The authors wish to acknowledge the contributions of the many individuals who participated in this project. This includes chip design and board design by teams in HP's Engineering Systems Laboratory in Fort Collins, Colorado, and in HP's Workstation Laboratory in Cupertino, California, and simulation tool support by HP's Design Technology Center in Santa Clara, California. Also, the authors wish to thank Denny Georg, Russ Sparks, and Steve Foster for providing the management support that made these accomplishments possible.

References

1. M.A. Forsyth, S. Mangelsdorf, E. DeLano, C. Gleason, and D. Steiss, "CMOS PA-RISC Processor for a New Family of Workstations," *COMPCON Spring 1991 Digest of Technical Papers*, February 1991.
2. R. Horning, L. Johnson, L. Thayer, D. Li, V. Meier, C. Dowdell, and D. Roberts, "System Design for a Low-Cost PA-RISC Desktop Workstation," *COMPCON Spring 1991 Digest of Technical Papers*, February 1991.
3. D. Magenheimer, "The HP Precision Simulator," *Hewlett-Packard Journal*, Vol. 37, no. 8, August 1986, pp. 40-43.
4. E. Delano, W. Walker, J. Yetter, and M. Forsyth, "A High-Speed Superscalar PA-RISC Processor," *COMPCON Spring '92 Digest of Technical Papers*, February, 1992.

Mechanical Design of the HP 9000 Models 720 and 730 Workstations

The CPU board, I/O board, graphics board, power supply, mass storage tray, and EISA board assembly are designed as easily accessible modules to support the design goals of low cost, accessibility, serviceability, and manufacturability. The appearance is new, attractive, and compatible with existing HP computer products.

by Arlen L. Roesner and John P. Hoppal

The dramatic increase in performance achieved by the HP 9000 Series 700 workstation computers did not come at the expense of the high emphasis on quality and usability expected of HP products. Good looks, accessibility, upgradability, and serviceability were major objectives for the mechanical design.

Package Design of the Models 720 and 730

From the beginning, the Models 720 and 730 were envisioned as computers that house many different modules, all of which can be accessed and replaced by the customer for a variety of reasons. The modules play a major role in meeting the design goals of low cost, accessibility, serviceability, manufacturability, and cosmetic appearance. Modules chosen for easy access were the CPU board, I/O board, graphics board, power supply, mass storage tray, and EISA board assembly. Designed in Series 300 fashion, all modules slide into the rear of the machine on specific card guides, mating with a backplane at the front of the machine. With the exception of the fans, front panel, and display circuitry, all functional components are found within these easily removable modules. Fig. 1 shows the disassembled Model 720/730 package.

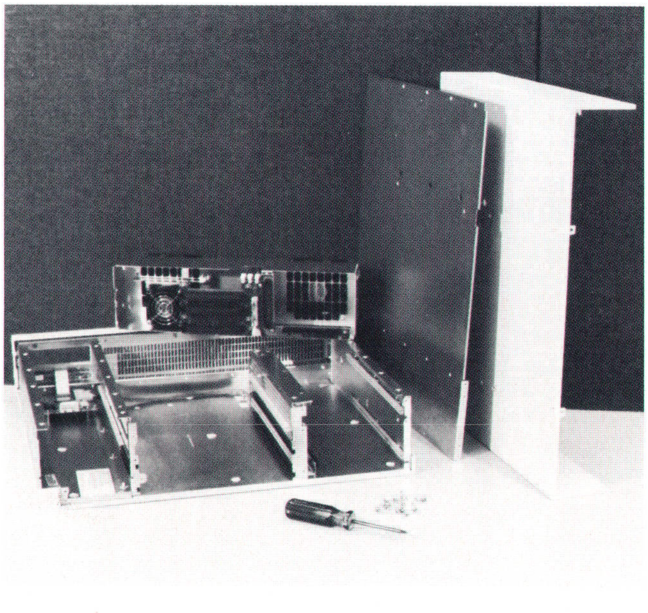


Fig. 1. Disassembled HP 9000 Model 720/730 SPU package.

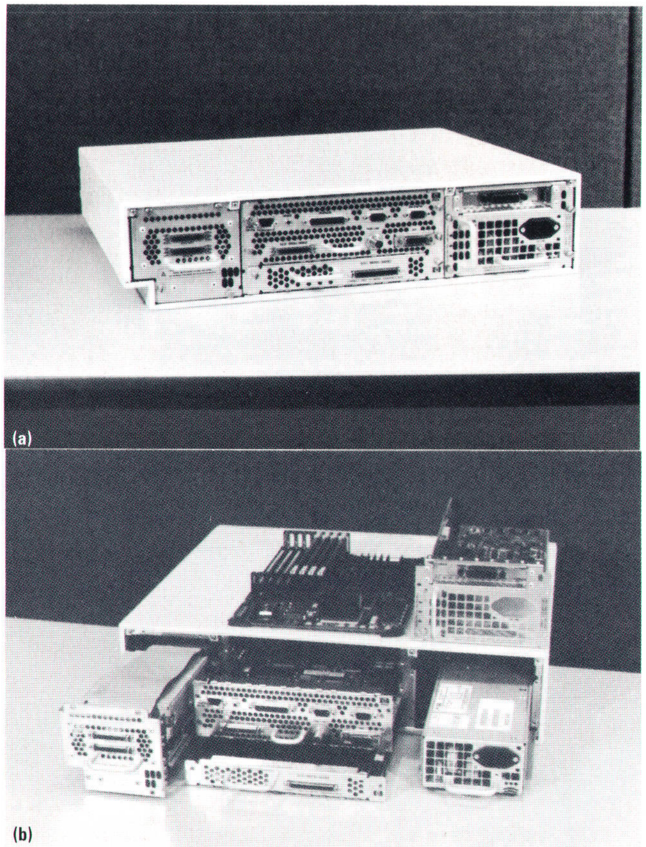


Fig. 2. Access to the disk tray, CPU board, I/O board, graphics board, EISA board assembly, and power supply is from the rear.

Access to modules is simple (Fig. 2). Each module is retained in the system package by screws fastened through holes in an associated bulkhead panel. To access any module, a maximum of five screws must first be removed from the rear, and then the module can be extracted by pulling on a handle. Access is equally easy for machines configured in either of the supported orientations: desktop or deskside. A standard flat-blade screwdriver is all that is needed to remove cables and screws.

Module Features

The Model 720/730 power supply is small given the 200 watts of power it delivers. At 1.6 watts per cubic inch, it was a challenge to fit the supply into its enclosure. For safety reasons, the supply is fully self-contained in its own sheet-metal housing. The edges of the printed circuit board protrude from the sides of the housing and are used to slide the supply into its card guides in the system package. This concept was chosen in lieu of folded sheet-metal flanges on the power supply housing because it provides good registration of the power supply connector to the backplane connector and eliminates the possibility of a nonfit situation caused by tolerance accumulation. Even with this feature, a set of blind-mating connectors is essential for the implementation. To replace the supply, a bulkhead covering the supply must be removed from the rear of the computer. A handle attached to the supply is then pulled to release the supply from the backplane and remove it from the box.

The CPU, I/O, and graphics boards all are similar in size and fit into the middle compartment of the system package. To achieve as much airflow as possible, high-density mating connectors (AMP Z-pack) were chosen for the backplane. These connectors allow for some misregistration and work well as blind-mating connectors, which was a necessity with the anticipated tolerance accumulations found in a card-guide application. Since any one of the three boards can physically slide into any one of the three slots available, the system package is keyed by adding flanges to the side walls of the cardcage, and notches are cut in the front end of each board to correspond with its associated flange. This prevents board insertion in the wrong location, which might result in the board's being destroyed upon power-up.

The EISA board assembly consists of an adapter board (motherboard), the EISA board, and an attached bulkhead plate (Fig. 3). The EISA board is attached to the adapter board in an L-shaped fashion, with the adapter board positioned vertically and the EISA board horizontally. The assembly is inserted into three card guides, one of which is for the exposed long edge of the EISA board. This concept is new to ISA/EISA accessibility. The combination of the EISA board and the motherboard is an easily removable assembly that does not require removal of the system package top cover as in typical personal computer products. Since top cover removal is not required, a customer can easily reconfigure the EISA card without dismantling the monitor and associated cables from the system.

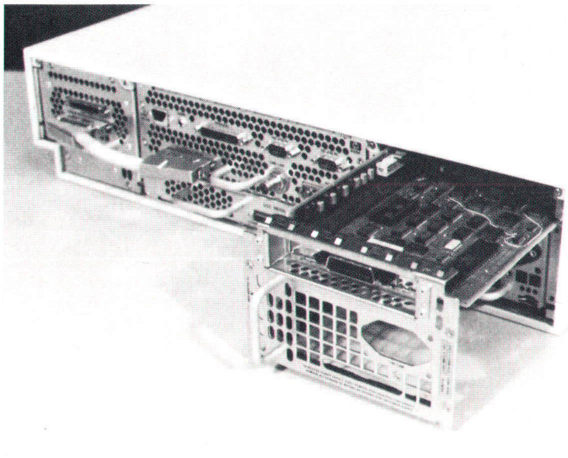


Fig. 3. EISA board assembly.

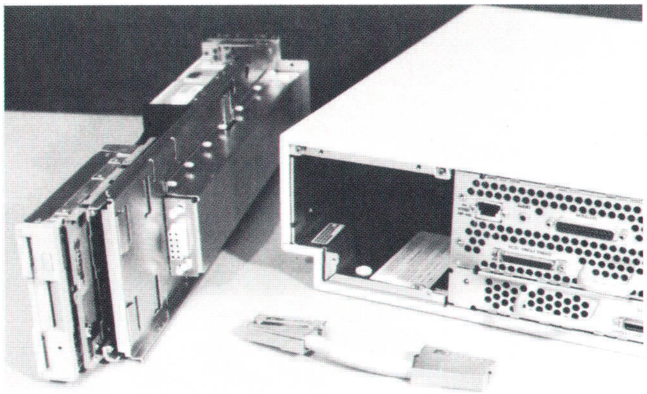


Fig. 4. Disk tray with external SCSI cable.

The disk tray (Fig. 4) is perhaps the most interesting concept in accessibility implemented in the Models 720 and 730. This tray permits the installation of up to two 3½-inch mechanisms (currently two fixed disk drives or one fixed disk drive and one flexible disk drive). The tray includes a floating blind-mating connector, which cables power to each mechanism. This mates with a separate backplane connector inside the system package. SCSI signal cabling is provided by a 50-position ribbon cable, which is terminated in two places on the tray bulkhead. This establishes an internal backplane connection for power and an external cable connection for the SCSI. With the simple removal of the external SCSI cables and four bulkhead screws, the tray pulls completely from the system package without any other disassembly. This becomes a valuable tool for software loading, disk troubleshooting, and mass storage security. For software loading, another computer can be attached to the signal ports of the system package by quick manipulation of some external cables. Software can then be transferred from a master disk to the local computer, after which the system package can be reconfigured with its own external cables and act independently. Mass storage can be easily secured by removing a disk tray and locking it inside a desk drawer.

Design Considerations

Along with the design goals of a fast-track schedule, high quality, low cost, small package size, and good manufacturability, the Models 720 and 730 were required to comply with regulatory (EMI/ESD), environmental (shock/vibration), and safety standards. As a product design is made more customer-accessible and therefore more complex, it is normally true that all of these requirements become more difficult to meet. For the midrange Series 700, cost was not as much of an issue as it would be for a low-end machine, but cost-cutting was still achieved by hard-tooling all of the many sheet-metal components. Quality was increased by postplating the sheet-metal parts and encasing the package with plastic. Each module was aggressively designed to fit into as small a space as possible, and efforts were made to make even the fans and display boards easy to install on the manufacturing floor.

Perhaps the most difficult problem facing the Model 720/730 package design, with its many removable rear bulkhead panels, was in the area of regulatory compliance, particularly ESD. With clock speeds in the 50-to-66-MHz region, CPU circuitry becomes very sensitive to electrostatic discharge. Points of entry for static discharge are any slots made by mating panels of the system package. With many bulkheads and thus many mating areas, the Models 720 and 730 presented a big challenge. Consequently, most I/O connectors are outfitted with their own ESD gaskets, and spring clips or dimples are added to the bulkhead surfaces to seal all holes or gaps. Around connectors, gaps are no more than 0.50 inch wide. For overlapping metal surfaces, dimples are placed to provide no more than 0.50 inch of slot width. Bulkhead spring clip spacing is less than 0.75 inch. With the more stringent regulatory requirements expected in 1992, and increasingly higher clock speeds, it is expected that these design parameters will become smaller in the future.

EMI becomes a problem when currents are allowed to travel inside the machine and emanate as electromagnetic energy. A feature of the Model 720/730 implementation that was aimed at controlling EMI is the design of the disk tray compartment. Because the Model 720/730 package also supports a flexible disk drive, a large frontal hole was required in the sheet metal. This large hole presented a potential area for EMI radiation. To reduce this possibility, the disk tray compartment is completely isolated from the other compartments by placing a sheet-metal wall between compartments, leaving only a small hole for the interfacing power cable. Disk signal lines are routed externally on the shielded SCSI cable. These two measures effectively control EMI emissions from the disk compartment.

In the area of environmental compliance, the largest area of concern was the placement of disk drives in a tray that slides into somewhat loose-fitting card guides. Disk drives are historically very sensitive to shock and vibration environments, and loose fits tend to amplify these levels. To alleviate this amplification, two measures are implemented in the Models 720 and 730. The first is a panel-mounted bulkhead on the tray. This provides some hard-mount features by stabilizing the rear portion of the tray. The second feature is the use of shock-absorbing isolation grommets between the drive and its associated mounting bracket. These grommets damp out most vibrations at frequencies that correspond with the drive resonant frequency. These grommets also allow greater displacement of the drives in shock testing, so adequate space for movement is designed into the drive brackets and disk tray.

Safety compliance presented no serious challenges to the Model 720/730 design, except that all of the removable components of the system are designed without sharp corners or edges that pose a danger to an operator. Fans are outfitted with finger guards where customers have access to them. The power supply's sheet-metal enclosure prevents any chance of electrical shock to the user.

Industrial Design

The industrial design for the Series 700 products took place in parallel with the industrial design efforts for the Series 400 workstations. Two industrial design firms—ZibaDesign and Roche-Harkins—were hired to help with the industrial design. The HP corporate industrial design group was also heavily involved. Most of the concepts and ideas came from a joint effort of the Series 400 R&D design teams in Chelmsford, Massachusetts and Fort Collins, Colorado and the Series 700 R&D design team in Fort Collins.

The overall industrial design objectives for the Series 700 workstations were as follows:

- To create a new look for a new family of workstation products that would visually convey innovation
- To create a new look that would be compatible with both new and existing HP computer products
- To explore alternative volume configurations and shapes to maximize performance and value
- To create a design framework to guide the development of the workstations
- To fit extremely well with an existing graphics product that was expected to be included with a large portion of the shipments.

Model 720/730 Industrial Design Process

The overall task was divided into five phases. During each phase, reviews were held with the HP corporate industrial design group and with the other groups developing new workstations. This was done to ensure that the individual implementations worked together and with the existing computer products.

Phase 1: Two-Dimensional Concept Development. This phase started by determining the product requirements from a component standpoint. All of the the internal components and their rough sizes were decided upon. Interconnect, thermal, EMI, and customer access requirements were also evaluated. Using these requirements, hand-drawn sketches were done showing a number of possible layout and shape concepts. At the beginning of this phase the following product requirements were determined:

- Customer access to the major individual components. This was very important. A design that was easily serviced, manufactured, and upgraded was also desired.
- Support for both desktop and deskside configurations using the same basic package. This implied both vertical and horizontal orientations of the product in the end-use configuration.
- Support for two 3½-inch disk drives. One of the drives could be a flexible drive and thus would require frequent access by the end user from the outside of the package.
- Support of a single ¼-panel graphics board.
- Support of a single ¼-panel processor board.
- Support of a single ¼-panel I/O board.
- Support of eight small memory boards.
- Support of a 200-watt power supply.
- Support of a single EISA slot along with an interface board.

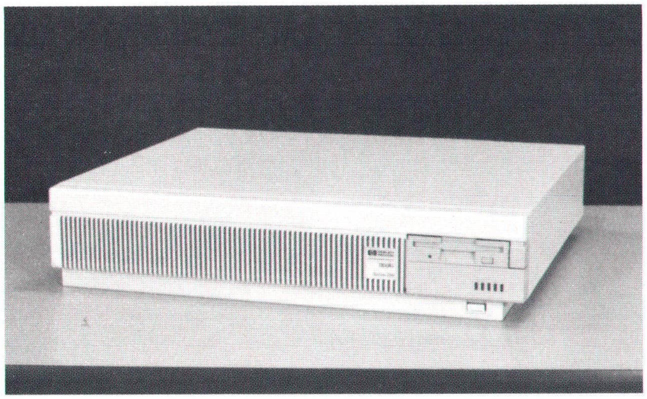


Fig. 5. HP 9000 Model 720/730 SPU package in the horizontal orientation.

- Forced convection cooling. It was decided that the airflow direction would be front to back with the cool air entering the front of the package and the hot air exiting the back of the package. The airflow direction was chosen to keep the heated air from exiting directly out to the end user. This was considered especially important in the desktop configuration.

Phase 2: Two- and Three-Dimensional Concept Development. During the second phase, foam core mockups of the various components were created and used to explore the alternatives from Phase 1 in more depth. Having the foam core components also gave better visual feedback on overall shapes and sizes.

Phase 3: Concept Refinement. As we exited phase 2, we had decided upon two or three of the best alternatives. The deliverables from the third phase were urethane foam models and their corresponding layouts. Some preliminary detailing was also looked at during this phase.

Phase 4: Concept Refinement. In this phase the final direction was chosen. All of the first-pass detailing for the chosen alternative was completed. The deliverable from this phase was a single urethane foam model showing all the proposed detailing. The corresponding layout was also included.

Phase 5: Concept Finalization. The final phase consisted of reviewing the model and making minor detailing changes based on inputs from the groups involved. Final cosmetic control documents, color requirements, and one full-size hard cosmetic model were outcomes of the final phase.

Family Concept

One of the major requirements for the new look was to create a family appearance for all the new workstation products while still leaving enough freedom for each product to deal with its own set of specific requirements. This was the most difficult requirement to meet. By means of frequent meetings and discussions among all the groups involved, a set of guidelines was determined that would be used to accomplish this goal. The major unifying pieces decided upon were:

- The use of parchment white as the primary color. This also helped to tie in existing HP products.
- The use of intersecting rectangular shapes.
- The use of color change (dove gray) to specify areas for customer interaction.
- The use of the radii specified for the existing HP computer products.
- The use of a small recess to separate the front panel visually from the rest of the package. This feature is common to all products that follow the industrial design standard of the existing HP computer products.
- The use of standard HP colors for the LEDs, keycaps, and switch buttons.
- The use of the existing products' grill pattern to tie in these products.
- The use of rectangular nameplates that can be rotated so that they work in both the horizontal and vertical orientations.
- The use of specific LEDs to show the end user overall activity. The five specified are power on, heartbeat, incoming LAN activity, outgoing LAN activity, and disk activity. The icons used for describing the LEDs are also specified.
- Front access to the power switch.

Model 720/730 Package

The Model 720/730 package is a sheet-metal enclosure to which cosmetic plastic covers are added. The cosmetic pieces form overhangs on the top front and top right of the package. These overhangs give the impression of two distinct intersecting rectangular boxes.

The bottom, top, and sides of the enclosure are covered by two large parchment-white parts. These parts also form the overhang on the top right edge of the product. When the unit is used in the horizontal configuration, four small parchment-white rubber feet are added to recesses in the bottom cover. When the product is positioned vertically, the left side of the package then becomes the bottom. There are small hole plugs in the cover that are removed in this case, allowing access to two press-in nuts. Screws are then used to attach the pedestal foot to the main unit. The pedestal helps support the product and keep it from tipping over. The pedestal is an I-shaped foam molded part which visually appears as four separate rectangular feet when in the installed position.

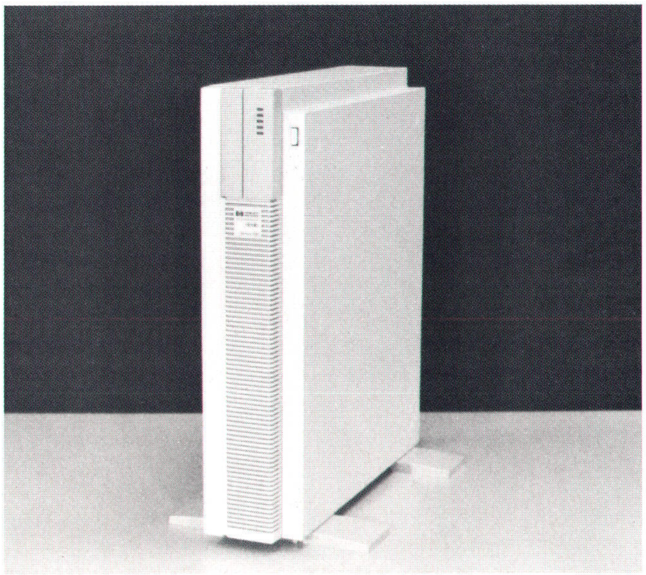


Fig. 6. HP 9000 Model 720/730 SPU package in the vertical orientation.

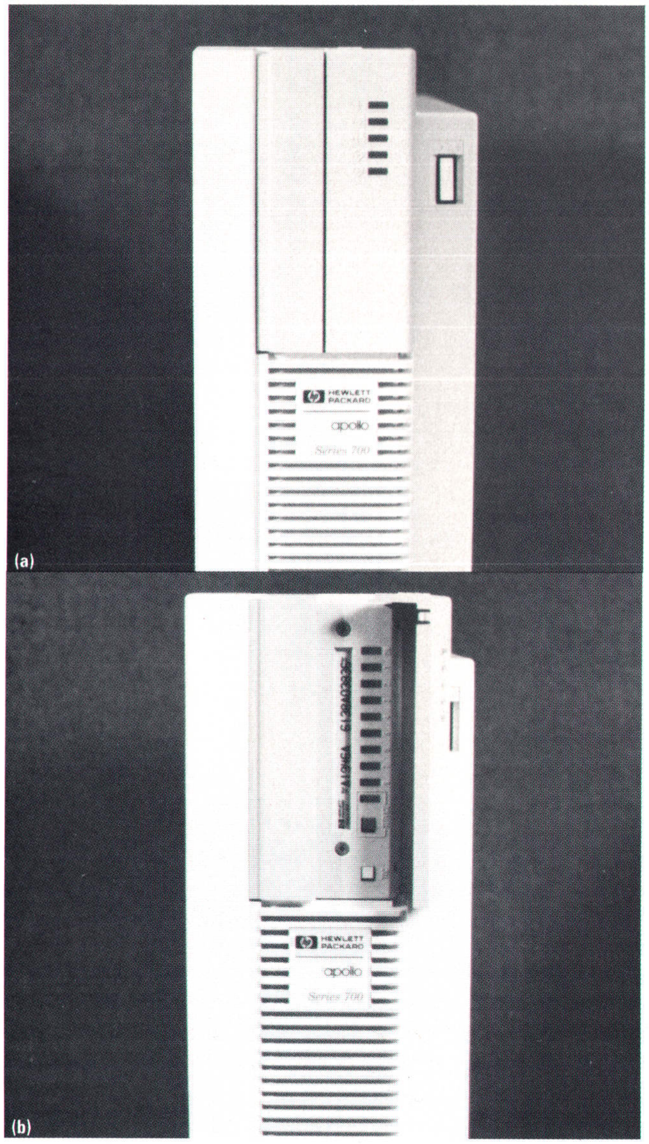


Fig. 7. Control panel area. (a) With the door closed, the five primary LEDs can be seen from the outside. (b) With the door open, the remaining LEDs and switches are accessible.

The cosmetic front of the enclosure is formed by either three or four separate parts depending on whether the optional flexible disk is present. The front face of the package was the major industrial design challenge since it had to work well with the horizontal grill pattern used on current HP products, whether the product was used horizontally or vertically. This could have been accomplished easily by the use of two distinct front panels, but it was felt that it would be much more desirable and easily managed if one universal front panel could be used. It was determined that the majority of the users would use the product in the vertical orientation. Therefore, it was decided that the grill pattern should run horizontally when the product was used in the vertical orientation. The issue then was to make the product also work with vertical grills when in the horizontal orientation. This was accomplished by adding detailing perpendicular to the grillwork. Four major details give the feel of horizontal lines even though the grillwork is vertical. The first detail is a 5-mm-wide recess along the entire front of the product and around the corners of the front panel. The second detail is the overhang. This is a very strong horizontal feature that tends to lessen the visual effect of the vertical grillwork. The third detail is a horizontal line formed at the intersection of the control panel bezel and the control panel door. The final detail is the horizontal slot for the flexible drive. The unit in a horizontal configuration is shown in Fig. 5. In Fig. 6, the unit is shown mounted vertically with the pedestal foot.

The control panel area of the front panel is located at the right edge of the front panel in the horizontal orientation or the top of the front panel in the vertical orientation. It is formed by a combination bezel and door. When the door is closed, the five primary LEDs can be seen from the outside. By opening the door, the user has access to the remaining troubleshooting LEDs and the service/normal and soft reset switches. The keycaps for the two switches are molded, using custom HP colors. The 5-mm-wide recess is molded into the bezel, making that feature flow completely across the front face. A closeup of the control panel area can be seen in Fig. 7. In Fig. 7b, the door is open.

If the unit has a flexible disk drive, then a different bezel/door combination is used. This assembly has a rectangular hole into which the disk drive bezel slides. The bezel was retooled to add the 5-mm recess feature (see Fig. 8). All of the control-panel bezels, the door, and the flexible drive bezel, door, and button are molded in the HP specified dove gray color.

The graphics package was also an integral part of the design. Its package had already been completed, and initially, it was unclear how to make it work with the Models 720 and 730. It was determined that two different bezels should be

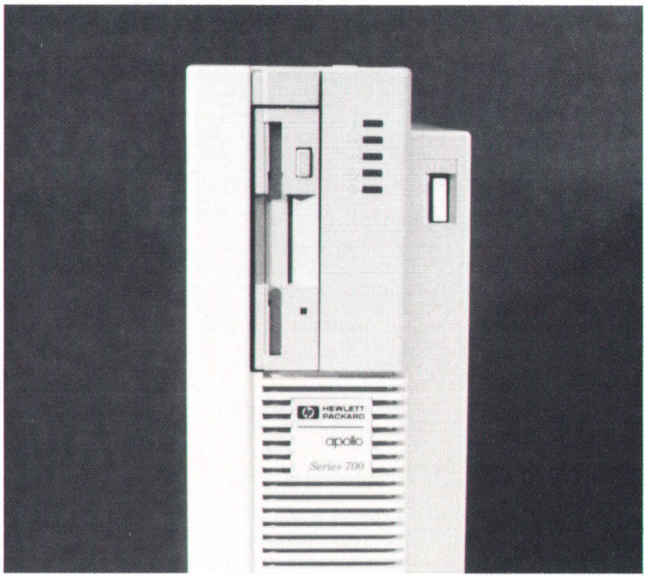


Fig. 8. Bezel and door design for the flexible disk drive.

done for the graphics package. One bezel has vertical vents and is used with the Models 720 and 730 in the vertical orientation. The second bezel has horizontal vents and is used in the horizontal configuration. In addition, the graphics lab designed a mounting scheme that allows the graphics package to be hard-mounted to the SPU easily. A Model 720 workstation with the optional graphics package is shown in Fig. 9.

Acknowledgments

It is very important to acknowledge the work of many others who played significant and in many cases leading roles in the new-look industrial design. They include Don Lahey of HP corporate industrial design who kept a close eye on the group to ensure that the new look also fit in well with the overall HP product family, Paul Febvre who worked on the Series 400 and Series 700 industrial design, Dan Busby who worked on the Series 400 industrial design, and Rex Seader who worked on the graphics package.

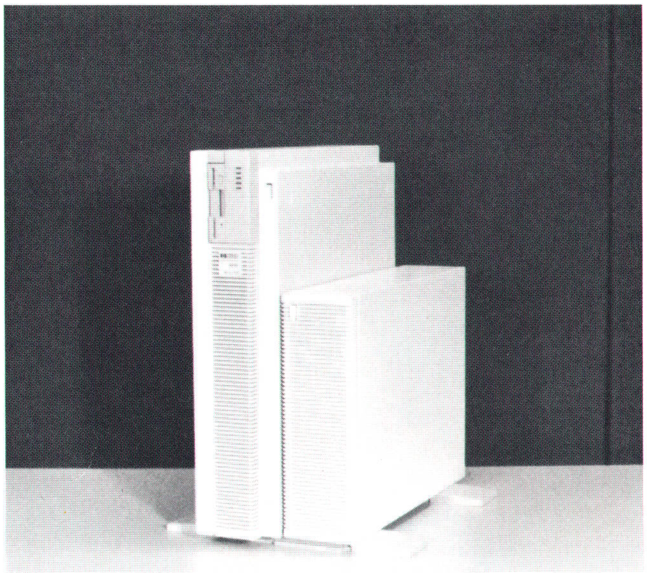


Fig. 9. SPU with optional graphics package.

Meeting Manufacturing Challenges for PA-RISC Workstations

To meet the time-to-market goals for the HP 9000 Series 700 workstations, major contributions were made in design for manufacturability and in expediting standard processes. One manufacturing operation installed a new surface mount production facility and developed a new printed circuit production process simultaneously.

by Spencer M. Ure, Kevin W. Allen, Anna M. Hargis, Samuel K. Hammel, and Paul Roeber

In the HP 9000 Series 700 workstation development project, manufacturing was challenged to participate with the design and marketing teams to meet the time-to-market goals for the Series 700 while significantly improving the manufacturability of PA-RISC workstations. The Series 700 workstations were the first of a new generation of PA-RISC workstations for which high-volume delivery requirements were imposed on the worldwide manufacturing operations within the HP Computer Manufacturing Division.

Manufacturing management was represented on the Series 700 program team by a manufacturing development engineering manager to provide direction to the worldwide manufacturing team. Overall tactical program management was led by a manufacturing new products engineering manager. Project managers were identified at each of the computer manufacturing operations around the world. Each manufacturing operation created a manufacturing tactical team responsible for local issues (see Fig. 1).

Strategic Contributions

Manufacturing was challenged to make major contributions in two key areas: design for manufacturability and expediting standard processes.

Design for Manufacturability. In this area the challenge was to deliver a new printed circuit board assembly process that would allow the hardware design team to take full advantage of the performance capability of the VLSI designs for the Series 700. The goal was divided into two parts. One was to release a printed circuit assembly process that provided surface mount capability along with through-hole parts (pin grid arrays and connectors) on both sides of a 12-layer

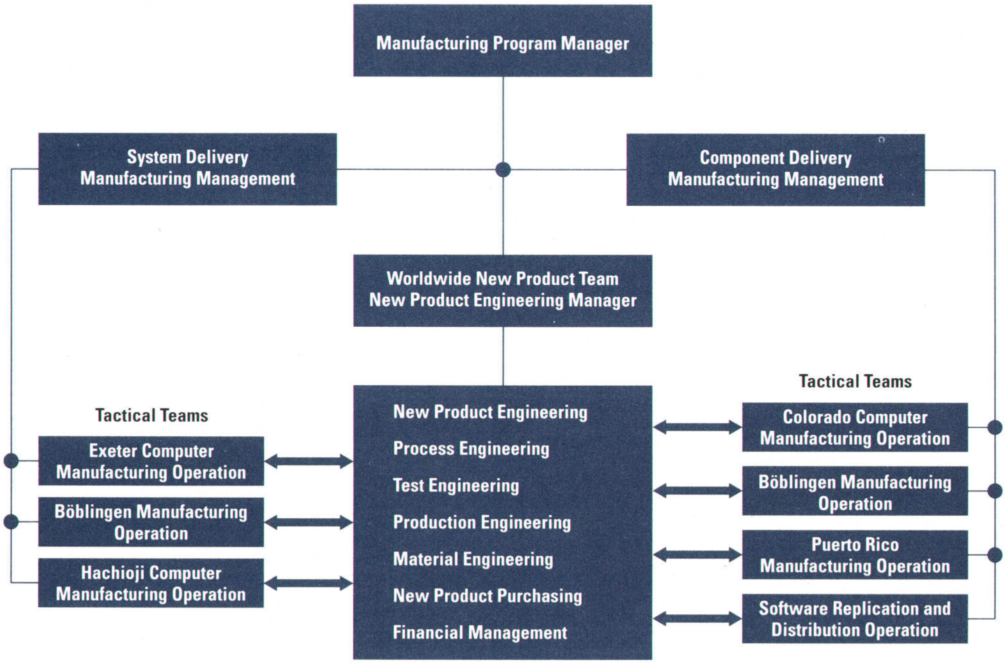

Manufacturing Program Manager
System Delivery Manufacturing Management
Component Delivery Manufacturing Management
Worldwide New Product Team New Product Engineering Manager
Tactical Teams
Exeter Computer Manufacturing Operation
Böblingen Manufacturing Operation
Hachioji Computer Manufacturing Operation
New Product Engineering
Process Engineering
Test Engineering
Production Engineering
Material Engineering
New Product Purchasing
Financial Management
Tactical Teams
Colorado Computer Manufacturing Operation
Böblingen Manufacturing Operation
Puerto Rico Manufacturing Operation
Software Replication and Distribution Operation

Fig. 1. HP 9000 Series 700 workstation manufacturing program organization.

* Parts of this article are reprinted with permission from *Circuits Assembly* magazine, September 1991.[1]

printed circuit board. In parallel with this development, one manufacturing operation was required to install and release a new surface mount production facility, and at the same time support the Series 700 project and meet the Series 700 time-to-market goals.

Other goals were to deliver a new, highly responsive software release process within manufacturing, and to deliver a new test process to reduce the cost of testing and the time required to diagnose and isolate failing components and assemblies. The goal was to reduce test and diagnostic time to less than 50% of the time required for the previous generation of PA-RISC workstation products.

Expedite Standard Processes. The hardware release team used an HP standard process called the Hardware System Product Life Cycle (HSPLC) as the guideline for release of the Series 700 from engineering into manufacturing. To achieve the time-to-market goals, it was critical that time lines be managed within the HSPLC to minimize unnecessary delays.

The key delivery requirement identified as most critical was to provide rapid turnaround time for new prototype hardware designs and delivery of preproduction systems. The goal was to eliminate any noncritical issues that might delay delivery, and to allow the manufacturing team to focus on a few critical issues and minimize their impact on the schedule.

R&D and manufacturing used two-person bring-up teams to facilitate the fastest possible delivery of prototype printed circuit assemblies and systems. Each member of the team was responsible for supporting the other member by looking for ways to eliminate problems before they happened.

Printed Circuit Assembly Process

Developing a high-mix, medium-volume surface mount manufacturing process is a challenge. On the technical side, equipment selection and installation, line flows, interconnect technologies, product mix, and setup must be addressed. On the operational side, engineering, technician, and operator staffing and training must be considered. Adding the variables of a new product and a new technology provides a demanding and risky challenge. This was the situation facing Hewlett-Packard's Colorado Computer Manufacturing Operation (CCMO).

The Series 700 family of workstations was already under development in R&D. The printed circuit assemblies were designed and laid out to meet an aggressive schedule, and CCMO was given until October 1990 to prepare to assemble prototype boards.

A cross-functional team was established, since loading a surface mount board involves more than just the physical assembly. This team consisted of representatives from process development engineering, finance, computer systems, materials engineering, purchasing, production control, test, quality, and production. The team's first assignment was to establish goals and objectives for the new production line.

A vision of the surface mount manufacturing process was developed, and goals and objectives were defined. The goals were to minimize space, inventory, direct labor, and overhead while maximizing flexibility. Specific, measurable objectives were established to compare with the current through-hole process metrics in the areas of responsiveness, quality, and cost.

After discussion of process options, differences between surface mount and through-hole technologies, and the changes required in each of the organizations to accomplish these goals, the needs of each area were defined. Other HP entities with surface mount manufacturing capability were visited to discuss their successes and areas of concern. A strategy was defined for this new surface mount manufacturing process by leveraging the experience of other sites, working with the CCMO surface mount team, and modeling the production process. This strategy allowed us to achieve our goals in a rapidly changing technical environment while attaining competitive assembly costs.

Technology

Over the last six years, Hewlett-Packard's product generation process organization has developed, characterized, and documented a set of core processes. These consist of one single-sided and two double-sided processes, which HP has designated SMT-1, SMT-2, and SMT-4.

Because of performance requirements for the Series 700 CPU board, the printed circuit board design was not compatible with any of the released processes. The SMT-2 process guidelines were violated by the presence of fine-pitch components, over 2100 through-hole leads, and a 0.079-inch-pitch through-hole connector. The SMT-4 process guidelines did not accommodate the design requirement for SOJ-26 SRAMs, tantalum capacitors, and SOT-23 components on the bottom side of the board.

However, a new process, SMT-2thp, was under development by the product generation process organization. The objective of SMT-2thp was to expand the current SMT-2 process capability to include thicker boards, more through-hole components, fine-pitch surface mount components, and finer-pitch through-hole components. SMT-2thp was considered a high-risk development because of the mutually exclusive objectives of trying to push solder paste into through-holes while applying a minimal amount of paste to the fine-pitch pads. CCMO joined the development effort to ensure that SMT-2thp would be an acceptable manufacturing process for the Series 700.

Production Equipment and Strategy

As the technology requirements were being defined, the equipment procurement process was initiated. Product characterization was completed through an understanding of the panel sizes, surface mount versus through-hole component mix, surface mount component sizes, and lead pitch requirements. The standard HP surface mount equipment capabilities were reviewed to determine whether they would meet the project goals and objectives for the given product mix. The remaining process equipment was selected and purchased to enhance the standard equipment's ability to meet the goals and objectives.

The manufacturing process was then modeled to determine the optimum production strategy. Many scenarios were analyzed to determine the competitive solution. A sensitivity analysis was conducted to determine the effects of technology, product mix, and volume changes. Based on this analysis, continuous flow manufacturing philosophies with a family-based product line were adopted to achieve the team's flexibility and responsiveness goals.

High-Quality Printed Circuit Assembly

The assembly of high-quality products begins in the product design phase. Hewlett-Packard has developed and maintains standard design guidelines for printed circuit boards assembled at its manufacturing entities. These design-for-manufacturability guidelines assist designers in making design/manufacturing trade-offs. They also reduce the risks of design related quality defects and simplify the transfer of products between HP sites. These guidelines serve as a vehicle for relaying information between manufacturing and the design labs.

To prepare for the surface mount implementation, the supervisors and operators received surface mount training. Twelve operators were loaned to the HP Colorado Surface Mount Center (CSMC) for hands-on surface mount equipment and process training. In combination with this core set of trained operators, additional operators were used at the facility whenever they were required. This minimized costly learning curve inefficiencies.

Supervisor training was initiated by inviting a supervisor on the through-hole line to become involved early in the development of the team's process vision, goals, and objectives. Process engineering met biweekly with production operators to solicit inputs and discuss the surface mount process. Both the team and the long-term manufacturing process benefited from the expertise and knowledge gained through this early training and teamwork.

Accurate, up-to-date documentation and training are keys to quality manufacturing. The process development engineers worked with production and technicians to provide documentation to describe the manual assembly processes, machine operations, and the flow strategy within the shop. Procedures were written for each step in the assembly and test process.

An operator/technician surface mount technical training and certification program was then established to qualify operators for each job. Initial operator training was completed by process development engineers. This training was based on the machine, assembly, test, ESD, safety, and flow strategy documentation. It included a review of procedures, hands-on machine training, and flow training. A workmanship training program was added to help the operators recognize, report, and repair quality problems.

Statistical process control charts were kept at each of the machines to monitor and maintain process control of the key variables. The advantage of using these charts is illustrated by the case of paste viscosity testing. Any batch of paste that measured beyond the specification limits was discarded. Operators used only batches that measured within the control limits, thereby preventing potential solder defects.

Operators inspected their own work throughout the process and checked for solder defects and other obvious problems. In addition to the statistical process control efforts, if an operator discovered a problem generated in another area, these findings were immediately communicated to the source. The proximity of the assembly and test areas fostered this quality feedback loop. This immediate feedback along with the minimal levels of work-in-process reduced the risk of producing large quantities of defective printed circuit boards. This resulted in a reduced defect rate, simplified repair loops, increased output, and decreased throughput times.

Results

As a result of the planning, development, and teamwork efforts, the responsiveness, quality and, cost goals were achieved. Process development was completed in time for the product ramp. The space requirements were well within the goals, but also had the flexibility to change.

Compared to the through-hole process metrics, the throughput time on the surface mount line was reduced by 83% and the parts-per-million defect level was reduced by 40%. The CCMO team's process goals were achieved. The CCMO surface-mount manufacturing process has the ability to manufacture high-mix, medium-volume lots of printed circuit boards.

The quality level achieved far exceeded our expectations and the defect levels for both solder and component defects were well below the goals. This was achieved largely through joint design efforts of manufacturing and R&D.

The manufacture of the Series 700 boards began on schedule. The SMT-2thp process for assembling the Series 700 CPU board has been duplicated at the printed circuit assembly facility at the Böblingen Manufacturing Operation (BMO) in Germany.

Software Release Process

Hewlett-Packard's software and hardware manufacturing divisions worked together to improve the software release process for the Series 700 workstations. Software Reproduction and Distribution Operation (SRDO) and Exeter Computer Manufacturing Operation (ECMO) established a software delivery team to address software release problems. The goals were to reduce the turnaround time from the labs to shipment for HP-UX* software products and to improve communications between the functional organizations. The problem was broken down into two release processes, one for the HP-UX operating system and one for compilers and application software products. Each type of release required a different release model.

In the past, the software development laboratories released software to manufacturing randomly. Each software product was released as soon as the quality goals set for it were reached. Manufacturing attempted to prepare these products for shipment on a first come, first served basis, but critical needs sometimes caused sudden changes in priorities, which made it difficult to achieve manufacturing efficiencies.

The project team reviewed numerous options and changed the release model from random, unstructured releases to structured releases based on product needs. The HP-UX operating system release model was allowed to continue very much as before, that is, it is based on achieving quality goals and releasing as soon as goals are attained. The compiler and application software product release process was changed from a first come, first served basis to a batch and critical-need release process based on a fixed interval of time. Products are accepted from the labs only in batches at two-month intervals. Manufacturing can then put each batch into full production before receiving the next batch.

Structuring these releases into different models and communicating the new release process to the labs allows a standard release template for each product timeline. As long as the labs meet the timeline, the complete release can be shipped at once. In this way, SRDO is able to achieve operating efficiencies and therefore reduce the time to market for the software products. This new application and compiler software product release model is called DART (Delivery Application Release Train).

The project team performed a design-for-manufacturability analysis on these products and recommended different packaging options to reduce the product cost. The standard set of HP-UX manuals shipped with every system was reduced to a set of overview manuals and numerous miscellaneous items were eliminated. The team also created a dedicated manufacturing team at SRDO to manufacture the software packages, established communication vehicles between divisions, simplified the product structures, and delivered Series 700 systems with HP-UX operating system software and key application software products.

The changes to the process have reduced turnaround time by 50%, reduced the product cost by 80%, and led to a successful and timely release of the operating system and compiler and application software products on the Series 700 workstations.

Lowering Test Costs

The test process for the Series 700 is highly automated at both the board and system level. In particular, the controller system for the the Series 700 board level functional test provides hands-off, high-throughput testing as well as detailed data collection and tracking. The board test system also automated the socket board test for the three major ASICs (application-specific integrated circuits) during the production ramp. It is a relatively inexpensive means of providing quick, accurate feedback on line returns and supplying production with reliable components.

One of the primary objectives in the design of the board test system was to support automated functional testing and reliable data collection with throughput quantities of four times the forecast at no additional cost. The ability of the board test system to support four times the forecast has allowed CCMO to ship virtually without delay when material problems demanded meeting commitments with sporadic, very high-volume builds. To improve throughput, the board test system also attempts automatic failure isolation to the failing component when possible and includes the suggested repair. The board test system features such as voltage margining, modular test applications, and automatic isolation, along with testability design hooks in the the Series 700 hardware and firmware, have allowed CCMO to meet cost and quality objectives for the Series 700.

Another important objective of the board test system was ease of use (no training required) by conforming to emerging user interface standards. Because the schedule for the new user interface standard did not coincide with the Series 700 schedule, it could not be used. The board test system user interface, however, is quite user-friendly and meets the ease-of-use objective. The new interface standard can easily be employed when completed.

Data Access for Repair

The data collection and reporting capabilities of the board test system facilitate rapid improvement of both component reliability and the process itself. The board test system tracks every board tested. It records every test run and the corresponding output for boards that fail. The data for all boards shipped is archived monthly, allowing easy access to the history of any particular part number. In addition, the board test system manages a repair queue and automatically assimilates the data from the functional test and repair process into useful, readable summaries.

The board test system is also the server for the technician repair system. Here technicians troubleshoot board failures that the board test system couldn't automatically isolate. The technician repair system provides access to all of the board test system's data so that technicians can quickly see the history of every board needing repair. Repair entry features and several other aids are available to facilitate troubleshooting and repair of the Series 700 boards. Both the board test system and the technician repair system make batch reports nightly.

System level testing is automated by the system test controller. The configuration and status of each system are tracked all the way through the process. This controller is linked to the data tracking system and quickly identifies the unit needing replacement when a failure occurs.

The high level of test automation at both board and system levels is a key factor in the manufacturing of the Series 700 workstations. It provides the ability to ship high-quality workstations at quantities well in excess of forecasts when material problems make it necessary. In addition, the easy access to detailed data facilitates the repair of precious printed circuit assemblies.

Quality

Quality was a primary concern throughout the development and prototyping of the first Series 700 products. Product quality was never compromised. The Series 700 team set forth a rigorous collection of qualification and production processes to ensure that Series 700 products would be of the highest delivered quality. The key items that were put in place include:

- A rigorous product qualification test
- A rigorous board and system production test process
- An oven aging process for early prototype and early production units
- A commitment to developing a highly reliable production process in preparation for volume shipments.

The quality organization and the R&D lab undertook the product qualification process, while the new product engineering teams in manufacturing undertook the task of developing the required production processes. After designs are verified, strife (stressed life) and environmental tests are used to identify latent defects in product hardware. Systems are tested under varying conditions: temperature, humidity, shock, vibration, and altitude. These factors accelerate normal product stress and determine whether or not the system meets design specifications. When the qualification has been completed, one can be fairly certain that the product will operate properly in all required customer environments, and will withstand normal use and abuse.

Test Processes

At board test, a series of offline diagnostics and online exercisers were run during early production to ensure that component or process defects were detected as quickly as possible. As the production process improved and component problems were isolated and corrected, tests were removed to reduce the cost and time required to produce the product (see Fig. 2).

System test is broken into four sections: integration, oven age, run-in, and post. At integration, assembled systems are connected to keyboards and monitors and offline diagnostics are run. Integration testing is done to verify that the system is assembled properly and that there are no major problems with system components. Next, systems move to oven age, where they undergo temperature cycling while running offline diagnostics. Oven age testing is done to stress system components and determine if there are any temperature sensitive components or process defects. Systems then move to run-in, where they are power cycled and offline diagnostics and online exercisers are run. The length of time spent in run-in depends on product quality. As the quality of the products and the process improves, run-in test time can be reduced. Finally, systems move to post test, where monitor and keyboard functionality are checked one last time before boxing the system. At post test, an operator boots the operating system and runs a quick graphics test program.

The oven age process was designed to run on early prototype systems and to be eliminated before entering full production. For all Series 700 products this has been accomplished. The early prototypes were run through a series of one-hour temperature soaks at 0°C and 55°C. With each new introduction, the oven age process identified process, component, and design flaws early in the process, and before systems were shipped. This early detection of faults was instrumental in holding to schedule. Soon after flaws were corrected, the oven age process was no longer needed and was removed. The process is still used to qualify new components or design changes, but has not been necessary to maintain product quality.

With volume readiness being the focus, processes need to be able to support the required flow of product, and product and process quality levels need to be at or near established goals to maintain the production process cost effectively. Series 700 process readiness and quality were in place at introduction to support a volume ramp. The best testimony to volume readiness came in October 1991, when nearly 5000 Model 720 and 730 workstations were delivered in less than four weeks.

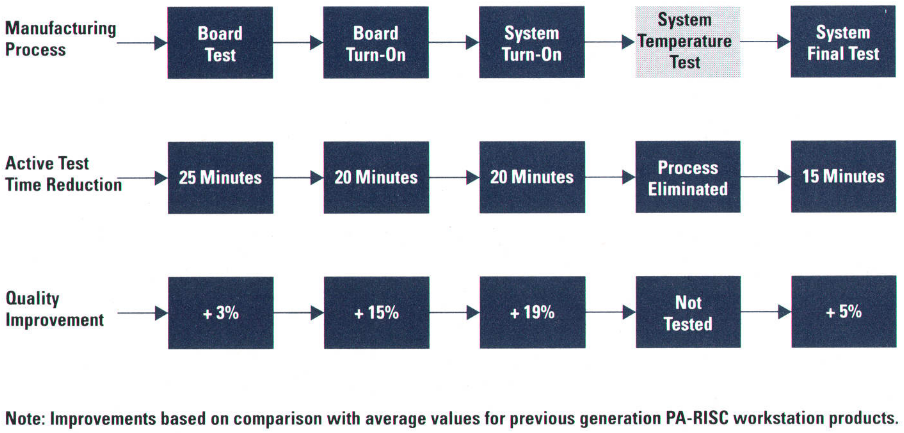


Fig. 2. Series 700 test process flow and improvements.

Acknowledgments

The key to the successful introduction of the Series 700 product family into production was not all the processes mentioned above, but a commitment from a large number of people in manufacturing and the R&D labs to get the job done. Like any new product introduction, the quality of the product and process is dependent upon the people responsible for closely monitoring progress and implementing the required changes. The Series 700 manufacturing introduction was a success because of that commitment from management, engineering, procurement, planning, production, finance, distribution, and quality.

The authors would like to acknowledge the contributions of Jim Getches, new product engineering manager, Les Hoogeveen, manufacturing development engineer, Bob Wrenn, business team manager, and Dennis Floyd, manufacturing development engineer.

References

1. K.W. Allen and A.M. Hargis, "Surface Mount Assembly: Meeting the Challenges," *Circuits Assembly*, Vol. 2, no. 9, September 1991, pp. 30-39.

HP-UX is based on and is compatible with UNIX System Laboratories' UNIX* operating system. It also complies with X/Open's* XPG3, POSIX 1003.1 and SVID2 interface specifications.

High-Performance Designs for the Low-Cost PA-RISC Desktop

This paper presents the processor, memory, graphics, multimedia, and built-in core I/O design of the new HP 9000 Models 705 and 710 entry-level, scalable, PA-RISC workstations. The use of a buffered CPU/memory interconnect is important for scaling the high-frequency, high-performance processor design to the entry-level desktop.

by Craig R. Frink, Robert J. Hammond, John A. Dykstal, and Don C. Soltis, Jr.

The HP 9000 Models 705 and 710 (Fig. 1) are new, entry-level members of the Series 700 performance workstation family. They complement the higher-performance Models 720, 730, and 750 workstations (see article, page 6), offering lower cost and broad functionality. The Series 700 workstation products are based on Hewlett-Packard's PA-RISC architecture,[1] providing application-code compatibility across a broad range of applications. The systems address applications ranging from those characterized by cost sensitivity, diskless workstations, and grayscale graphics to those requiring higher-performance 3D color and maximum computational ability.

The 35-MHz Model 705 is designed for desktop applications that can benefit from 35-MIPS, 8.4-MFLOPS, and 34-SPECmark performance at the lowest cost.* The highly integrated graphics system design produces graphical user interface (GUI) performance exceeding 4800 (geometric mean) on the X11perf benchmark and 450,000 2D X11 vectors per second.

Cost-sensitive applications requiring even greater performance will benefit from the 50-MHz Model 710 color workstation. It produces 59-MIPS, 12-MFLOPS, 49-SPECmark performance. Its GUI performance exceeds 7290 (geometric mean) on the X11perf benchmark and 500,000 2D X11 vectors per second. This makes it an ideal platform for low-cost, 2D electrical and mechanical computer-aided design.

* MIPS stands for million instructions per second. MFLOPS stands for million floating-point operations per second. The SPEC benchmark suite is a set of benchmark routines representing realistic tasks. The SPECmark number is the geometric mean of the SPECratios for all of the routines. The SPECratio is obtained by dividing a reference machine's time to complete a routine by the test machine's time. The reference machine is a VAX 11/780 computer.

Fig. 1. The HP 9000 Models 705 (right) and 710 (left and center) workstations are 35-MHz and 50-MHz entry-level members of the HP 9000 Series 700 family.

The Model 705/710 design produces an unparalleled level of desktop price, performance, and functionality in a quick-time-to-market design. Through the integration of system components and architecture and the leveraging of other Series 700 designs, the Model 710 was ready to ship in less than 12 months from the initial product concept. Both models support up to 64M bytes of memory, an 8-kHz-sample-rate voice-quality audio design, and built-in I/O interfaces including intelligent, 32-bit, DMA-capable SCSI II and IEEE 802.3 LAN. Both models also support both grayscale and color configurations.

This paper details aspects of the Model 705 and 710 design that were important in producing a single, reliable, low-cost, high-performance, grayscale and color desktop system design on an aggressive 12-month schedule. It also compares the processor and graphics performance of the Models 705 and 710 with respect to other desktop members of the Series 700 product family.

System Overview

The Models 705 and 710 are designed to meet the need for low-cost desktop systems while providing scalability to higher performance. A single system board design enables the Model 705 to offer a very low price, while frequency scalability enables the Model 710 to reach higher performance levels at approximately twice the price. The Model 705 is able to use lower-frequency components—PA-RISC CPU, floating-point coprocessor, and cache SRAMs—to meet lower-cost objectives. The Model 710 uses higher-frequency components to reach higher performance levels.

The single system board design for the Models 705 and 710 is shown in Fig. 2. It fits within a desktop package 14 inches deep by 16 inches wide by 3 inches high, with the system electronics integrated on a single printed circuit board. The package accommodates two 3.5-inch storage devices with fixed or removable media. To better accommodate multimedia applications, the package integrates a CD-ROM player with no increase in desktop area.

The Model 705 provides the following features:

- 35-MHz PA-RISC CPU
- 35-MHz PA-RISC 64-bit floating-point coprocessor (FPC)
- 32K-byte instruction cache
- 64K-byte copyback data cache, 64 bits wide
- 5-megabyte-per-second SCSI II interface (NCR53C700)
- 10-megabit-per-second IEEE 802.3 LAN interface (Intel 82596 with built-in ThinLan transceiver and AUI)
- 8M to 64M bytes of error correcting SIMM memory (8M bytes in diskless configuration only)
- Integrated 8-kHz, 8-bit, voice-quality audio
- 8-plane 1280-by-1024-pixel color, 8-plane 1024-by-768-pixel color, or 8-plane 1280-by-1024-pixel grayscale graphics
- Two integrated 3.5-inch 420- or 840-Mbyte fixed disk drives
- One of the following three removable media devices (uses a fixed drive position): CD-ROM, 3.5-inch PC-compatible flexible drive, or 3.5-inch 2.0-Gbyte DDS (digital data storage) tape drive.
- Two RS-232 modem-control serial ports
- 350-kbyte/s bidirectional Centronics port
- HP-HIL device interface for a keyboard and up to 6 other possible pointing or input devices.

The Model 710 enhances Model 705 performance and functionality with the following differences:

- 50-MHz PA-RISC CPU
- 50-MHz PA-RISC 64-bit floating-point coprocessor.

Processor Design

The processor consists of a PA-RISC CPU,[2] a PA-RISC floating-point coprocessor,[3] and separate instruction and data caches. It executes one instruction (integer or floating-point) on every clock cycle. The instruction cache and 64-bit data cache are accessed simultaneously, producing an execution rate exceeding 57 MIPS and 12 double-precision MFLOPS in the Model 710, and 35 MIPS and 8 double-precision MFLOPS in the Model 705.

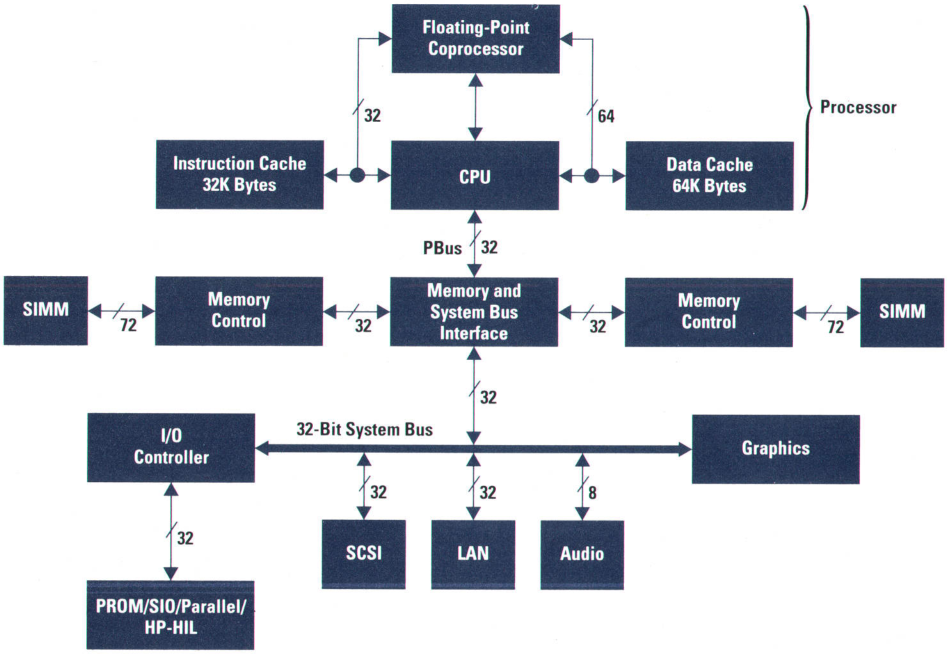


Fig. 2. Block diagram of the system board of the HP 9000 Models 705 and 710 workstations.

The instruction and data caches are separate, direct-mapped, virtually indexed, fast memories, each of which is capable of supporting an access every processor clock cycle. The instruction cache is 32 bits wide and 32K bytes in size, and is organized as 1024 lines of 32 bytes each. The data cache is similar in design, but is 64 bits wide and 64K bytes in size, and is organized as 2048 lines of 32 bytes each. The data cache also enforces a copyback memory update policy.

The CPU interfaces with the memory and system bus through the PBus, an interconnect defined for the PA-RISC CPU. This interface operates synchronously with respect to the CPU and supports memory data transfer rates up to 200 megabytes per second and I/O transfer rates up to 100 megabytes per second. The small signal count—53 signals—allows the design to interface easily with standard, low-cost, plastic quad flat package (PQFP) ASICs (application-specific integrated circuits).

Memory and System Bus Interface

A standard method used to connect the CPU, memory, and I/O subsystem in desktop systems involves the use of a three-state bus. A bus arrangement often suffers from the fact that a significant number of attachment points cause a large capacitive load—and the resulting low impedance—to be seen by a driving device. This results in what has been termed the bus driving problem.[4] It can often lead to slow operation in systems relying on standard low-cost TTL or CMOS interfaces.

In the Model 705/710 design, the CPU is connected to the memory and I/O system components through the memory and system bus interface. The memory and system bus interface implements a small buffered interconnect between the CPU, the memory, and the I/O system bus. It is the active equivalent of the shared signal bus, in that it connects the CPU, memory, and I/O subsystems. However, it has an advantage in that the separate subsystem interfaces are connected in a point-to-point fashion. In the Model 710 design, the isolated interconnect and resulting transmission line characteristics make it possible to operate the CPU and its PBus interconnect at 50 MHz using standard 8-mA TTL-compatible CMOS buffers.

Fig. 3 shows the memory and system bus interface and its connections to the CPU PBus, the memory controller ports, and the system bus. In the Model 710, the CPU PBus and the system bus interconnect operate at 50 MHz and 25 MHz, respectively. In the Model 705, the CPU PBus and the system bus interconnect operate at 35 MHz and 17.5 MHz, respectively. The memory and system bus interface is designed to exploit the capability of the CPU PBus through the use of two interleaved memory controller data ports, each having up to 100-Mbyte/s capability. The memory and system bus interface also provides the interface to memory for all I/O-initiated DMA transactions.

In an effort to reduce the size, cost, and power consumption of the design, logic normally found separate in the I/O controller interface is integrated into the memory and system bus interface. This includes byte accumulation and replication functions used in conjunction with the I/O controller to allow byte addressing of 8-bit-wide devices over the 32-bit system bus. The memory and system bus interface also contains logic to improve the performance of the CPU and ease the implementation of PBus interface protocol. This is done by including memory copyout and copyin buffers, and buffering the system bus interface to pipeline I/O and graphics write operations.

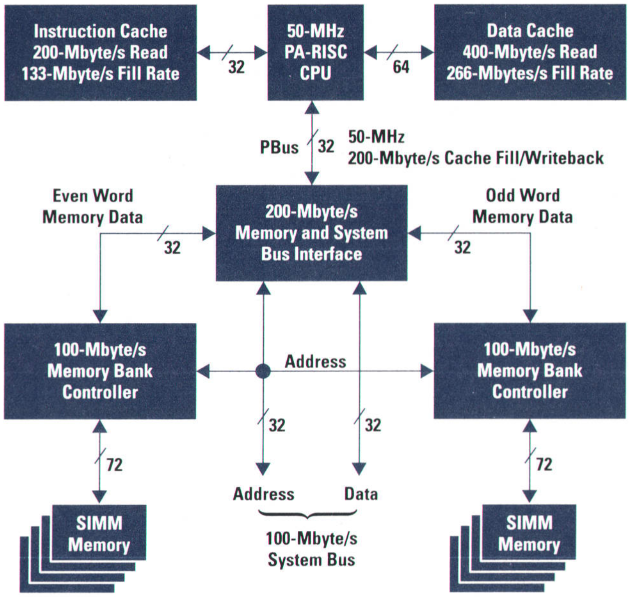


Fig. 3. Memory and system bus interface interconnect ports.

Fig. 4 shows the important blocks and data paths of the memory and system bus interface design.

The memory and system bus interface is implemented by two 160-pin, standard-cell ASICs. The ASICs are identical in design, fabricated in Hewlett-Packard's low-cost, 1-μm, CMOS34 process, and packaged in low-cost PQFPs. The memory and system bus interface ASICs operate from a single clock, creating a synchronous interface to the CPU, memory, and system bus. The memory and system bus interface operates at 35 MHz and 50 MHz, respectively, in the Models 705 and 710.

The memory and system bus interface is implemented with two smaller, plastic-packaged ASICs instead of one larger, more expensively packaged ASIC because of the need to limit total system cost, including ASIC processing and packaging cost, printed circuit board assembly cost, and system cooling cost. Distributing the power and signal I/O between two chips instead of one allowed the design to use HP's, low-cost, quick-time-to-market, CMOS34 ASIC process. This resulted in a small, low-cost, silicon die with an I/O pad ring that is evenly matched to the active area and capable of being inexpensively packaged in a surface mount plastic package.

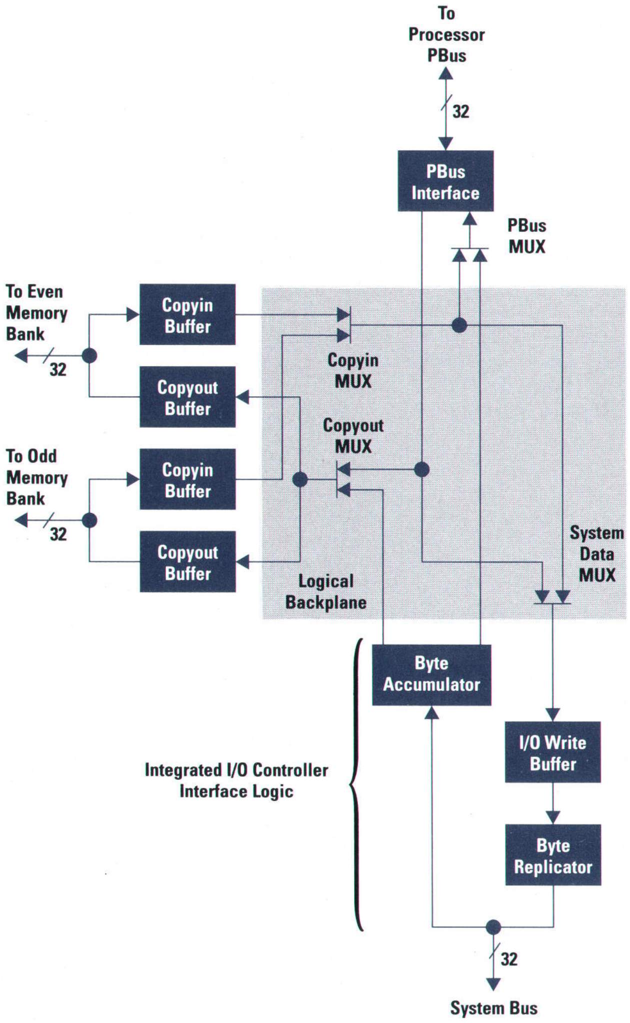


Fig. 4. Memory and system bus interface data path.

Table I provides an overview of the HP CMOS34 process parameters used in the memory and system bus interface implementation.

Table I
HP CMOS34 Process Parameters

N-channel FET L_{eff}	0.85 µm
N-channel FET L_{drawn}	1.0 µm
P-channel FET L_{eff}	0.85 µm
P-channel FET L_{drawn}	1.0 µm

	Interconnect Pitch	
	Contacted	**Uncontacted**
Metal I	3.8 µm	3.0 µm
Metal II	3.8 µm	3.0 µm

Table II shows some of the resulting specifications for the memory and system bus interface ASICs.

Table II
Memory and System Bus Interface Chip Specifications

Die Size	0.7 by 0.7 cm
Total Used Gate Count	18,000 (approximately)
Pads	160
Package	160-pin PQFP
Frequency	dc to 50 MHz
Operating Temperature (junction)	100°C
Operating Voltage	4.5 to 5.5V

Copyout Buffer. The memory and system bus interface incorporates memory copyout buffers to mitigate the effects of having "dirty" or modified data (resulting from the copyback cache architecture) written back to memory on a data cache miss. By buffering copyback data until the completion of the read miss, the read miss latency observed by the CPU is reduced significantly. This extra level of buffering is important, even for a low-end workstation, considering the probability that, on average, half of the data lines written to memory contain dirty data.[5]

Copyin Buffer. The memory and system bus interface also incorporates separate copyin buffers for each of the memory bank controllers. The memory subsystem is divided into two independent memory banks. These buffers are used to resolve asynchronous behaviors inherent in the design of the multibank memory implementation during memory read operations. If one of the memory banks is busy during a memory read, then the data from the other bank is buffered in the copyin buffer until the busy bank responds. At that time, data is transferred to the CPU.

I/O Write Buffer. The memory and system bus interface implements a pipelined, separate address and data, system bus. It is designed for the efficient transfer of randomly addressed data values to and from I/O and graphics devices. The pipelined nature of the bus allows a randomly addressed 32-bit data value to be transferred on every 25-MHz system bus clock cycle.

In support of the bandwidth potential of the system bus, an I/O write buffer is used within the memory and system bus interface to overlap CPU PBus transactions with system bus transactions. This makes it possible for the CPU to write to I/O and graphics devices at its maximum rate of 50 Mbytes/s. This is an important feature, considering that without it, X Window display performance would have been degraded significantly.

I/O Byte Accumulator/Replicator. The built-in I/O controller used in the Models 705 and 710 was designed for use in the HP 9000 Models 720, 730, and 750 systems. As part of the bus interface in those systems, the I/O controller manipulates latch-based data bus transceivers for the purpose of interfacing a 32-bit byte-aligned bus to 8-bit wide devices, such as PROM. In the Models 705 and 710 design, the latch-based data bus transceivers controlled by the I/O controller are an integral part of the memory and system bus interface design. This results in a substantial cost, power, and board area savings for the system.

System Clock Design

The Model 705/710 design relies on standard, low-power, low-cost, TTL clock circuitry to produce well-defined, controlled-skew clock signals for the CPU, memory and system bus interface, memory, graphics, and I/O controller. In the Model 705/710 design, fixed-relationship clock signals of 17.5 and 25 MHz and 35 and 50 MHz, respectively, are produced for all of the system devices. The low-power, low-cost design uses minimal printed circuit board area compared to alternative implementations relying on ECL logic.

CPU Clock. The Model 705/710 clock circuitry is designed to drive a combination of ECL and TTL clock input structures. The PA-RISC CPU has a differential ECL clock receiver. The higher-performance Models 720, 730, and 750 designs implement this clock circuitry using differential ECL logic (see article, page 23). This design provides precise clock skew control (controlled, predictable, edge rates and small propagation delay variation) and allows these models to operate at higher frequencies than the Model 710. However, these ECL designs tend toward high cost, power consumption, and printed circuit board utilization.

The Model 705/710 design takes advantage of the differential input characteristics of the CPU clock receiver to produce a single-ended TTL clock signal instead of a differential ECL clock pair. This TTL clock signal is generated by the system clock circuitry, delayed appropriately to meet the timing requirements of the CPU and memory and system bus interface designs, and then driven to one of the differential inputs of the CPU, which are terminated at the CPU. The terminations are determined by the transmission line characteristics and the limits imposed on the voltage swing of the clock signal. The voltage swing is limited to produce predictable rise and fall characteristics. This makes the threshold crossing more predictable with respect to the other differential input. The second differential CPU clock input is fixed by a voltage-compensated source at a voltage representative of the midpoint of the TTL clock signal swing, thereby producing a symmetrical clock signal within the CPU. Clock symmetry is important for higher-frequency systems because the PA-RISC CPU uses both rising and falling edges.

Delay Elements. Many system characteristics interact to establish the range of frequencies over which a synchronous design can operate. These include ASIC device characteristics, interconnect topology, and the design of the system clock circuitry. The Model 705/710 clock design relies on the characteristics of inexpensive TTL clock driver components to produce predictable output-to-output skew and delay. This establishes a fixed basic clock relationship. However, even with this basis established, it is often necessary to adjust each clock edge independently. This allows each clocked interface to be tuned to its maximum interconnect frequency, resulting in the optimal operating frequency for the system.

The Model 705/710 design employes discrete delay elements to adjust each clock independently to an optimal interconnect relationship. The delay elements used in the Model 705/710 design are implemented with discrete components as low-pass filters or transmission line segments. Two capacitors and an inductor make up each delay element. The values are chosen to preserve rise and fall characteristics. The design produces inherently predictable delay values with controlled edge and line matching impedance characteristics. Clock signals requiring longer delay values use multiple concatenated segments. The design is flexible, uses readily available components, and uses little printed circuit board area.

Memory Design

The goal for the Model 705/710 memory design was to provide a large, reliable, system memory that would produce leading-edge performance at very low cost. The design supports from 8M to 64M bytes of memory using 72-pin commercially available single inline memory module (SIMM) cards. The memory design takes advantage of the CPU's copy-back cache architecture (CPU-initiated memory operations are always 32-byte cache blocks) to reduce the complexity and cost of implementing memory error detection and correction.

Memory Error Correcting Code. The Model 705/710 memory design calculates the error correcting code on a 64-bit data word to minimize the number of DRAMs needed in the design. This results in significant cost savings compared to an implementation calculating the error correcting code on a 32-bit word. The 64-bit error correcting code requires only 8 bits of DRAM while the 32-bit error correcting code requires 14 bits of error correcting DRAM for the same amount of protected memory. Additionally, compared to simple parity checking designs used in other desktop systems, the 64-bit error detection and correction achieves the same DRAM cost, but with much greater reliability. The Model 705/710 design corrects all single-bit errors on the fly and detects all double-bit errors. Detection of all triple-bit and quadruple-bit errors in any adjacent 4 bits makes it possible to detect a single failing 4-bit wide DRAM.

Memory Bank Controller. The memory design consists of two independent memory bank controllers. Each memory bank controller supports 4 DRAM SIMMs for a system total of 8 SIMMs. Two SIMMs are required per controller to form a 64-bit error detection and correction memory word. The combination of two memory bank controllers forms a 128-bit memory word and has a cycle time of 80 ns (or 200 megabytes per second) in the Model 710 design. The memory and system bus interface acts as the central memory controller, working in conjunction with the two memory bank controllers to assemble memory data into CPU cache blocks during reads and distribute data to the bank controllers during memory writes.

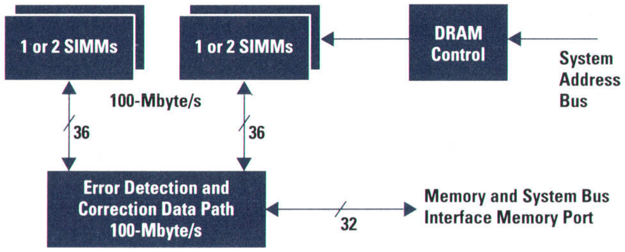

1 or 2 SIMMs
1 or 2 SIMMs
DRAM Control
System Address Bus
100-Mbyte/s
36
36
Error Detection and Correction Data Path 100-Mbyte/s
32
Memory and System Bus Interface Memory Port

Fig. 5. Memory bank controller.

The memory and system bus interface interleaves the memory bank controllers on 4-byte boundaries, producing even and odd memory banks. This interleaving makes it possible to overlap sequentially addressed 32-bit memory words by directing them to alternate memory bank controllers. In this way, read-modify-write behavior resulting from a 32-bit DMA write (the error correcting code is calculated on a 64-bit memory word) is hidden. This element of the design provides the necessary DMA memory bandwidth along with low-cost error detection and correction.

Fig. 5 is a block diagram of a memory bank controller. It consists of a DRAM controller ASIC, an error detection and correction data path ASIC, and SIMM memory. The bank controller receives its address information from the system address bus, while control is directed by the memory and system bus interface. Timing and control to address, access, and refresh locations in the memory banks are provided by the DRAM controller ASIC. The detection and correction of errors and the funneling of 64 bits of DRAM data and 8 bits of error correcting code into 32 bits are provided by the error detection and correction data path ASIC. Each ASIC is a low-cost HP CMOS34 circuit packaged in a 160-pin PQFP.

Graphics Design

In addition to CPU performance, graphical user interface (GUI) performance is often important in characterizing application throughput. Today's desktop applications are built on standard window environments like the X Window System to increase their ease of use, to provide a common look and feel to the user, and to allow greater application portability. Providing high levels of X Window performance requires a balanced system design characterized by high levels of integer and floating-point performance, efficient virtual memory support, fast memory-to-frame-buffer data transfers, and a fast frame buffer. The Models 705 and 710 support this balance by moving much of the graphics pipeline into the CPU so that graphics performance gains by leveraging CPU performance. Overall system cost is decreased by eliminating redundant hardware.[6]

Model 705/710 graphics support consists of eight planes, a cursor, and a 256-entry color look-up table. Three configurations are available: 1280-by-1024-pixel color or grayscale and 1024-by-768- pixel color. The Models 705 and 710 use the same graphics architecture as the rest of HP's Series 700 family. The design also leverages significant amounts of hardware and software from the Models 720, 730, and 750.

Fig. 6 shows the Model 705/710 graphics subsystem. There are four blocks of dedicated graphics hardware: the system bus interface, the frame buffer controller, the frame buffer memory, and the video data path.

System Bus Interface. The Model 705/710 graphics system bus interface resides in ã single gate array on the system board. This ASIC accepts commands from the system bus. It generates commands for two independent buses. One is a high-speed interface to the frame buffer controller, while the other is an auxiliary bus to the video data path. The high-speed interface between the system bus interface and the frame buffer controller follows a typical sequence. First, the system bus interface defines a data mode, which tells the frame buffer controller how all subsequent data is packed on the bus. For example, one data mode sends four 8-bit pixels down the 32-bit bus. Next, the system bus interface sends a beginning frame buffer address. Finally, a sequence of data packets follows, accompanied by an address modification rule. The frame buffer controller uses this address modification rule to synthesize the addresses of subsequent pixels.

The system bus interface provides a second important function—it contains hardware for vector, rectangle fill, and block move command generation. Software sets up the hardware with information about the particular operation, and the hardware then synthesizes the necessary commands for the frame buffer controller. X Window library calls take advantage of this drawing support when possible, with software implementing the balance of the commands. 2D and 3D wireframe applications are segmented so that the CPU performs display file and geometry processing and the system bus interface performs vector generation.[6]

Frame Buffer Controller and Memory. The Models 705 and 710 implement the frame buffer controller in a second gate array on the system board. The frame buffer resides in twelve 1M-bit VRAM (dual-ported video DRAM) chips. This provides storage space for 1280 × 1024 8-bit pixels as well as offscreen memory for a shadow color look-up table and the cursor bitmap.

The frame buffer controller accepts commands from the system bus interface and builds six-pixel tiles to access the frame buffer. This means that every frame buffer memory cycle can read or write up to six pixels in parallel. A single memory controller provides the protocol for all of the frame buffer VRAM. The frame buffer controller organizes the memory so that each VRAM row contains two scan lines of

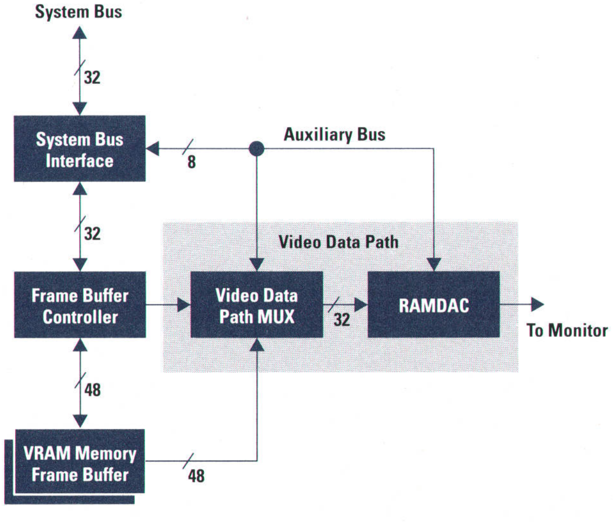


Fig. 6. Model 705/710 graphics.

information. With increasing rectangle fill and block move sizes, the frame buffer controller becomes quite efficient because it fully packs the six-pixel tiles, and it maintains memory page mode.

A VRAM-based design provides several advantages for the graphics subsystem. VRAM has two independent and asynchronous ports: a random port and a serial port. The frame buffer controller reads and writes the memory array through the random port. At the same time, the video data path reads the serial port for screen refresh. This simplifies the design of the frame buffer controller because the frame buffer controller no longer has to arbitrate between random port reads and writes and screen refresh serial reads. Also, this isolation of screen refresh accesses increases random port bandwidth between the frame buffer controller and the frame buffer. The frame buffer controller takes advantage of other features of the VRAMs to increase the frame buffer fill rate.

Video Data Path. The video data path consists of two ICs: a third gate array (video data path multiplexer) and a Brooktree Bt458 RAMDAC. The video data path is the output end of the graphics subsystem, and its primary responsibilities are screen refresh and look-up table support.

The gate array receives data from the serial ports of the frame buffer VRAMs, and receives control from the frame buffer controller. It accepts six pixels in parallel and serializes the data to four pixels in parallel before sending it to the RAMDAC. This gate array also detects when the cursor should be displayed and shifts this data out in place of regular image data.

The RAMDAC contains the 256-entry look-up table. It also performs the final serialization of the frame buffer data, and converts the resulting full-speed pixel stream to analog signals that drive the monitor.

Audio Design

Through the integration of audio and video capabilities on the desktop, multimedia applications can provide important ways to improve user productivity. Although many possibilities exist for the application of integrated video and CD audio, studies indicate that voice-quality audio is the single most important capability needed for improving user productivity. For this reason, the Models 705 and 710 include an inexpensive voice-quality audio design as part of the base product, instead of more complex and costly audio and video capabilities.

The Models 705 and 710 are designed with the built-in capability to record and play back voice-quality sound, where voice-quality means an 8-kHz sample rate and 8-bit or 16-bit samples. An internal speaker, a headphone connector, and a microphone connector are provided as audio destinations and sources. The audio design stresses low cost, while providing the functionality necessary for voice-quality audio applications. These applications include voice annotation, voice notification, computer-based training, and emerging application areas such as verbatim recording and audio command and control.

Audio Software Support. The audio subsystem is designed to work within a client/server configuration with the client running either remotely or locally. Software support for audio application developers is provided as a standard part of HP-UX* operating system release 8.07, and consists of device drivers, an audio server, an audio data type conversion library (CUlib), an audio library (Alib), an audio toolkit, and a widget set. The audio library architecture is similar to Xlib in the X Window System; this aids in the integration of audio into X Windows applications.

The audio library provides the application program interface to the audio application. An application can manipulate the characteristics of the audio device, prepare an audio file to be played, and tell the audio device to play the audio file via calls provided in the audio library. Similarly, an application can record audio data into an audio file. Audio data streams and raw audio data are supported by the audio library, which allows an application to play or record without having data files. An audio toolkit and audio widgets are designed to work well within the X toolkit and widgets environments.

Audio Hardware Design. The low-cost nature of the Model 705/710 audio design led to the use of a telecommunications CODEC (coder/decoder) with 8-bit u-law, 8-bit A-law, and 16-bit linear sigma-delta conversion, software-programmable gains, filtering and output destination selection, and ring and tone generators. The CODEC provides the analog-to-digital and digital-to-analog front end of a telephone/modem/fax connection with sufficient quality for voice recording and playback. Fig. 7 shows a diagram of the Model 705/710 audio subsystem.

The two 8-bit conversion algorithms (u-law and A-law) are voice compression standards for North America and Europe, respectively. Both compress a 13-bit sample into an 8-bit sample without losing significant voice-content information. A 16-bit mode can be used for future modem/fax front ends. The converter itself is of a delta-sigma design, which makes it possible to use inexpensive single-pole reconstruction and anti-aliasing filters rather than more precise filters which are more expensive in both cost and printed circuit board area. Additional anti-aliasing filtering is done with a built-in digital signal processor that also provides separate receive and transmit gain or attenuation and some frequency equalization.

The CODEC is connected to the processor through a 128-byte transmit FIFO buffer, a 128-byte receive FIFO buffer, and a 16-byte command FIFO buffer. The transmit (out to speaker or headphones) FIFO interrupts the processor

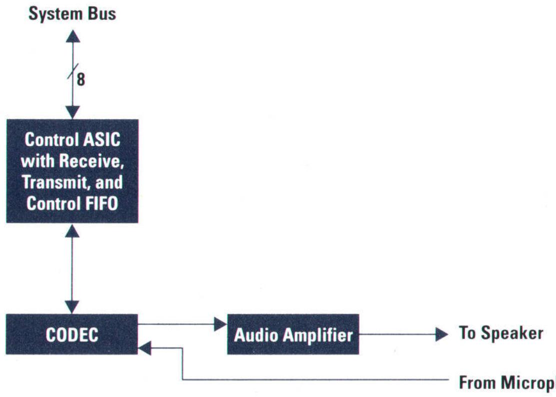


Fig. 7. Audio CODEC (coder/decoder) and FIFO buffer diagram.

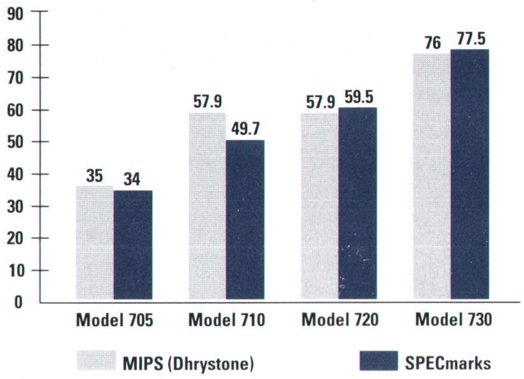


Fig. 8. Processor performance comparison.

whenever the FIFO is less than half full so that processor interruption frequency is minimized. Similarly, the receive (data in from microphone) FIFO interrupts the processor when the FIFO is more than half full. At the 8-kHz sample rate, in 8-bit mode, the processor has 8 ms to respond to the interrupts before samples are dropped. The operation of these FIFOs can be fully overlapped with the command FIFO, which is used to send control information to the CODEC to set gains, filter coefficients, output destinations, and so on.

Input/Output

To minimize the overhead typically associated with managing less sophisticated interface controllers, the Model 705/710 design incorporates intelligent controllers for the SCSI II and LAN functions. These controllers access memory using efficient 32-bit DMA interfaces to reduce data latency and minimize system bus and memory use.

The SCSI controller used in the Model 705/710 is the NCR53C700. This single-chip controller manages the single-ended SCSI II bus protocol and provides an intelligent interface to system software. A 32-bit DMA engine, coupled with an internal 32-byte FIFO, is used for efficiently moving data to and from the 5-Mbyte/s SCSI bus and main memory.

The LAN controller used in the Model 705/710 is the Intel 82596DX. This single-chip controller, along with an Intel 82C501 encoder/decoder chip, implements the LAN auxiliary unit interface (AUI). A transceiver is used for the ThinNet BNC connection. The controller manages the LAN protocol and provides an intelligent interface to system software. A 32-bit DMA engine, coupled with internal 128-byte receive and 64-byte transmit FIFOs, produces an efficient interface to system memory.

The Model 705/710 keyboard and up to six other HP-HIL devices connect to the system through the HP-HIL interface. The design also makes available two full modem-control RS-232 ports. These ports are implemented with 9-pin D-sub connectors. For high-speed parallel communication, the design implements a bidirectional version of the Centronics interface, allowing for the connection of input devices such as the HP ScanJet scanner.

Performance

The "true" performance of a workstation from any particular user's perspective depends heavily on the application workload to which the system is exposed. For the purposes of this paper, we depend on certain benchmarks that in many cases approximate the characteristics of typical application workloads. We have chosen the SPEC and Dhrystone benchmarks to characterize the processor and memory performance of the system, and the X11perf benchmark and some peak vector performance numbers to characterize the graphics performance. We have selected the desktop HP 9000 Model 720 and 730 workstations for our processor and graphics performance comparisons because of their desktop performance leadership.

A comparison of the processor performance of the Models 705 and 710 and other desktop members of the Series 700 workstation family is shown in Fig. 8.

Fig. 9 shows the relative X11 and 2D/3D vector graphics performance of the Model 710 and other desktop Series 700 family members.

Conclusion

The HP 9000 Model 705 is an entry-level, low-cost, desktop color workstation providing high levels of application performance. The system offers attractive levels of price/performance for cost-sensitive users and applications. The design produces a balance of integer, floating-point, memory, graphics, audio, and I/O capabilities at an affordable desktop price. The Model 710 is a desktop color workstation that provides very high levels of performance for cost-sensitive customers.

The development project for the Models 705 and 710 relied on a single design and standard ASIC technology instead of custom VLSI to provide high performance, fast turnaround, and low cost. This resulted in a 50-MHz design that was completed and released for shipment in just under 12 months from initial concept. Other factors contributing to this short development included the use of high-level hardware description languages, the realization of gates through

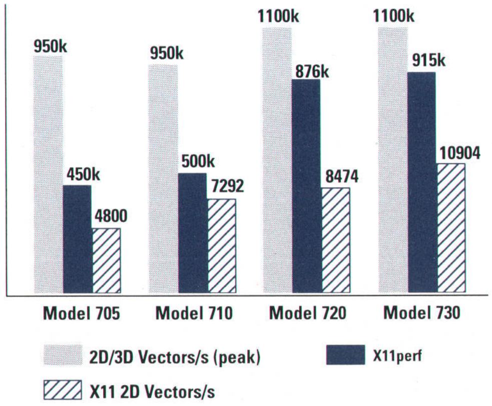


Fig. 9. Graphics performance comparison.

synthesis techniques, and the ASIC design and release methodology employed by Hewlett-Packard. Hardware ASIC emulation was also used to supplement the logic design verification portion of the development to guarantee the correct design of our ASICs. The system board and first-pass ASICs were fully operational and running the HP-UX operating system in a networked environment (with X Windows) just minutes after delivery of the first and only pass of silicon.

Acknowledgments

The authors would like to thank the many individuals involved in making the Models 705 and 710 possible. These include those who contributed directly to the project, as well as those who contributed indirectly through design efforts that were leveraged. These leveraged designs included the other Series 700 PA-RISC products, as well as the Series 400 Motorola 68K products.

References

1. R.B. Lee, "Precision Architecture," *IEEE Computer*, January 1989.
2. M. Forsyth, et al, "CMOS PA-RISC Processor for a New Family of Workstations," *COMPCON Spring 91 Digest of Technical Papers*, February 1991.
3. Steiss, et al, "A 65 MHz Floating Point Coprocessor for a RISC Processor," *ISSCC Digest of Technical Papers*, February 1991.
4. R.V. Balakrishnan, "The Proposed IEEE 896 Futurebus—A Solution to the Bus Driving Problem," *IEEE Micro*, August 1984.
5. A.J. Smith, "Cache Evaluation and the Impact of Workload Choice," *Proceedings of the 12th International Symposium on Computer Architecture*, Boston, June 17-19, 1985, pp. 64-73.
6. C. Dowdell and L. Thayer, "Scalable Graphics Enhancements for PA-RISC Workstations," *COMPCON Spring 92 Digest of Technical Papers*, February 1992.

The HP-UX operating system is based on and is compatible with UNIX System Laboratories' UNIX* operating system. It also complies with X/Open's* XPG3, POSIX 1003.1 and SVID2 interface specifications.

UNIX is a registered trademark of UNIX System Laboratories Inc. in the U.S.A. and other countries.

X/Open is a trademark of X/Open Company Limited in the UK and other countries.

Low-Cost Plain-Paper Color Inkjet Printing

The HP DeskWriter C and DeskJet 500C printers are based on advanced thermal inkjet technology in the form of a 300-dpi three-color inkjet print cartridge. The printers and software drivers that use this cartridge were developed on an aggressive one-year schedule.

by Daniel A. Kearl and Michael S. Ard

The use of color in written communications has revolutionized the printed-output industry. The ability of vivid colors to draw attention and stimulate retention is immense. High-density color monitors and high-quality color applications for personal computers have reset customer expectations. However, an affordable, plain-paper color printing solution has not been available.

The introduction of the HP DeskWriter C printer in the summer of 1991, followed by the HP DeskJet 500C printer, set new standards for low-cost color printing.[1,2] At the heart of these printing solutions is HP's three-color 300-dpi thermal inkjet print cartridge. This print cartridge provides the printer with the ability to deliver high-quality 300-dpi color images and graphics on a wide variety of "plain" office papers. This 300-dpi plain-paper color capability represents a major increase in price/performance for personal printers. It is a good example of a technology-enabled performance increase. Fig. 1 shows the HP DeskWriter C and DeskJet 500C printers and Fig. 2 shows examples of their output.

Color Print Cartridge

The color print cartridge for the HP DeskJet 500C and DeskWriter C printers is another extension of HP's thermal inkjet printing technology (see "Thermal Inkjet Review...," page 67 and references 3, 4, and 5). This technology was first made available in 1984 with the introduction of the HP ThinkJet line of printers. These printers initially used a black print cartridge that had 96-dpi resolution and required special paper. Later enhancements brought plain-paper and single-color printing to the personal printer user. In 1987 HP introduced the HP PaintJet series of printers. These printers provided a fully integrated color printing capability on special paper at 180-dpi resolution. The price and performance of the HP PaintJet printers represented a significant advance at the time of their introduction. The HP DeskJet line of printers was introduced in 1988. These printers offered high-quality 300-dpi black printing on a wide variety of office paper types. As noted above, the color-capable versions of the DeskJet family were introduced in the latter part of

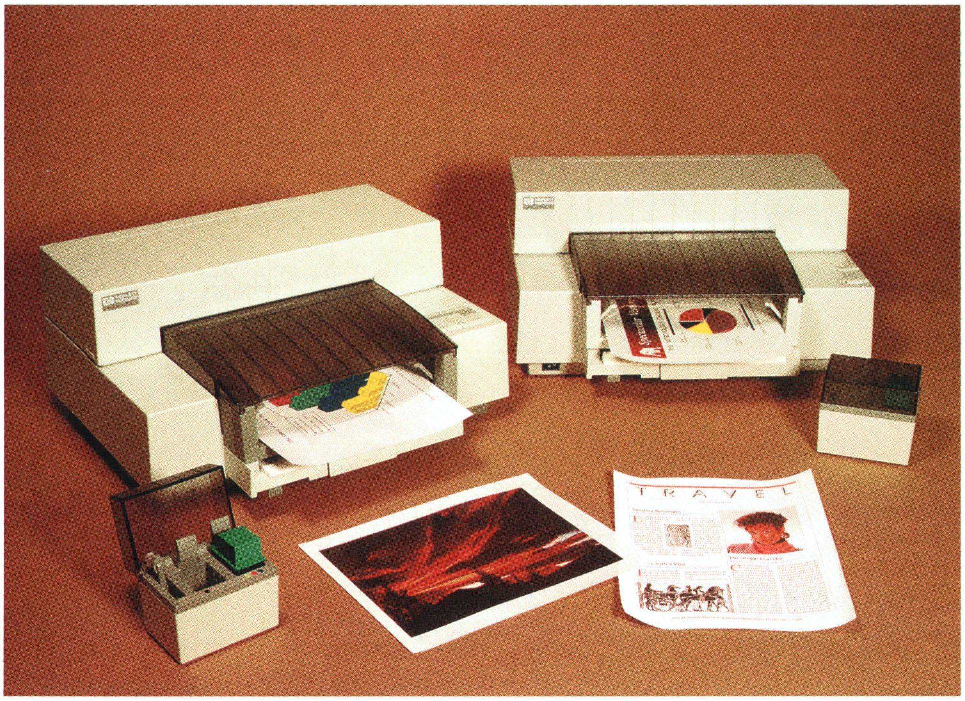


Fig. 1. HP DeskWriter C and DeskJet 500C color thermal inkjet printers.

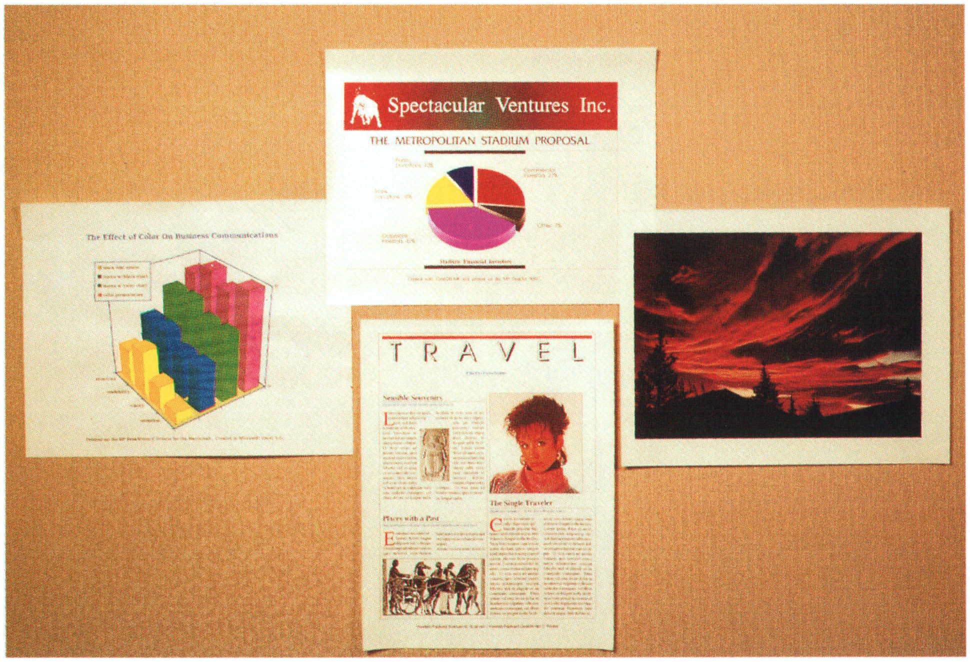

Spectacular Ventures Inc.
THE METROPOLITAN STADIUM PROPOSAL
The Effect of Color On Business Communications
T R A V E L
Sensible Souvenirs
Places with a Past
The Single Traveler

Fig. 2. Examples of HP DeskWriter C and DeskJet 500C printed output.

1991, the HP DeskWriter C printer for Macintosh computers and the HP DeskJet 500C for the PC. Print cartridge technology development has played a critical role in the successful introduction of each of these generations of thermal inkjet printers.

Reduced to the simplest terms, the HP DeskJet 500C/DeskWriter C color print cartridge is a compact, low-cost, high-resolution color dot generator. The printer dictates to the print cartridge when and where to deliver the color dots. This particular cartridge delivers three different colors of dots: cyan, yellow and magenta. These colors are known as subtractive primary colors. The size, shape, and optical properties of the dots produced should ideally be independent of the media and the printing environment. These dots should be delivered to the media at a very high rate of speed with a positional accuracy commensurate with the high resolution of the device. All of the dot generator properties should remain constant over the storage and printing life of the print cartridge. These relatively simple color dot generator performance goals represented a major development challenge for the HP Inkjet Components Division.

In outward appearance, the color print cartridge for the HP DeskJet 500C and DeskWriter C printers is very similar to the original HP DeskJet black print cartridge (Fig. 3). The plastic body of the print cartridge has been enlarged somewhat to provide room for the three individual ink reservoirs. Electrical interconnection to the printer is accomplished using the same flex circuit technology as the black print cartridge. The location of the pressure interconnection pads is identical to those on the black print cartridge. Provisions have been made in the printer drive electronics to sense which cartridge has been installed and respond with the appropriate drive signals.

In general, the thermal inkjet drop generator portion of the color print cartridge for the HP DeskJet 500C and DeskWriter C represents a natural extension of the existing HP DeskJet black print cartridge technology. Careful inspection of the gold-plated nickel nozzle assembly shows that the nozzles have been arranged into three groups or *primitives*, one for each of the subtractive primary colors (Fig. 4). The three primitives all share a common silicon thin-film substrate and circuitry. This thin-film substrate is fabricated using processes very similar to those used for the HP DeskJet black and HP PaintJet print cartridges.[3,4] The primitives are staggered with respect to one another to provide separation for the ink delivery channels on the back side of the substrate. Each of the primitives consists of two columns of eight nozzles each. The nozzles within a given column are spaced on 1/150-inch centers. Each column in a given primitive is offset with respect to its neighbor by 1/300 inch, so that the combination of the two nozzle columns results in an array of 16 nozzles with a vertical resolution of 1/300 inch.

The major technology contributions in this color print cartridge are in the areas of ink chemistry and manufacturing processes. Ink formulation is the key to producing high-quality plain-paper color images and graphics. A discussion of the nature of plain paper has been presented in an earlier

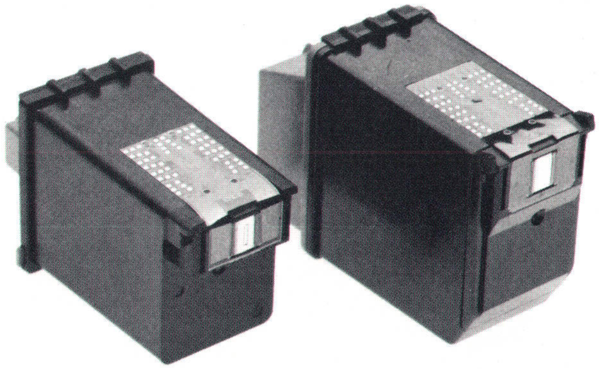


Fig. 3. The original HP DeskWriter/DeskJet 500 black print cartridge (l) and the new three-color HP DeskWriter C/DeskJet 500C 300-dpi print cartridge (r).

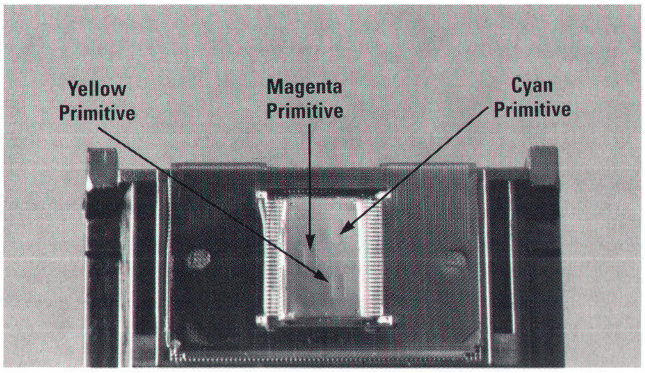


Fig. 4. The nozzles of the three-color print cartridge are arranged in three groups or *primitives*. Each primitive prints one of the three primary colors: cyan, magenta, and yellow.

issue of this publication.[6] The very substantial plain-paper color ink formula development effort is the subject of the article "Ink and Print Cartidge Development for the HP DeskJet 500C/DeskWriter C Printer Family," which follows this article. Several challenges unique to color printing were met in this ink development project.

The additional complexity of three independent ink reservoirs presented substantial challenges in the print cartridge manufacturing area as well. These challenges are discussed in three articles. The first, "Automated Assembly of the HP DeskJet 500C/DeskWriter C Color Print Cartridge" (page 77) provides insight into several of the problems of high-volume manufacturing of this three-color thermal inkjet device. Machine vision is extensively used in these manufacturing operations, and is the topic of the article "Machine Vision in Color Print Cartridge Production" on page 87. Finally, the demanding applications of adhesives technology are discussed in the article "Adhesive Material and Equipment Selection for the HP DeskJet 500C/DeskWriter C Color Print Cartidge" on page 84.

Fast-Track, Leveraged Product Development

The HP DeskWriter C and DeskJet 500C printer designs were leveraged from the successful HP DeskWriter and DeskJet 500 printers, both high-quality 300-dpi monochrome products. Leveraging from successful products still required enormous effort to deliver color printing solutions to market in approximately one year. Prioritized and focused program execution by all departments and functional areas was critical in accomplishing the following major steps in this aggressive development program.

First, there was an immediate focus on key feature-set requirements and crucial program objectives. The urgent market opportunity drove the schedule as the primary objective. Early customer research in the form of extensive phone screening and user focus groups provided important data on required features, applications, and pricing. Usability studies were also conducted to verify the acceptability of a one-cartridge color-printing solution in terms of both ease of use and throughput performance. With this data, the product team identified the critical market "must" features, leaving the balance of design opportunities for future product revisions or new product developments. This research data also facilitated trade-offs required late in the development stages to meet aggressive schedule objectives.

Next, there was intense execution of the program objectives by the development staff. Participating on a leading-edge product with such apparent potential provided great team motivation. At the heart of this team effort was an experienced core of engineers whose insights and awareness were invaluable in making real-time design decisions and trade-offs to support a market-driven schedule. The product development teams were organized into two primary groups: printer-product and environment-driver development. Printer-product development consisted of a mechanical team and an electronic and firmware team. Environment-driver development included a Macintosh driver team and a Microsoft® Windows driver team.

Each development group faced significant design issues. The printer-product group focused on supporting the new color and monochrome print cartridges. The mechanical team focused on physical pen support, including carriage mounting, movement, electronic interconnect, service and storage of the print cartridges, plus enhanced media flexibility and support. The electronic and firmware team pursued development of color support algorithms and logic. These required key architectural enhancements and additions, plus complex host interaction and task partitioning.

The environment-driver group supported the majority of the color science requirements, including intelligent formatting, rasterizing, color matching and dithering algorithms for optimum output results (see "HP DeskWriter C Printer Driver Development," page 93). The contribution of the environment drivers in providing an extensive, high-quality, color-optimized solution was of major significance. An example of this is the printers' support of over sixteen million unique colors based on 24-bit color data. The drivers also provide ways to overcome limitations of the printhead and ink technologies. An example is the extension of the environmental print range by providing user-adjustable intensity or ink depletion settings for the output.

Because of the tight schedule, the development activities tended to parallel standard textbook approaches to fast-track product development. There was a clear market focus, clearly defined program objectives, a firm product definition, total site commitment to the development effort, experienced development and management teams, process flexibility, and significant leverage opportunity. Unique to this product development were several activities that had significant impact on minimizing schedule and reducing design risk.

First, the HP DeskWriter C printer for the Macintosh environment was targeted for initial introduction, with the DeskJet 500C for the PC market offset by two months. This allowed initial attention to be focused on the Macintosh environment solution. Key developments and insights from the Macintosh printer and driver were then leveraged and applied to the PC environment printer and the Microsoft Windows driver.

Next, effort was made to ensure that ample engineering resources were available for this fast-track program, especially on the mechanical team. This provided flexibility to move development resources where they were most urgently needed, when they were needed. Understaffing fast-track development projects is a major failure risk.

(continued on page 68)

Thermal Inkjet Review, or How Do Dots Get from the Pen to the Page?

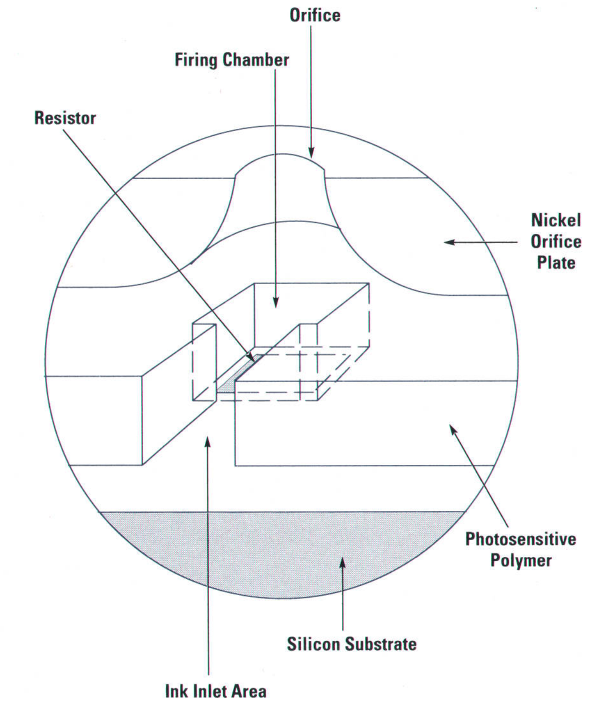


Fig. 1. An exploded cross-sectional view of a single inkjet nozzle.

The print cartridge for the HP DeskJet 500C/DeskWriter C printers delivers dots to the page using thermal inkjet technology. The fundamental technology is the same as that used in previous Hewlett-Packard inkjet products such as the HP ThinkJet, PaintJet, and DeskJet printers. Detailed descriptions of these products[1-4] and more in-depth discussions of thermal inkjet theory and background[1,5] can be found in the literature.

In its simplest form, an inkjet device consists of a tiny resistor aligned directly below an exit orifice. Ink is allowed to flow into the resistor area, and when the resistor is heated, the ink on the resistor essentially boils and forces a tiny droplet of ink out of the aligned orifice. This is called *firing* the nozzle.

A cross-sectional view of a single inkjet nozzle is shown in Fig. 1. On the floor of the firing chamber is a resistor. This resistor is patterned onto a silicon substrate using conventional thin-film fabrication procedures. Leads are connected to the resistor through the thin-film substrate. These leads ultimately travel out to the flexible circuit on the body of the print cartridge, through which a voltage can be applied across the resistor. The resistor is the heart of the thermal inkjet device and the size of the resistor is the primary factor governing the volume of the ejected droplets.

The walls of the firing chamber are made up of a photosensitive polymer. This polymer serves to define the walls of the firing chamber and determines the spacing between the resistor surface and the orifice. The thickness of this photosensitive barrier and the dimensions of the firing chamber are critical to the production of a well-formed droplet.

The photosensitive polymer also defines the dimensions of the inlet area to the firing chamber. Ink enters into the firing chamber through this inlet area. Like the barrier thickness, the inlet dimensions greatly affect the characteristics of the ejected droplet.

Finally, a gold-plated nickel orifice plate sits on top of the barrier. An orifice is formed in this plate directly above the firing chamber. This orifice hole is formed using an electroforming process. The diameter of the orifice has a direct bearing on the volume and velocity of the ejected droplets.

To fire a drop, a voltage pulse is applied across the resistor. This pulse is typically very short, on the order of 2 to 5 microseconds in duration. The voltage pulse causes the resistor to heat up, temporarily bringing the resistor surface to temperatures up to 400°C. Heat from the resistor causes ink at the resistor surface to superheat and form a vapor bubble. Formation of this vapor bubble is a fast and powerful event and expansion of the bubble forces some of the ink in the firing chamber out of the orifice at velocities of typically 10 meters per second.

By the time a droplet is ejected, the resistor has cooled down and the vapor bubble has collapsed. Through capillary forces, more ink flows into the firing chamber through the inlet area, thus readying the system for the firing of another droplet. The frequency at which the printhead can repeatedly fire droplets is determined by several factors including the inlet dimensions, the barrier thickness, and the fluid properties of the ink.

The device described above is essentially a droplet generator. The device designer has a fair amount of control over the characteristics of the ejected droplets. For example, the volume of the ejected droplet can be controlled by changing the size of the resistor—bigger resistors give droplets of larger volumes. In addition, the diameter of the orifice can be used to control droplet volumes. Droplet velocity is also controlled primarily by the diameter of the orifice.

The frequency at which droplets are ejected can be controlled by altering the size and shape of the barrier and by changing the rheological properties of the ink. For the particular design chosen for the HP DeskJet 500C/DeskWriter C printhead, and for the ink fluid characteristics, the typical operating frequency of the printhead is about 3 kHz.

Droplet characteristics, as they relate to print quality on the media surface, can be optimized through careful control of orifice profiles and resistor/orifice alignment. Ink, which also has a dominant effect on print quality, is discussed in more detail in the article on page 69. Ink properties such as surface tension, viscosity, and thermal stability all play important roles in the production of useful droplets.

In real-life inkjet printheads, multiple nozzles are lined up on a single silicon substrate/orifice plate assembly to form an array of droplet generators. Fig. 4 on page 66 shows the layout of nozzles for the HP DeskJet 500C/DeskWriter C print cartridge. These nozzles can be fired in rapid succession as the printhead is scanned across a sheet of paper or other media. The firing of these multiple nozzles at high frequencies under control of a microprocessor can produce high-resolution, high-quality text and color graphics at scan rates of 10 to 20 inches per second. Thus we see that thermal inkjet technology offers a compact, highly tunable method for delivering droplets from the pen to the page.

References

1. R.R. Allen, J.D. Meyer, and W.R. Knight, "Thermodynamics and Hydrodynamics of Thermal Ink Jets," *Hewlett-Packard Journal*, May 1985, pp. 21-27.
2. J.P. Baker, D.A. Johnson, V. Joshi, and S. J. Nigro, "Design and Development of a Color Thermal Inkjet Print Cartridge," *Hewlett-Packard Journal*, August 1988, pp. 6-15.
3. R.A. Askeland, W.D. Childers, and W.R. Sperry, "The Second-Generation Thermal Inkjet Structure," *Hewlett-Packard Journal*, August 1988, pp. 28-31.
4. W.A. Buskirk, D.E. Hackleman, S.T. Hall, P.H. Kanarek, R.N. Low, K.E. Trueba, and R.R. Van de Poll, "Development of a High-Resolution Thermal Inkjet Printhead," *Hewlett-Packard Journal*, October 1988, pp. 55-61.
5. N.J. Nielsen, "History of ThinkJet Printhead Development," *Hewlett-Packard Journal*, May 1985, pp. 4-10.

James P. Shields
Research & Development Chemist
InkJet Components Division

A substantial and thorough testing effort was put in place early in the program. Specific and tailored test processes covered both the printer-product and the environment-driver developments. This investment allowed early identification of design defects and concerns. Where changes were required, early identification enabled easier and more cost-effective solutions while minimizing schedule impact.

Aggressive schedules were broken into short development-based milestones. This allowed close monitoring of progress, and kept all developmental activities visible. Issues and concerns were monitored at the program level on a weekly basis to ensure that problems were addressed and resolved expeditiously.

Finally, frequent prototyping throughout the development cycle allowed immediate evaluation of design solutions. Prototype builds in large numbers provided printer units to facilitate all cross-functional development, testing, and extended support activities. To meet fast-track, aggressive schedules, an organization must be willing to spend money—especially on prototypes in sufficient quantities to meet a large demand. Outstanding support by manufacturing and the model shop in this regard was crucial. Similarly, early availability of print cartridges in large quantities was essential to keeping fast-paced development activities on schedule throughout the design, implementation, and test cycles.

Acknowledgments

The success of these products can be attributed to the major development efforts and cooperation of two HP divisions: the Inkjet Components Division (ICD) in Corvallis, Oregon, and the Vancouver, Washington Division (VCD). ICD developed the color printing technology in the form of a new color cartridge containing high-resolution, plain-paper inks. VCD adapted this technology to printer products and host environments to produce total system solutions via fast-track, leveraged product development. Special assistance was also rendered by HP Laboratories and the San Diego Printer Operation (SPR).

The dedicated support, direction, and cooperation of division and functional management from ICD and VCD was a prime factor in the success of this program. Recognition and appreciation go to the program management staff, including Bob McClung, Ron Prevost, and Tom Braun. The contributions of other management staff and their respective organizations were significant, particularly those of John DiVittorio, Gary Cutler, Dan Weeks, Joe McGuckin, John Coyier, Melissa Boyd, and Bob Conder. There were also numerous individual contributors whose efforts, accomplishments, and personal sacrifices were key in this most aggressive program undertaking.

References

1. “DeskWriter C”, *MacUser*, March 1992, p. 63.
2. “HP’s DeskJet 500C: ‘C’ Is for Cost-Effective Color,” *PC Magazine*, January 28, 1992, p. 40.
3. *Hewlett-Packard Journal*, Vol. 39, no. 5, October 1988, pp. 51-98.
4. *Hewlett-Packard Journal*, Vol. 39, no. 4, August 1988, pp. 6-56.
5. *Hewlett-Packard Journal*, Vol. 36, no. 5, May 1985, pp. 4-39.
6. S.J. Bares, “Printing on Plain Paper with a Thermal Inkjet Printer,” *Hewlett-Packard Journal*, Vol. 39, no. 6, December 1988, pp. 39-44.

Ink and Print Cartridge Development for the HP DeskJet 500C/DeskWriter C Printer Family

A new trichamber print cartridge allows the low-cost HP DeskJet printer platform to print in color. The ink vehicle, dyes, dye concentrations, and interactions had to be carefully traded off to optimize performance with respect to color bleed, color saturation, composite black production, edge acuity, drying time, and resistance to crusting.

by Craig Maze, Loren E. Johnson, Daniel A. Kearl, and James P. Shields

Development of the print cartridge for the HP DeskJet 500C and DeskWriter C printers required a combination of ink chemists, print cartridge architects, and design engineers, along with inputs from the fields of color science and product marketing. This article deals with the design and development of the inks and the print cartridge.

The major objective of the printer development project was to provide a low-cost desktop printer that produces laser-quality black print and also offers color capability to the user. To enable the low-cost HP DeskJet printer platform to print in color, a trichamber color print cartridge was designed that can be exchanged for the black print cartridge when color printing is desired.

Incorporating a trichamber color print cartridge into the existing HP DeskJet print platform posed several challenges for ink chemistry and print cartridge design. Like the ink for the black-only HP DeskJet printer family, the color inks had to work on a wide variety of "plain" papers. Plain paper printing using thermal inkjet technology has been reviewed.[1] One important consideration is ink drying time. In addition, when inks of different colors are laid down next to one another on the paper, they must not bleed or diffuse into adjacent regions of different color, a fault commonly referred to as color bleed. A significant level of color bleed is undesirable and causes the border between the two colors to appear ragged and undefined.

Another challenge involved the production of composite black. This term refers to the production of black using the three primary colors (cyan, yellow, and magenta) from the trichamber print cartridge. Composite black results from printing dots of cyan, yellow, and magenta directly on top of each other. This requirement exists because only the black print cartridge or the color print cartridge can reside in the printer stall at any given time. Thus, when the color cartridge is installed, any black print must be produced using composite black. As discussed later, this required some trade-offs between the composite black color characteristics, the color quality of the primary and secondary colors, and ink robustness.

Finally, the greatest challenge was perhaps in the development of an ink that performed well on a wide variety of plain papers. These paper types range from high-quality cotton bond papers to the papers used in high-speed copiers. Finding an ink that met all of the challenges involved delicate trade-offs among objectives that were often in conflict.

Evolution of the Ink Vehicle

Water-based color inks have evolved from formulations designed to print on special paper to those capable of printing on plain paper. Many hundreds of different mixtures were tried before obtaining satisfactory performance over a large variety of papers. In general, inks changed from those containing large amounts of diethylene glycol, about 60%, to those containing different organic solvents at greatly lowered concentrations.

Some of the requirements on the ink are in conflict. For example, drying must be slow in the print cartridge to prevent plugging of orifices and the associated loss of print quality, but rapid on paper to facilitate paper handling and maintain printing speed. Color bleed, if too great, will ruin a print sample (see Fig. 1). It is also one of the most difficult parameters to control by ink formulation alone. Composite black

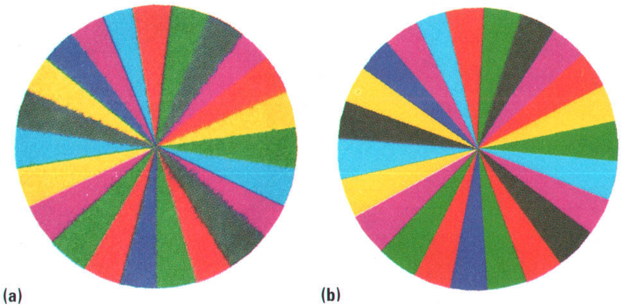


Fig. 1. Color bleed is the undesirable mixing of two colors printed next to each other. (a) Unacceptable color bleed. (b) Minimal color bleed.

text print quality must be high, which means a dark composite black and high edge acuity. Unfortunately, ink formulas that give improved text print quality tend to have increased amounts of color bleed.

It is difficult to meet all of these requirements, and some compromise is unavoidable. Market surveys were used to separate performance factors that are important to our customers from those that matter least. Samples were printed with known defects on different papers over a range of temperature and humidity conditions and with different inks. These were submitted to potential customers for their evaluation and the results were used to guide ink development. For example, high color saturation was a strong customer requirement, and ink component concentrations were adjusted accordingly.

The first major challenge was to provide a formulation that would not bleed and would deliver saturated colors. Bleed was suppressed by adding surface-active reagents to speed penetration of the ink into the paper. This reduces the time available for dyes to diffuse on the paper and for colors to mix. Unfortunately, penetration into the paper reduces color saturation, so allowances were made to keep color saturation high.

High-quality composite black text print requires a dark black with high character edge definition. The former is achieved by keeping the dye on the surface of the paper, and the latter by minimizing penetration of the ink into the paper. These requirements can be at variance with factors that control bleed.

It became apparent that to deal with the various trade-offs in an efficient manner a different experimental approach was needed. After some exploratory work, these trade-offs were dealt with all at once by conducting a series of designed mixture experiments. Optimum component concentrations were obtained that achieved acceptable bleed control, color gamut, edge acuity, and color saturation. In general, solvent concentration was kept low, surface tension was reduced, and viscosity behavior modified.

Dye Selection Criteria

In selecting and using the dyes for the HP DeskJet 500C/DeskWriter C color print cartridge, four primary attributes were considered. Any dye to be considered for use had to address all four of the following concerns.

The first performance parameter is resistance to crusting of the ink (nozzle clogging upon loss of vehicle). This parameter was tested by allowing print cartridges to sit uncapped for various lengths of time and then firing them and looking for lost nozzles.

Several characteristics of a dye determine how resistant it will be to crusting. The purity of the dye plays a very important role. The presence of any diluents, reaction by-products, secondary colors, or unreacted feedstocks can degrade crusting performance. As a consequence, some initial effort was expended in cleaning up the dyes that looked promising so that crusting performance could be reliably evaluated.

The structure of the dye also plays a very important role in the crusting performance of an ink. As a nozzle sits uncapped, exposed to the air, the more volatile components of the ink evaporate. The solubility of the dye in the resultant evaporated ink at the nozzle influences crusting performance. This solubility is determined in part by the structure of the dye and its counterions.* Dyes that exhibited crusting problems because of their structure were eliminated from further consideration.

The counterion used to balance the charge on the dye molecule also effects the crusting performance of the dye. In many instances this counterion can be changed to improve crusting performance. For dyes that showed good performance in other respects, the counterion was altered to maximize crusting performance.

A property of thermal inkjet performance called “kogation” (from the Japanese *kogasu*, to scorch) is the second primary attribute that is important to dye selection. Kogation occurs when the ink, which contains the dye, is exposed to the high temperatures generated on the printhead resistor surface during operation. These high temperatures sometimes break down the dye into insoluble fragments that stick to the resistor surface. Alternatively, the dye itself can be bound to the surface. The buildup of this carbonaceous layer on the resistor reduces heat transfer to the ink. In severe cases the print cartridge will no longer fire because of the insulating character of the kogation.

Dyes are tested for kogation performance by firing inks containing the dyes to be tested from a print cartridge. As the print cartridge is fired the volume of the ejected drops is monitored. The presence of kogation tends to reduce the drop volume over the life of the print cartridge, resulting in degraded print performance.

Structure and counterion play a role in dye kogation. Dyes that exhibited poor kogation characteristics because of their structure were discarded. In some cases, kogation can be improved by altering the counterion attached to the dye.

The third primary attribute used for dye selection is the color characteristics of the dye. Hue and chroma are parameters of utmost importance to a color printer. Dyes were chosen to have hues within a given range so that a balanced palette could be generated. Dyes were selected to have the highest chroma characteristics possible.

Finally, given that the dyes must function in a trichambered print cartridge, where the same nozzle plate is shared by the three different primary colors, dye-to-dye interactions had to be considered. A dye must not react adversely with any other dye used in the ink set. Examples of such reactions are precipitation and hue change.

Color Balancing and Dye Concentration Adjustment

Once a set of dyes was selected using the above criteria, the concentrations were adjusted to give good secondary and primary hues and to achieve maximum saturation. These two goals had to be achieved without sacrificing the robustness of the print cartridge performance.

(continued on page 73)

* Solutions must be electrically neutral. A counterion is an ion that balances out the charge of another ion in solution, that is, its charge is the opposite of that of the ion in question. Thus, for example, a negatively charged dye molecule might have a positively charged sodium ion as its counterion.

Color Science in Three-Color Inkjet Print Cartridge Development

During the development of the three-chamber color print cartridge there was a need to describe and measure the colors that were printed. There are a number of different ways to describe colors. The purpose of this article is to explain how colors are generated with the color print cartridge, what color means, how colors are measured, and how color measurement can be used in print cartridge development.

How the Color Print Cartridge Prints Colors

The print cartridge has three different inks: cyan, yellow, and magenta (CYM). Eight colors can be generated at 300-dpi (dots per inch) resolution by printing different combinations of the primary inks. The eight colors are printed as follows:

- Not printing any ink gives "white," the paper color.
- Printing one of the primaries gives cyan (a greenish blue), yellow, or magenta (a bluish red).
- Printing a combination of two primaries gives the secondary colors, which are red (a mix of yellow and magenta), green (a mix of cyan and yellow), and blue (a mix of cyan and magenta).
- Printing all three primaries gives the tertiary color black.

Note that the various colors are defined by the printing process, so the names given to them here (red, blue, etc.) are not necessarily accurate.

More Colors with Halftoning

To obtain a larger number of colors using the print cartridge, the ratio of the CYM proportions must be changed. This is not possible at 300 dpi because the HP DeskJet 500C is a binary printer. Halftoning algorithms are used to increase the number of printable colors. Each picture element is composed of a number of dots of ink (pixels). In a simple case, each picture element is composed of 4 pixels in a 2-by-2 halftone cell (sometimes called a superpixel). This yields an effective resolution of 150 dpi. Process yellow is now 4 drops of yellow ink in the 4 pixels, 1 drop per pixel. Process red is 4 drops of yellow and 4 drops of magenta, 1 drop of each ink in each pixel. By reducing the number of magenta drops to 3, 2, or 1, three more colors can be produced between yellow and red. This process can be extended to larger halftone cells to increase the number of available colors. There are many different halftone algorithms which yield different colors and textures but all have the ability to increase the number of colors at the expense of resolution. For all these print methods, any color available to the HP DeskJet 500C printer can be produced by specifying how much of the CYM inks to print and where to place the inks. This leads to the concept of a process color space.

Process Color Space

As stated earlier, any color achievable with the color print cartridge can be described in terms of the amounts of CYM being printed for a particular halftone cell. This is an efficient color space for the printer but is not useful for describing the printed color. The problem with process spaces, in general, is that they are system dependent. All printers have the same process color space—the proportions of CYM being printed. Unfortunately, the color printed is not constant, because different printers have different CYM inks. This was apparent in developing the inks for the HP DeskJet 500C print cartridge. Different dyes and vehicles were formulated, all yielding different colors upon printing. A better description of color other than the process space was needed.

Visually Based Color Spaces

Visually based color spaces or color order systems are created by empirically arranging a set of colors according to the observed relationships between adjacent colors. The criteria describing the relationship can be varied, yielding different color systems. Color order systems are all three-dimensional, independent of the sorting criteria. These spaces are very useful because the colors are "uniformly" spaced and based on human perception. A unique value can be placed on any color by visually matching the sample color to an example from a standard color order system.

A widely used color order system is the Munsell notation. Colors are described by:

- Value, a measure of darkness and lightness
- Hue, the shade of the color
- Chroma, the "brilliance" of the color.

This three-dimensional color space has cylindrical coordinates: value is the z axis, chroma is the radial distance, and hue is the angle. It is a satisfying color coordinate system for describing colors because it arose from an empirical sorting of colors into a standard set. The Munsell color space is a uniform color space but has problems in day-to-day use. To use this color order system, a visual comparison between the sample color and the standards has to be made. This is prone to errors and is time-consuming, and the exact standard required must be available.

Measurement-Based Color Systems

The optimal system would be a measurement system that yields values that correlate with a color order system. This has been achieved using the CIE color measuring system.

For a particular observer and illuminant, colors can be uniquely described by three coordinates. Therefore, to measure colors, three values need to be obtained. A one-to-one mapping between the three measurement values and a visually based color space must exist for the measurements to be truly useful.

Three components are necessary for color measurement:

- A light that illuminates the sample
- A sample that has a characteristic reflectance spectrum that changes the relative spectral distribution of the light source
- An observer that detects the reflected light.

Common standard light sources are D65, a representation of daylight, A, representing incandescent light, and F2 or CWF, representing "cool white fluorescent" lights. Standard detectors include the CIE 2-degree and 10-degree observers. At HP we have typically used the D65 source and the 10-degree observer. This source and observer have been selected and standardized to give the CIE system of color measurement.

The CIE system applied to a reflectance spectrum yields X, Y, and Z tristimulus values. These values can be used to specify any color for a given observer and source. Unfortunately, the mapping of the XYZ space onto a visually based color space such as the Munsell space is very nonlinear. Many transformations of the XYZ space were investigated to yield a uniform measurement space. A common transformation is the CIELAB color space. The coordinates of this space, called L* a*, and b*, map quite well into the Munsell space except that a rectangular coordinate system is used to describe the colors rather than a cylindrical coordinate system, as shown in Fig. 1.

Color Measurement

Once the source and the observer are specified, the reflectance spectrum of the sample determines the measured color. In developing the color print cartridge, the reflectance spectrum was considered to be the only variable under the control of the printer. The ink, print cartridge, media, and print modes had to be developed to deliver the appropriate reflectance spectra and therefore colors to meet the market needs.

Color measurement can be done with two types of instruments: either a colorimeter or a color spectrophotometer. A colorimeter is designed to measure the XYZ tristimulus values directly, relative to a specified illuminant and observer. A spectrophotometer measures the reflectance spectrum such that the XYZ tristimulus values for any combination of source and observer can be subsequently calculated. A colorimeter is useful for relative measurements while a spectrophotometer is designed more for absolute measurements. The flexibility of the spectrophotometer comes at an additional cost, as might be expected.

Further Manipulations of CIELAB

CIELAB is a rectangular coordinate system in which L* is the lightness color coordinate, a* is the red-green coordinate, and b* is the yellow-blue coordinate.

It is often useful to monitor the color of a sample using hue and chroma metrics because these yield information similar to the Munsell system. In the CIELAB system, hue (h) can be defined as the arctangent of the a* and b* coordinates, and chroma (C*) is the radial distance from the L* axis as represented in Fig. 1:

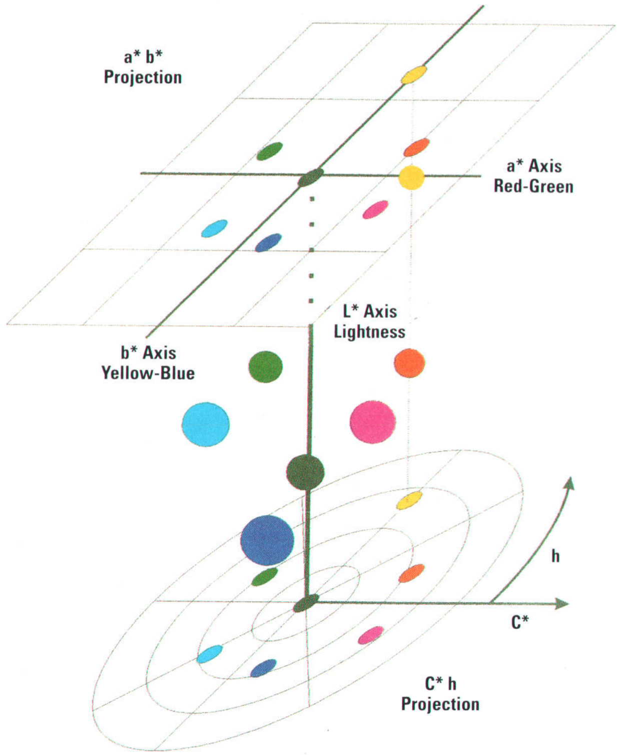


Fig. 1. 3D representation of the CIELAB space showing projections onto the a*b* and C*h planes.

$$h = \arctan(a^*/b^*)$$

$$C^* = \sqrt{a^{*2} + b^{*2}}.$$

Hue and chroma derived from the CIELAB space are not the same as the Munsell hue and chroma.

Another quantity derivable from the CIELAB values is color "saturation," which gives a sense of the depth of color:

$$S^* = C^*/L^*,$$

where S* is saturation and C* and L* are defined above.

The difference between two colors can be easily described using the CIELAB system. Since each color is described by three coordinates, the color difference can be given by the vector length between the two colors, as shown in Fig. 2. A color match is obtained when the vector difference is zero, or the two colors have the same L*, a*, and b* values. The vector difference between the two colors is called ΔE^* and is given by the following equation:

$$\Delta E^* = \sqrt{\Delta L^{*2} + \Delta a^{*2} + \Delta b^{*2}},$$

where ΔL^* is the difference in L* values, Δa^* is the difference in a* values, and Δb^* is the difference in b* values between the two color samples.

Other color difference metrics can be defined, such as ΔC^*, the difference in chroma. Care needs to be taken with ΔE^* because the critical ΔE^* for a color change to be noticeable depends on the color of the sample. Yellows are forgiving while reds are not.

The selection of the appropriate color difference metric depends on the task for which it is being used. ΔE^* is a scalar quantity. It gives the size of the color difference but not the direction. Breaking ΔE^* into its components often yields more useful data because the direction and main component of the color difference can be obtained.

Uses of Color Measurement

Process Control. During the development of the color print cartridge it was important to maintain constant colors. Variations in the color output over time can be caused by many variables. Measuring the color printed with a print cartridge gave some measure of process control. Other methods of process control could have been implemented but colorimetric data is particularly useful because the process limits are related to human perception.

Dye and Ink Selection. Both the dye and the ink vehicle affect the colors generated with the color print cartridge. Given target colors, inks can be formulated to match the targets. Colorimetric data is particularly useful here, especially CIE hue. The hue values of the secondaries are very important and depend strongly on the dye selection. For a given set of dyes, saturation was also found to be a useful metric.

Grey Balancing. Because the HP DeskJet 500C is a one-print cartridge color printer, black must be printed with a mixture of the CYM primaries. The best black was not obtained with a tertiary black but by using a different ratio of C:Y:M. Color measurement was used to optimize the ratio of cyan, yellow, and magenta to print when black was requested. This had two benefits. The first is that the quality of the black print is improved because the best possible black is printed. The second is that less ink is needed to print this black, so the bleed performance of the printer is improved.

WYSIWYG Printing. Just as measurement spaces can be mapped onto visually based spaces, process spaces can also be mapped onto any color space. This makes it possible to print device independent colors and duplicate the desired colors. For example, CRT monitor colors are described by their own process space, RGB. There is a one-to-one mapping, very dependent on the particular monitor, between RGB and CIELAB. There is also a one-to-one mapping between CIELAB and the printer CYM space. By knowing both maps the colors shown on the monitor can be accurately duplicated on the printer, yielding WYSIWYG printing (What You See Is What You Get). The success of this process depends on the characterization of the mapping from the printer CYM space to the CIELAB space. This involves extensive color measurement and characterization of the printing system using a spectrophotometer. Problems do occur with WYSIWYG printing when either the printer or the monitor cannot print or display the requested colors. Sophisticated image manipulation has to occur so that a successful mapping can be accomplished.

John M. Skene
Development Engineer
Inkjet Components Division

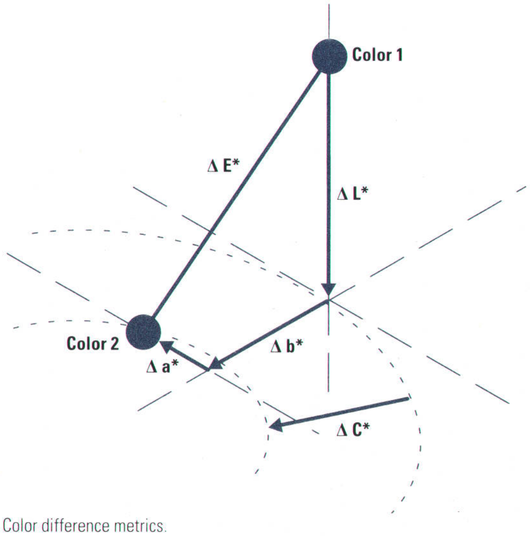


Fig. 2. Color difference metrics.

To guarantee print cartridge performance, the condition of the ink over the life of the print cartridge was determined. Even in an enclosed system such as the HP DeskJet print cartridge, there is vapor loss over time through the plastic walls. This vapor loss alters the ink vehicle composition and can ultimately affect the crusting performance. Once the rate of this loss was determined, inks were made up having vehicle compositions that a print cartridge would have at the end of its rated life. Dyes at various concentrations were placed in these vehicles and the crusting was tested. The maximum dye concentration that could pass the crusting test was established as the temporary maximum for that particular dye.

Next, a set of inks was made having a range of dye concentrations, from low up to the previously established temporary maximum. Print samples were made with these inks and the L*a*b* coordinates of the samples were measured (see "Color Science in Three-Color Inkjet Print Cartridge Development," page 71). If a dye showed a rebound point (the dye concentration at which color performance decreases because of overloading), the rebound point was set as the new temporary maximum. These temporary maximum dye concentrations were then corrected for the effects of vapor loss so that the ink performance would be maintained throughout the projected print cartridge life.

The next step in the process of color balancing was to determine what colors were most important in the color DeskJet printers. The relative order of desirability of various reds, blues, and greens for secondary colors was established by market studies. The ratios of the different dyes needed to produce the optimum red, blue, and green secondaries were then determined by trial and error.

For example, the concentration ratio of magenta to yellow that produces an optimal red might be 1.8:0.7. The optimal ratios for the blue and green secondaries of the dye set would be different. In most cases the ratios didn't match exactly. For instance, you could use the magenta-to-yellow ratio to get a good red. Then you could use that magenta concentration with the optimal ratio of magenta to cyan to get a good blue. However, the resulting cyan and yellow concentrations would then not be optimal for green. Consequently, optimizing the overall color palette required that some trade-offs be made.

In making the trade-offs and arriving at the final dye concentrations for use in the ink, an anchor dye and concentration were chosen. This anchor dye was chosen by evaluating the effect of dye concentration on chroma. The dye whose chroma level was most sensitive to dye concentration was chosen as the anchor dye. The concentration of this dye was determined by using the temporary maximum ratios obtained earlier in conjunction with the optimal ratios needed to achieve good secondaries.

For instance, say that yellow is chosen as the anchor color and that the maximum concentrations for magenta, cyan, and yellow are 5%, 4%, and 4%, respectively. In addition, say that the optimum ratio of magenta to yellow for achieving a good red is 2.5:1 and the optimum ratio of yellow to cyan for achieving a good green is 1.5:1. The new concentration for yellow is then calculated by dividing the maximum concentrations of the added colors, cyan for instance, by the optimal mix ratios. In this example, this yields a yellow concentration of 2.0% in the case of the red secondary and a concentration of 2.7% in the case of the green secondary. The smallest of the two secondary concentrations is then chosen as the maximum concentration of the anchor color, which in this case is 2% yellow for the red secondary.

Once optimal concentrations were found for the other two primaries in relation to the anchor dye concentration, print samples were made using inks with these dye concentrations. Based upon a visual evaluation of the print samples, the concentrations of the two dependent primaries were varied to make various trade-offs in the secondary color space.

For instance, it may be acceptable to alter the hue of the red slightly to achieve a markedly better blue. This final stage of color balancing involves repeated trials and fine tuning of the two dependent primaries within the bounds established by the temporary maximum concentrations. The process of printing samples, evaluating the color space, and then making concentration adjustments was repeated to arrive at the most acceptable color space.

Matching the Ink and Printhead

Thermal inkjet ink, printhead, and printer design are all excellent examples of highly interactive technology development. Development progress cannot be made on an inkjet ink formulation without at the same time having an appropriate printhead and printer in which to test the formulation. Likewise, printhead architecture and printer development efforts are very closely coupled to the performance characteristics of the ink formulation. Progress cannot be made in any one of these three arenas without the aid and support of the other two. This very interactive development environment presents a variety of technological and organizational challenges. In this environment, a carefully constructed ink screening strategy was required for timely printer system development.

The ink screening strategy that was used for the color HP DeskJet 500C/DeskWriter C print system development was based on rapid iterations of ink formulation, print cartridge architecture, and printer test beds. Empirical performance testing was conducted on dozens of ink formulations over a period of a few short months to determine the final ink composition. A progressively wider variety of tests were run on each ink formulation and printhead architecture combination. Once a given combination had passed quick, easy-to-complete initial testing, it was progressively subjected to tests requiring greater time and resource investments. The intended outcome of this effort was a printing system that provided outstanding plain paper color printing performance over a wide range of media and printing environments.

For a given printer resolution, the interaction of ink, print modes, and print media determines the final print quality. As discussed earlier, this interaction is largely controlled by the nature of the ink vehicle. For a given droplet volume, different ink vehicles will produce differing dot sizes on the various types of media. Ideally, one would like absolute control of dot size and placement. Given an attractive ink formulation, variations in the printhead architecture are used to control dot size and placement (see "Thermal Inkjet Review...," page 67).

To produce optimal print quality, the ink and the delivered drop volume need to be carefully matched. If the drop volume is too low, paper coverage will be incomplete. This results in desaturation of colors and irregular area fill. When the drop volume is too high, resolution and edge acuity suffer, and adjacent colors have a greater tendency to bleed into one another. Local printhead architecture largely determines the delivered drop volume. The surface area of the thin-film drive resistor and the exit diameter of the nozzle bore are the most significant factors in determining ejected drop volume. Ink viscosity, surface tension, and the size and shape of the photosensitive barrier film that surrounds the thin-film resistor largely determine the frequency response of the thermal inkjet device.[2] These are the main variables used by the printhead designer to optimize print quality and speed. These variables were tailored for each of the ink formulations tested.

Printhead Performance Tests

Candidate ink formulations were subjected to a wide range of performance tests. This regime consisted of both printing and print cartridge reliability tests. The central theme behind all of this effort was to test the print system's ability to deliver outstanding plain paper color print quality consistently over the life of the printhead.

The printing tests were conducted over a large sample of different media types. An international selection of "plain" papers was included in the sampling. In addition, printing performance was measured on HP LX JetSeries transparency film and HP CX JetSeries special paper. These special media products have been developed by HP to provide maximum thermal inkjet print quality. In addition to printing on a large number of different media, testing was conducted using different settings of the printer driver intensity control (see article, page 93). Environmental conditions for printing were systematically varied to cover the printer operating environment (5°C to 40°C, 10% to 80% relative humidity).

Several specific print quality attributes were examined during the course of the performance tests. These attributes include color bleed, color saturation, composite black text quality, and "waitbanding" performance (defined later).

As discussed earlier, color bleed is defined as the undesirable intermixing of two different colors when printed immediately adjacent to each another (Fig. 1). This intermixing results in an irregular deviation from the intended color interface profile. It is most visible when the two colors are of high relative contrast. Yellow and red or yellow and black are the color pairs most often tested for color bleed. Color bleed was quantified by visual comparisons against a set of standards. These standards were prepared such that varying degrees of bleed were produced. A bleed index number was then assigned to each of the standards. In addition to being a strong function of ink formulation and delivered drop volume, color bleed is also closely coupled to the dot density, media, and printing environment. It is aggravated by high temperature and high humidity. A significant portion of the color bleed testing was completed at temperatures of 35°C to 40°C at 80% relative humidity.

Color saturation was quantified using either spectrophotometers or colorimeters (see "Color Science in Three-Color Inkjet Print Cartridge Development," page 71). It is defined as:

$$S^* = C^*/L^* = \sqrt{a^{*2} + b^{*2}}\,/L^*$$

where: C^* = measured color chroma
a^* = red-green color coordinate
b^* = yellow-blue color coordinate
L^* = lightness color coordinate.

The major factors affecting color saturation include colorant concentration, vehicle formulation, drop volume, print mode, media, and printing environment. A cold, dry printing environment was found to reduce color saturation. Testing for color saturation was completed at 5°C to 10°C and 10% to 15% relative humidity.

The creation of high-quality black text characters and area fills using composite black is a particularly demanding task. As was the case for color bleed, internally generated comparative standards were used to quantify composite black performance. Ink vehicle formulation, color balance, and printing media are the factors most strongly influencing composite black text and area fill quality.

Waitbanding is a rather interesting printing artifact associated with the nonsimultaneous raster printing of colors. The nozzle arrangement in the color printhead is such that the different colors are printed serially onto the print media (Fig. 2). Cyan is the first ink to be delivered to the page, followed by magenta, then yellow. Nozzles for the three colors are spatially separated on the printhead by four dot rows (4/300 inch) in the paper advance direction. This results in a time separation of successive color ink delivery on the order

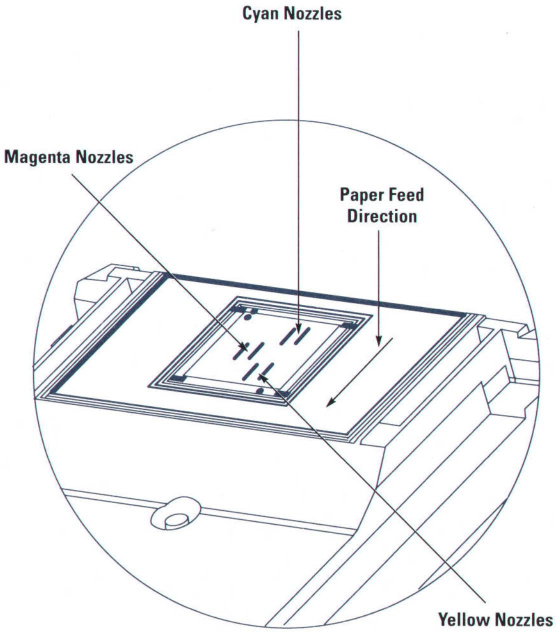


Fig. 2. The nozzle arrangement in the color printhead is designed to deliver the three colors serially to the media, cyan being the first. Sixteen nozzles are provided for each of the three colors.

of 1 to 2 seconds, depending on print mode and scan width. During the time interval between delivery of one color and delivery of the next, there is a significant amount of media penetration and evaporation of the ink vehicle. When mixed, or secondary colors (red, green, blue, and composite black) are formed, two or more of the primary colors are delivered to a given local area of the print media.

Waitbanding can occur if the time interval between successive raster scans during printing is significantly varied when creating secondary colors. When this time interval is varied, differing amounts of ink penetration and evaporation take place before the next primary color is printed onto the first primary color. The result can be the creation of a horizontal band of varied lightness and chroma. This variation in color appearance was quantified using the following color difference relationship:

$$\Delta E = \sqrt{\Delta L^{*2} + \Delta a^{*2} + \Delta b^{*2}},$$

where ΔL^*, Δa^*, and Δb^* are the differences in the color coordinates in the region of the waitband relative to an adjacent region outside the waitband.

The control of the time interval between scans involves a rather complex interplay between the user's computer hardware, the application software, the operating system, the communication network to the printer (if used), and the printer processor and firmware. Several steps have been taken in the development of the color HP DeskJet 500C/ DeskWriter C system to minimize the variability in the time interval between successive scans and its impact on waitbanding. Ink formulation also can play a major role in minimizing the magnitude of the waitband color difference. It is for this reason that waitbanding testing was included as one of the screening parameters in the ink formulation selection regime.

Several print cartridge performance attributes not directly associated with a print sample were measured as well. In addition to safety and regulatory considerations (see page 76), the selection of ink components is also often limited by compatibility with the other materials in the printhead and printer system. These system-level limitations are typically determined using elevated temperature materials compatibility and printhead tests. In their simplest form, these tests consist of immersing a given printhead or printer component in the ink or ink constituent of interest. The immersed component is then subjected to elevated temperatures for a period of several weeks. Degradation of the component and changes in the ink or ink constituent are then measured at the end of this period. Arrhenius modeling[3] of degradation reaction rates is often used to assess the impact on print system longevity and reliability.

Reducing the tendency for the very small printhead nozzles to clog or crust over presents an ongoing challenge for the ink chemist and product designers.[4] Ink formulation testing for the time to the first misdirected droplet was conducted throughout the ink development effort. The major printhead design factors affecting crusting behavior include colorant species and concentration, vehicle solvents, and the environment in which the printhead is being operated. The printer designers can contribute to the prevention of clogging by providing special print cartridge servicing functions. These functions include tight capping of the print cartridge's orifice plate when not in use, plus periodic wiping and firing of the nozzles before and during printing.

An additional area of ink formulation screening is kogation testing. As discussed earlier, kogation is defined as the plaque-like buildup found on the thin-film resistor surfaces of thermal inkjet devices after many firings of the printhead.[4,5] Kogation testing consists of monitoring the drop volume or color saturation a given printhead produces over the full lifetime of the head. Careful selection of ink constituents and control over the thin-film resistor drive energy are needed to reduce or eliminate kogation problems.

Ink Quality Control

The inks for the HP DeskJet 500C and DeskWriter C printers were carefully formulated to provide outstanding performance on a wide variety of plain papers. Careful control of ink component concentrations and purities during ink production is critical to ensure that customers receive consistently good performance from their print cartridges.

The consequences of ink impurities or incorrect ink component concentrations can be quite severe. For instance, incorrect dye loads and impurities in the dyes can cause significant shifts in the hues that are produced by the printer. Incorrect pH values can cause chemical instabilities in the ink as well as undesirable interactions with some of the print cartridge materials. Incorrect ink viscosity can drastically alter the firing characteristics of the print cartridge, resulting in poorly formed droplets and poor print quality. Improper concentrations of certain vehicle components can lead to a loss of bleed control between different colors. Impurities in the dyes or other vehicle components can cause nozzle plugging and reduce bleed control and print quality.

The physical ink parameters that are monitored include: static surface tension, viscosity, pH, solution absorbance, and conductivity. A Hewlett-Packard 5890A gas chromatograph (GC) with an HP 9000 Series 300 workstation is used to determine the concentrations of some of the components in the ink. In addition, the GC is used to verify the purity of one of the ink constituents before it is added to the ink. Dyes are purified using a variety of procedures including reverse osmosis. Selected ion levels are monitored using atomic absorption spectrometry.

Not all of the components in the ink are determined by a specific analytical method. Certain practical compromises were necessary. The concentration of some components is very difficult to determine in the ink matrix or requires very specialized and expensive analytical equipment that is not cost-effective in a production environment. For these components, indirect methods are used to verify proper concentrations, along with the very careful use of process control procedures such as weight logs.

Delivery to the customer of consistent, high-quality output relies on many factors, one of which is a consistent ink composition. By carefully controlling ink component concentrations and purities, delivery of this consistent performance to the customer is ensured.

Making HP Print Cartridges Safe for Consumers Around the World

The ability to provide assurance that our products do not cause harm to consumers and the environment is central to Hewlett-Packard's thermal inkjet product stewardship. Ink safety evaluation is the most important part of the overall safety and regulatory process.

Thermal inkjet print cartridges contain liquid inks designed to deliver laser-printer-like print quality. These liquid inks contain dyes, solvents, and special additives, which are essential to the operation of the print cartridges. Chemists spend considerable time developing these inks and often synthesize new chemicals to improve print quality.

To ensure the safety and marketability of our print cartridges worldwide, all new ink and print cartridge ingredients are screened for potential safety (toxicology) and regulatory concerns during the research and development phase.

To begin with, all ingredients are screened to eliminate any ingredients that may elicit harmful toxic responses in humans and the environment. The scientific literature is searched to determine what is known about the ingredients. Structural activity relationships are used to estimate the toxicity of newly synthesized chemicals. In addition, a tier approach is used to determine the scientific tests to be conducted. The tier system consists of both Hewlett-Packard's safety standards and government regulatory requirements.

Regulatory concerns for newly synthesized chemicals used in ink and print cartridge development are more complex. Since we sell our print cartridges worldwide, we must obey all laws governing chemicals contained within an article—in this case the print cartridge. These laws differ by country and we must understand and develop inks accordingly. For example, in the United States our print cartridges are regulated by the Environmental Protection Agency (EPA) and the Consumer Product Safety Commission (CPSC).

To sell print cartridges worldwide, we must obey many rules and regulations. For example, in the U.S.A.,we must file a Premanufacture Notification (PMN) with the EPA before we sell a print cartridge containing a new chemical not listed on the United States master inventory. The cost for filing a PMN is about $2,500 per chemical and there is a 90-to-180-day waiting period for approval. There are substantial penalties for noncompliance.

In Europe our print cartridge inks are regulated by the European Economic Community. The EEC requires a series of scientific tests before we can market a print cartridge containing a new ink. These tests can cost from $50,000 to $150,000 per compound and take up to one year to conduct. In addition, the Europeans have a 45-day waiting period before allowing the sale of any new ingredients.

Japan's laws (regulated by the Japanese Ministry) are slightly more favorable for our print cartridges because they only require the registration of certain chemicals. Since none of our inkjet chemicals are listed on Japan's list of "hazardous chemicals" and our print cartridges fall under an article exemption law, we do not have to register our newly synthesized ingredients. However, the ingredients must not impose any health risk to humans and the environment.

Obviously, there are many more countries, laws, and regulations we must obey to sell our print cartridges worldwide. Our print cartridge and ink ingredients must be safe and free of any legal restraints. By addressing safety and regulatory issues during the research and development stage, Hewlett-Packard intends to continue to meet the needs of its many markets worldwide.

Acknowledgments

Special thanks to the Business and Technical Information Center Personnel and Susan Dyer-Preston for their invaluable assistance.

Michael L. Holcomb
Toxicologist and Regulatory Engineer
Inkjet Components Division

Acknowledgments

The efforts of many people went into bringing the DeskJet 500C/DeskWriter C color ink and print cartridge to market in a successful and timely manner. The authors wish to thank the members of our ink technician staff and pen test groups for their high-quality work throughout the formulation of many inks and the testing of many inks and print cartridges. We also wish to thank Lynn Gibson, Joe Torgerson, Lew Barton, Jim Kearns, Greg Blythe, Terry Lambright, Allen Brown, Patrick Boyd, and Peter Burke of the architecture team and Norm Pawlowski, Michael Holcomb, Ken Parton, John Skene, and John Moffatt of the ink team for the roles that they played in helping to bring this product to market. Finally, we would like to thank Ron Prevost, John Coyier, Melissa Boyd, and Skip Rung for successfully managing a very diverse group of individuals.

References

1. S.J. Bares, "Printing on Plain Paper with a Thermal Inkjet Printer," *Hewlett-Packard Journal*, Vol. 39, no. 6, December 1988, pp. 39-44.
2. W.A. Buskirk, et al, "Development of a High-Resolution Thermal Inkjet Printhead," *Hewlett-Packard Journal*, Vol. 39, no. 5, October 1988, pp. 57-58.
3. W. Nelson, *Accelerated Testing—Statistical Models, Test Plans, and Data Analyses*, John Wiley and Sons, 1990, pp. 75-85
4. D.J. Palmer, et al, "Ink and Media Development for the HP PaintJet Printer," *Hewlett-Packard Journal*, Vol. 39, no. 4, August 1988, pp. 45-50.
5. D. Hackleman, "Where the Ink Hits the Paper...," *Hewlett-Packard Journal*, Vol. 36, no. 5, May 1985, p. 32.

Automated Assembly of the HP DeskJet 500C/DeskWriter C Color Print Cartridge

Roughly 60% of the assembly technology had to be developed especially for the color print cartridge. Plastic welding, adhesive dispensing, TAB circuit staking, and ink fill were among the challenges.

by Lee S. Mason and Mark C. Huth

One of the key considerations in the development of the color print cartridge for the HP DeskJet 500C and DeskWriter C printers was time to market. Because of this, every effort was made to leverage the design and assembly technology of the existing black print cartridge where possible, and to improve on the design where the opportunity presented itself.

Fig. 1 shows an exploded view of the color print cartridge. Fig. 2 shows the assembly line for this cartridge. Many of the stations were leveraged to some degree from the black print cartridge assembly line. The chart below lists the stations that were heavily leveraged (less than 20% of the station redesigned), moderately leveraged (20 to 70% of the station redesigned), and not leveraged (essentially a new design).

Heavily Leveraged	Moderately Leveraged	New Design
Bar-Code Label	0-1 Transfer	Nose Load
Head Attach	Date Code	Body Load
Head Cure	Print Test	Nose Weld
TAB Prep	Offload and Tape	Filter Stake
TAB Attach		Filter Stake Inspect
TAB Bond		Fail Offload
TAB Wrap		Adhesive Dispense
UV Cure		Adhesive Inspect
E-Test		Cheek Stake
1-2 Transfer		Side Heatstake
Pen Seat		Encapsulant Dispense
Clean and Dry		Foam Feed
Print Test Transfer		Foam Stuff
		Lid Load
		Lid Weld
		Ink Fill
		Cap Load
		Cap Weld

Areas of Improvement

The manufacturing development team identified six areas where there was significant opportunity to improve a process from the way it was performed for the black print cartridge assembly. These areas are:

- Cartridge Body Preassembly. A new subsection was added to the assembly line for preassembly of the two-part cartridge body and the filters.
- Bulk Part Feeding. Plastic parts, filters, and foam blocks are automatically fed into the assembly tooling from bulk part hoppers or bowls. These parts are delivered to the production line in bags and loaded into the hoppers as needed.
- Adhesive Dispense. Positioning and bead width control required substantial improvement for the color print cartridge.
- TAB Circuit Staking. To simplify the process for attaching the flexible TAB (tape automated bonding) circuit to the cartridge body, a direct staking process is used, eliminating the need for adhesive between the two parts.
- Ink Fill. The ink fill process was redesigned to allow more control of the process parameters.
- Throughput and Utilization. To make the assembly line more efficient, the fastest conveyor speed available is used, and the control of pallets through each assembly tool is handled by the process controller resident in the tool itself rather than a central line controller. To improve line utilization, space for part buffering between assembly processes was designed in, allowing parts to accumulate during short periods of tool downtime.

Production Line Layout

The basis for the color print cartridge production line is a “power and free” style conveyor system. In this system, two narrow, parallel conveyor belts run continuously. Square pallets rest freely on the belts, which move the pallets between assembly tools. These pallets contain fixturing that holds the print cartridge parts during assembly. All assembly and inspection operations are completely automated with the exception of prebond inspection. Operator responsibilities include material loading, finished product unloading, and tool and process monitoring.

The color print cartridge production line has three subsections referred to as Assembly 0, Assembly 1, and Assembly 2 (see Fig. 2). Each section uses its own unique pallet to fixture the print cartridge parts slightly differently, as required for that section’s sequence of assembly.

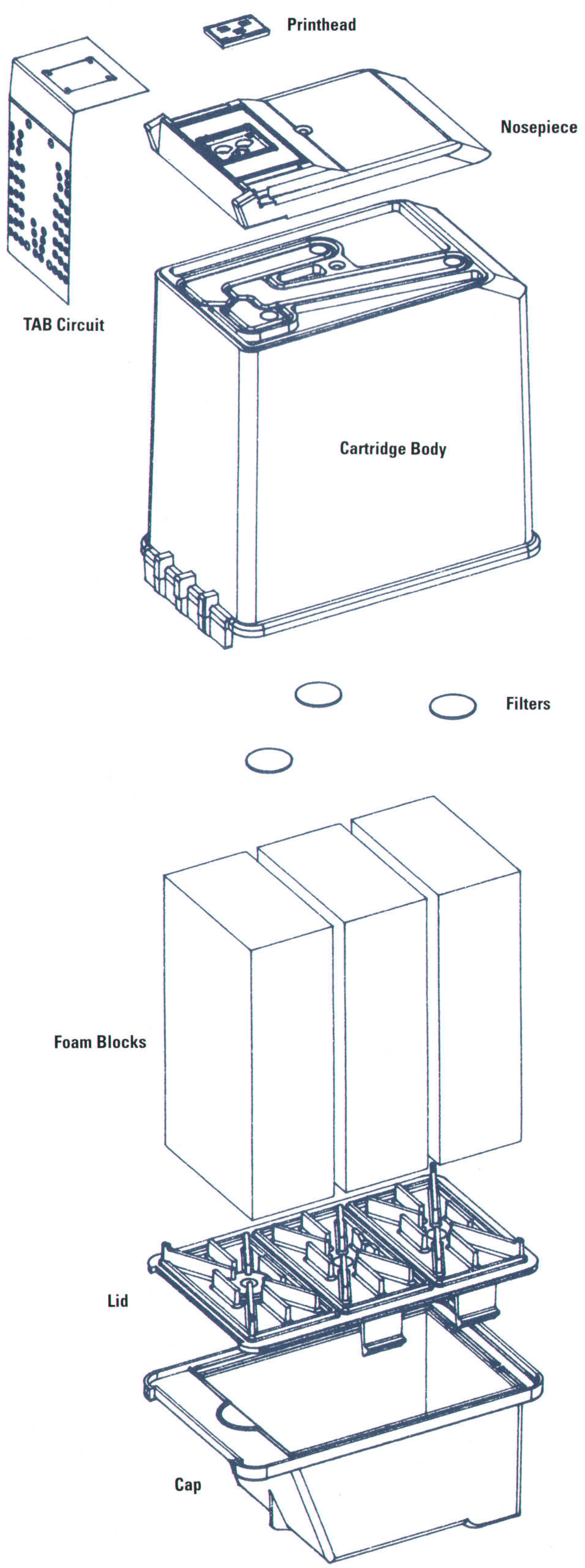


Fig. 1. Exploded view of the color print cartridge.

Assembly 0

This section of the line has the tools necessary for print cartridge body preassembly. It consists of two parallel and adjacent conveyor sections approximately 35 feet long. One conveyor moves Assembly 0 pallets between tools and the adjacent conveyor carries pallets with completed plastic body preassemblies back to the start of the next subsection of the assembly line.

The processes that occur during plastic body preassembly are nose load, body load, nose weld, filter stake, filter stake inspect, and fail offload.

At the nose load station, a plastic nosepiece is loaded into an empty Assembly 0 pallet. Vibratory bowl and inline feeders move the parts to a pick position, where a cam-actuated pick-and-place device grasps a nosepiece and places it into the pallet. A bulk hopper and elevator system is used to regulate the part level in the vibratory bowl.

The body load station loads the plastic body part into the Assembly 0 pallet on top of the nosepiece. It is virtually identical to the nosepiece load station except for the size and shape of the part being fed.

The nose weld station ultrasonically joins the plastic body and nosepiece.

At the filter stake station, a stainless steel mesh filter is heatstaked to the plastic body at the inlet to each of the three ink channels. Three parallel filter feeding and staking processes are used to accomplish this. In each process, the filter is first fed using a vibratory bowl to a pick position where it is picked using vacuum and presented to a heatstaking head by a pneumatically actuated pick-and-place device. The heatstake head then moves down into the plastic body and thermally joins the filter to the inlet of the ink channel.

After the three filters are staked into the plastic body, machine vision is used to inspect the quality of the process automatically. The filters are inspected for presence and location, and the completeness of the weld is determined. For a more detailed description see the article "Machine Vision in Color Print Cartridge Production," page 87.

The last process in the Assembly 0 subsection is the fail offload station. Here any plastic body preassemblies that have failed filter inspection are ejected from the Assembly 0 pallet into a scrap container.

Assembly 1

Assembly 1 consists of the operations that are performed with the nozzle (or printhead) side of the pen facing up. This subsection is configured in a large rectangle about 50 feet long and about 10 feet wide.

The stations on the Assembly 1 portion of the line are 0-1 transfer, bar-code label, adhesive dispense, adhesive inspect, head attach, head cure, date code, TAB prep, TAB attach, cheek stake, TAB bond, TAB wrap, side heatstake, encapsulant dispense, UV cure, and E-test.

The 0-1 transfer station uses a pneumatically actuated robot to remove the pen bodies from the Assembly 0 pallet, rotate them 180 degrees and place them onto the Assembly 1 pallet. The Assembly 1 pallet uses a set of two metal locating mandrels which fit inside the cavity of the pen body assembly to locate two pen bodies precisely on each pallet. Assembly 1 is the only subsection of the line that has two pens on a single pallet. This lowers the cycle time. The cartidge orientation necessary for the Assembly 0 and Assembly 2 operations prevented having more than one cartridge on a pallet.

The labeling machine applies a bar-code label to each pen body assembly. The bar-code label is used in downstream processes.

The adhesive dispense station uses x, y, and z servo-driven positioning tables and a metering pump to dispense the adhesive that secures the printhead to the plastic pen body assembly. Adhesive location and thickness are critical to proper adhesion of the printhead with no leaks between chambers. (Interchamber leaks cause contamination of the ink supply and will give unexpected colors.) For a more detailed description of the adhesive dispense operation, see the article on page 84.

The adhesive inspect station uses a machine vision system to verify that the adhesive pattern is complete and free of voids. For a more detailed description of this operation, see the article entitled "Machine Vision in Color Print Cartridge Production," page 87.

The head attach machine consists of several high-precision servo-driven positioning tables, a machine vision system, and a system controller. The machine removes the printhead die from a film frame, and using machine vision, aligns the pen body assembly with the die. The die is then placed into the adhesive previously dispensed by the adhesive dispense station. This machine must locate the printhead with respect to the pen body assembly within extremely tight tolerances to ensure proper operation of the print cartridge. For a more detailed explanation of this piece of equipment, see reference 1.

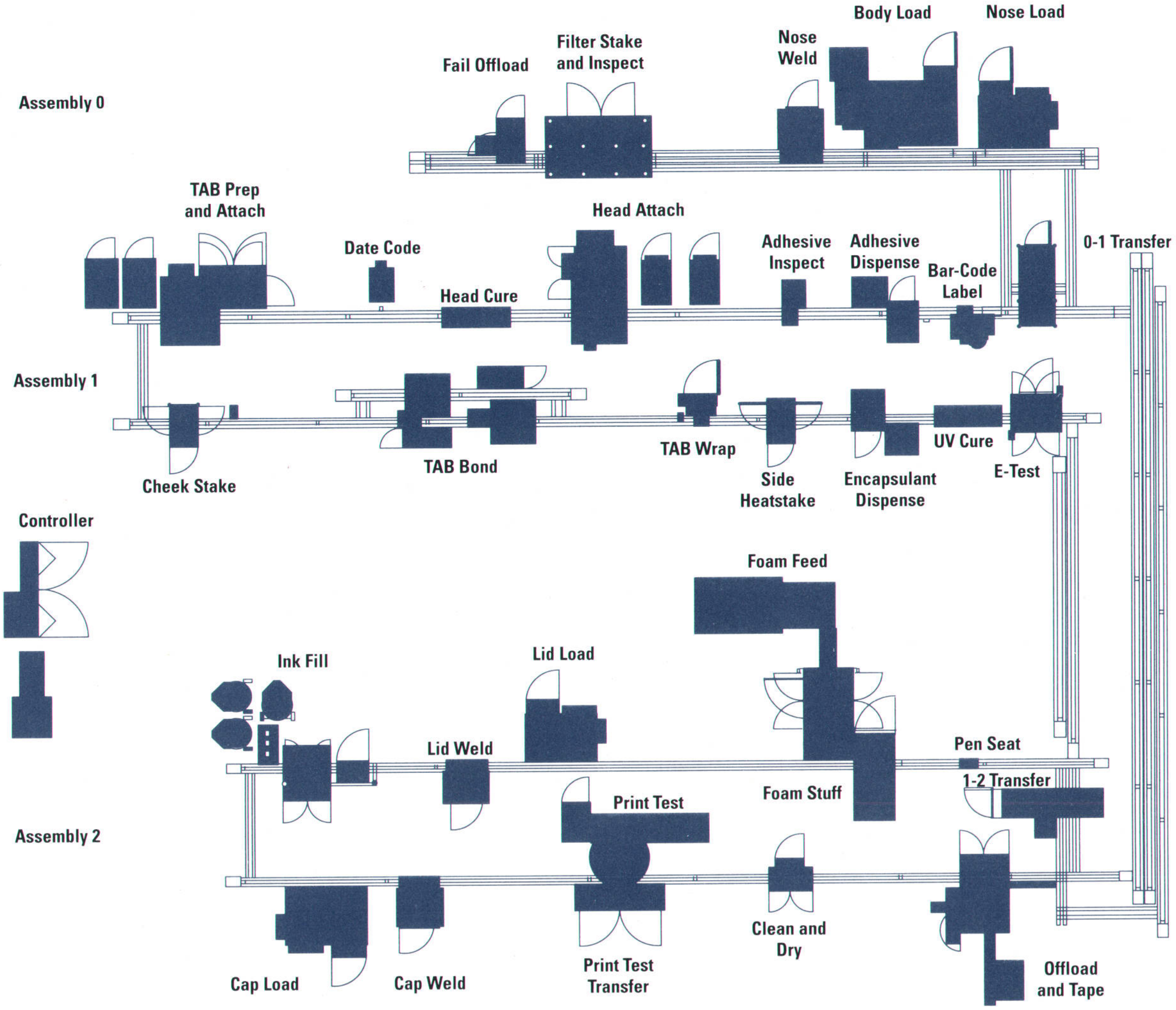


Fig. 2. Color print cartridge final assembly line.

The head cure station thermally cures the adhesive that attaches the printhead to the plastic pen body assembly. The head cure station consists of several identical stations in series, and the adhesive is partially cured at each station. Acceleration and deceleration of the pallets must be carefully controlled in the head cure stations to prevent printhead shift while the adhesive is in its uncured state.

The date code station uses an inkjet printer to imprint a date code onto the plastic pen body assembly. The date code is used within the Inkjet Components Division to track date of manufacture, thin-film printhead lots, yields, and other information.

The TAB prep machine unrolls the raw TAB stock from a reel, uses machine vision to inspect the part, and punches and presents good parts to the TAB alignment machine.

The TAB attach machine uses machine vision and servo-driven positioning tables to align the pen body assembly precisely with the TAB circuit. The TAB circuit provides the electrical interconnect between the pen's printhead and the printer. Once the alignment has been performed, the TAB circuit is tacked into place using a heated probe.

The cheek stake tool uses a heated platen to melt the plastic nose beneath the TAB circuit to attach the cheeks of the TAB circuit permanently to the pen body assembly. The platen is compliant, which is necessary to ensure that the entire surface of the TAB circuit adheres to the pen body assembly.

The TAB bond machines ultrasonically weld the "beams" of the TAB circuit to the interconnect. The station uses machine vision to align the ultrasonic probe with the TAB circuit.

The TAB wrap tool engages the tooling holes in the TAB circuit, folds the TAB circuit over 90 degrees, and holds it in position while it is tacked into place using a heated probe.

The side heatstake tool is similar to the cheek stake tool described previously. It uses a heated platen to melt the plastic pen body beneath the TAB circuit to attach the side of the TAB circuit permanently to the pen body assembly. The platen is compliant, which is necessary to ensure that the entire surface of the TAB circuit adheres to the pen body assembly.

The encapsulant dispense station is very similar to the adhesive dispense station. It uses x, y, and z servo-driven positioning tables and a metering pump to dispense the adhesive that protects the exposed beams of the TAB circuit. Reflective laser sensors are used to locate each pen body assembly precisely before dispensing the encapsulant.

The UV cure station cures the encapsulant adhesive with ultraviolet light. After cure, the bonds are stressed to break any TAB bonds that might otherwise fail in the field.

The E-test is the final station in the Assembly 1 subsection. Using a spring probe block, it electrically verifies all of the interconnections between the printhead and the TAB circuit.

Assembly 2

The last subsection of the assembly line consists of assembly and test operations performed with the printhead facing down and the ink containment cavities facing up. Assembly 2 consists of a conveyor loop approximately 45 feet long and 10 feet wide. One cartridge is held on each pallet. The operations that occur in Assembly 2 are 1-2 transfer, pen seat, foam feed, foam stuff, lid load, lid weld, ink fill, cap load, cap weld, print test transfer, print test, clean and dry, and offload and tape.

The 1-2 transfer station uses a pneumatically actuated robot to remove print cartridge assemblies from the Assembly 1 pallet, rotate them 180 degrees and place them into the Assembly 2 pallet.

The pen seat station uses a pneumatic cylinder to ensure that the print cartridges are completely seated within the pallet. This prevents damage to the downstream tools.

The foam feed system consists of a bulk hopper, a cleated belt elevator, and vibratory feeders, which deliver foam blocks to the three inlet positions of the foam stuff machine. The foam blocks are dropped into the compressor section of the tool, which compresses the foam in two axes. A ram then pushes the foam blocks up into three sheet-metal tubes with rectangular cross sections. The tube set is then moved down into the pen body and a ram drives the foam out of the tubes and into the pen body. The motions of the tubes and the ram during the stuffing sequence are precisely controlled by motor-driven cams.

At the lid load station, the plastic lid is loaded onto the top of the cartridge to cover the foam blocks. Vibratory bowl and inline feeders move the parts to a pick position where a cam-actuated pick-and-place device grasps the lid and places it onto the cartridge. A bulk hopper and elevator system is used to regulate the part level in the vibratory bowl.

The lid weld station ultrasonically joins the plastic lid and body.

The ink fill process is initiated by forming a chamber around the print cartridge and evacuating the chamber to a high vacuum. Ink is pumped into the cartridge using hollow needles, which pass through the three holes in the plastic lid.

The cap load station places the plastic cap onto the top of the print cartridge assembly. The purpose of the cap is to provide a handling surface for the user and to minimize water vapor loss from the cartridge over its life. This station is similar to the lid load station and the other plastic part loading stations.

The cap weld station ultrasonically joins the plastic cap and body. It is the last assembly process on the line.

The print test system is used to evaluate the print quality of the cartridge. Each cartridge is transferred to a four-position rotary table to allow multiple operations to occur in parallel. At one position on the rotary table, electrical contact is made with the print cartridge and a test pattern is fired while a strip of paper is passed beneath the cartridge. The test pattern is then moved beneath cameras and a vision system is used to determine if all nozzles are firing correctly. For a more detailed description of print quality testing see the article on page 87.

The cartridge clean and dry station uses a water loop, which contacts the printhead surface and cleans off the ink residue left behind during print quality testing. After cleaning with water, the printhead is dried with heated air.

The last operations on the production line occur at the offload and tape tool. Here the cartridges are first sorted based on the results of the print quality testing. Cartridges that have failed the print test are offloaded from the pallet onto a conveyor for removal by an operator. Cartridges that have passed print testing are offloaded from the pallet into a tool that tapes the printhead area to contain the ink during shipping and storage. These cartridges are then offloaded onto a separate output conveyor for removal by an operator.

Continuous Development

Many challenges were encountered during the startup phase of the high-volume color print cartridge production line. Initial yields were very low on the high-volume line, roughly 50% of what was seen on the prototype line. There were two main reasons for this discrepancy in yields. First, prototype yields were misleading because relatively few pens were built on the prototype line. Second, in an effort to get as many good test pens as possible, pens were "nursed" through the prototype manufacturing process. Unfortunately, most of these special steps were not readily transferable to the high-volume production line.

The low yields at initial line startup triggered an intensive series of experiments to determine how yield could be improved. Based on the results of these experiments, several new processes were developed and new stations were designed and implemented on the assembly line.

Many of these new processes are proprietary. One that is not is the addition of pallet decelerators in numerous places around the line. The production line uses very small, lightweight pallets, and to keep cycle times to a minimum, the line uses the highest-speed Bosch conveyor. Light pallets and fast belts combine to transfer large shock loads to the parts and may shift the positions of any loose parts. The pallet decelerators reduce these shocks and allow us to benefit from lighter pallets and faster belts without compromising pen quality.

Impact of Leverage

As mentioned earlier, some portions of the color print cartridge design were heavily leveraged from the previously introduced black print cartridge. Our team learned that when it comes to leverage, more is better. The portions of the assembly technology that were not leveraged from the black cartridge (roughly 60% were not) provided the most challenges and headaches.

For example, the development of a two-piece nose and body that can be ultrasonically welded together to form the three separate ink channels of the printhead ink path was extremely challenging (see "Color Inkjet Print Cartridge Ink Manifold Design," page 82). The plastic material was difficult to weld. So much energy was required to produce the weld that the weld horns would stress crack after a few hundred parts. Fabrication tolerances of the two molded plastic parts were very tight and critical to achieving leak-free welds.

The process tolerances for this product required a total redesign of the adhesive dispense equipment. The equipment was custom-designed in-house (as was most of the color print cartridge assembly equipment) because no commercial dispense systems could meet the critical locational tolerances. The color print cartridge also requires different types of adhesives because the color inks are not compatible with the adhesives used on the black print cartridge.

This print cartridge was the first to use heatstaking to adhere the TAB circuit to the plastic body. (The black print cartridge uses patches of die-cut hot melt adhesive to bond the TAB circuit to the body.) The compliant design of the cheek and side heatstakers allows the entire surface area of the TAB circuit to adhere to the body, even without the compliance provided by the soft hot melt patches.

Because of the three different-colored inks and three foam blocks, a special approach was necessary in the foam stuff process. The foam stuff machine was totally redesigned to accomplish the compression and simultaneous insertion of the three small foams.

The formulation of the colored inks presented numerous challenges to the ink fill process. During ink fill, the print cartridge is filled with ink and the ink front moves through the ink channels to the printhead. The print cartridge is first placed in a chamber and the air is evacuated. The vacuum level must be precisely controlled during the entire operation or one of several failure modes may be induced. The ink may froth as it is pulled past the print cartridge's filters. This produces a print cartridge that is prone to "drooling." (Drooling is ink dripping out of the printhead without the print cartridge's being fired.) The print cartridge may contain air bubbles and fail the print quality test. The ink must be removed from the surface of the printhead quickly as the ink front moves past the nozzles. If it is not, the inks will mix on the surface of the printhead and contaminate the ink supply. Dozens of enhancements were made after the ink fill station was online as we gained a better understanding of the process.

The last major deviation from the process for the black print cartridge involves the nozzle taping operation. The tape requirements for the color print cartridge are much more restrictive than for the black print cartridge. First, a tape leak in the color print cartridge will result in ink mixing, which is a functional failure. In the black print cartridge a tape leak is only a cosmetic concern. Second, the primitives (groups of ink nozzles) in the color print cartridge are very close together and close to the edge of the printhead. These factors make tape leaks much more probable with the color print cartridge. Proprietary process changes in the manufacture and application of the tape allowed us to solve these taping problems.

Color Inkjet Print Cartridge Ink Manifold Design

In a disposable printhead, the ink reservoir, the ink plumbing, and the ink firing device must reside within the print cartridge. One of the challenging aspects of the print cartridge design for the HP DeskJet 500C and DeskWriter C printers was the creation of the manifold required to deliver ink from the foam ink reservoir to the printhead. This article deals with the parts and processes used to create this manifold.

Ultrasonic welding is used to join the two parts that form the manifold. This process had been used on the previously released black inkjet pen to attach the plastic cap to the plastic body. Building on some of the prior welding experience, a joint was designed.

Joint Design

The joint is a double shear joint, mainly because the literature strongly suggested that shear joints are the best joint design for obtaining a hermetic seal. In designing an ultrasonic weld joint, three basic requirements must be met:

- Small initial contact area between the two parts. This concentrates the energy to allow the material to melt faster, reducing weld time.
- Uniform and intimate contact. The entire weld joint should be in uniform contact with the mating part. This allows uniform energy distribution to the entire weld joint.
- A means of alignment. If the joint does not provide a means for aligning the parts, some other means (e.g., pin and sockets) must be used.

Fig. 1 shows the weld joint cross section geometry. The small initial contact area is obtained by having the sharp corners of the male piece (nosepiece) interfere with the chamfer in the smaller opening of the female piece (print cartridge body).

Tight molding tolerances are required on both parts to maintain uniform and intimate contact. Fig. 1 also shows how the manifolds are formed by the weld joint. This weld line is not a simple straight line but a rather complicated irregular shape. Obtaining the tight design tolerances challenged the technology of multiple-cavity mold generation and plastic injection molding. All plastic parts were modeled as three-dimensional solids using the Hewlett-Packard ME 30 system. The 3D bodies were used to generate the tool paths for CNC-machining the electrodes used to make the molds.

Part alignment is obtained by the fit of the male piece into the chamfer on the female portion, which accurately locates the two parts with respect to each other.

Equipment

The welder consists of a power supply and an actuator. The power supply converts line voltage, 220V at 60 Hz, to 1,000V at 20 kHz, and also contains the controls for setting weld time or weld energy and trigger force. The 20-kHz electrical signal is converted to mechanical vibration by a piezoelectric transducer called the converter. The actuator holds the converter, booster, and horn. It also provides the controls for down speed and ram air pressure, and houses the encoder for determining horn location. The actuator is mounted on a rigid stand. While parts are being welded, they are held by a nest attached to the production line.

Weld Theory

Ultrasonic welding works by causing friction between two interfering plastic parts. This friction is present on both the microscopic and macroscopic levels. The heat generated by the friction melts the plastic at the weld joint. While the plastic is molten, further vibration causes polymer chains of the mating parts to intermingle. As the plastic cools, the two parts become one. When an ultrasonic weld is done correctly, the physical strength of the joint approaches that of the parent material. There are many benefits to using this method for joining plastics, not limited to the following list:

- It does not require the use of solvents or adhesives (no need for added ventilation).
- Part count is reduced (no need for gaskets, mechanical fasteners, etc.).
- It is easy to integrate into an automated assembly process (microprocessor control, fast cycle time).
- The highly localized heating results in little or no part deformation.

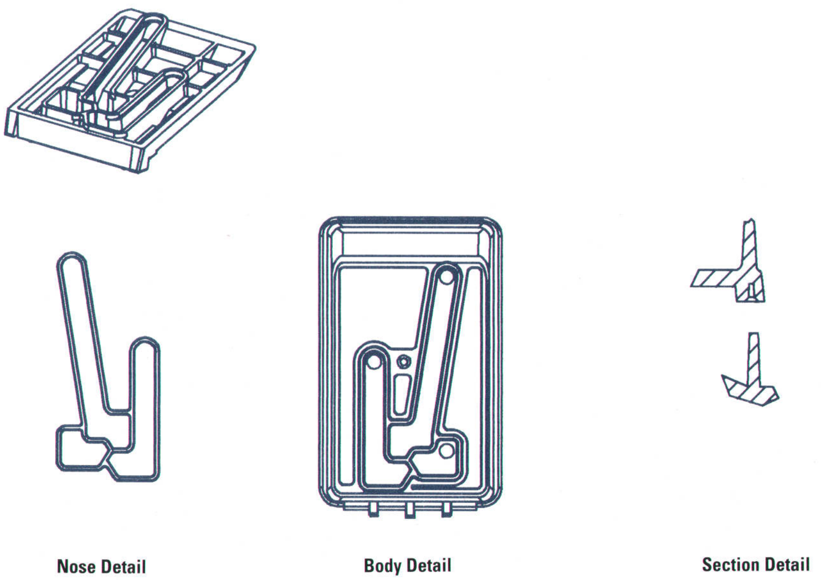
Process Variables

The mechanical energy delivered to the parts can best be described as a function of force, amplitude, frequency, and weld time. Force is dependent upon air pressure, ram area, down speed, and trigger force. Air pressure can be adjusted via a regulator. Ram area is a constant. Down speed is adjusted by regulating the flow on the backside of the ram. Trigger force is read from a transducer, and the triggering level can be varied and controlled by the power supply. The amplitude present at the horn/part interface is defined by the equation:

Amplitude = (Converter Output) × (Booster Gain) × (Horn Gain).

Converter output is constant at 0.0008 inch. Booster gain can be varied in discrete amounts. Horn gain is constant and depends upon the horn design. Frequency is a constant, in this case 20 kHz, although welders can be purchased with different frequencies. Weld time is a variable that can be adjusted and controlled by the power supply. Therefore, the way in which energy is delivered to the plastic parts can be varied by as many as five independent variables.

One of the more important parameters affecting weld quality is the amplitude of the mechanical energy. The "stack" used to deliver the weld energy to the plastic parts consists of the converter, the booster, and the horn. The horn is an acoustically



Nose Detail

Body Detail

Section Detail

Fig. 1. Weld joint cross-section geometry.

tuned part that transfers the vibration into the plastic parts. The booster is an acoustically tuned coupler used to connect the horn to the converter. The size and shape of a booster affect its gain. Standard booster gains range from 0.5 to 2.5. The amplitude can be varied by interchanging boosters with differing gains. As mentioned previously, the converter is a transducer that converts the electrical signal to a mechanical vibration.

Process Development

To define a process for welding these parts it was necessary to understand how each variable affected the quality of the weld. Moreover, a method for quantifying the quality of the weld joint was needed. The quality of the joint was measured by how well it met the part objectives, namely parallelism and leak tightness. The parallelism was measured with a drop indicator, a fairly trivial task. Deciding how to check for leaks, however, became a long affair with many unacceptable methods tried. At long last, it was decided that a sensitive pressure leak tester would be sufficient for leak testing. The quest became one of correlating the inputs (pressure, weld time, down speed, trigger force, and booster) with the outputs (leak rates in three chambers and datum parallelism). One of the largest advancements in process development came from using an experimental design and statistical analysis program to model the correlation. This program proved to be invaluable in choosing the parameters for the process.

The use of the program was an iterative process. A very broad-based experiment was first designed to determine which variables and interactions were significant. These results often yielded trends in variables that would further limit the possible operating conditions. These possible operating conditions were then used as center points for narrower experiments. This iteration continued until an acceptable operating condition was attained.

Pitfalls

This has not been an easy road. There have been some pitfalls associated with ultrasonic welding as well. These include horn manufacture and reliability, and tight plastic molding tolerances.

As described above, the horn is an acoustically tuned part used to transmit the vibratory energy into the plastic parts. The two metals commonly used for horn manufacturing are aluminum and titanium because of their good acoustical properties. During a horn's use, it undergoes longitudinal flexure and compression. The stresses involved depend upon the amplitude of the strain within a horn element. Horns are usually manufactured to handle half-wavelength and full-wavelength vibrations. Half-wavelength horns have one node, while full-wavelength horns have two nodes. During vibration, the material at or near a node undergoes the largest stresses. The larger the stresses, the shorter the horn life, as is the case with typical fatigue failures. Early designs of the horn were half-wavelength. Unfortunately, this placed a stress riser very near the nodal region of the horn. This stress riser proved to be the source of fatigue cracking in these horns. By changing the design to a full-wavelength horn, we moved the stress riser to a more benign location, farther away from a nodal region. The full-wavelength horn was chosen as the long-term solution for the manufacturing process.

Further problems were encountered because of plastic part variation. When the multiple-cavity molds were brought up, there were numerous permutations of nosepiece and body combinations. Of all the possible part combinations, only a small fraction worked with the existing process. Finding a process that would work for all part combinations was extremely challenging. There are three possible alternatives for finding a solution to this problem: make the parts closer to the same (decrease the tolerances for acceptability), find a very wide process window that will weld parts that are not quite identical, or some combination of the two. The third solution was chosen. Trade-offs in part tolerances and process margins were made. By changing certain part tolerances, a process window was established that could weld all part combinations. This challenge will remain whenever a new mold is brought on line, as long as this joint design is in use.

Acknowledgments

The author was not alone in the work on the manifold design. The success of this design and process has been fashioned by the hands of many engineers. Of note are Michael Allison, Marvin Wong, Bill Peters, and Patrick Boyd.

Gregory W. Blythe
Hardware Design Engineer
Inkjet Components Division

Conclusion and Acknowledgments

This project has been very successful for the Inkjet Components Division in Corvallis. The high-volume color print cartridge assembly line has come up to speed faster than any previous production line of its type, including duplications of the black print cartridge assembly line. This success is in large part because of the dedicated efforts of numerous groups and individuals. The manufacturing engineering project manager was Bob Conder. The manufacturing engineers who designed the line included Joe Santich, Bill Peters, Gary Lutnesky, Doug Reed, Mike Monroe, Dennis So, Wayne Traina, and Ken Frazier. We would also like to acknowledge the many contributions of the build technicians, maintenance technicians, and production operators.

Reference

1. M.C. Huth, et al, "CIM and Machine Vision in the Production of Thermal Inkjet Printheads," *Hewlett-Packard Journal*, Vol. 39, no. 5, October 1988, pp. 91-98.

Adhesive Material and Equipment Selection for the HP DeskJet 500C/DeskWriter C Color Print Cartridge

The adhesive joins the printhead to the cartridge body and maintains color ink separation at the interface. The encapsulant protects the electrical bonds. Special equipment was designed to dispense these materials with high precision in very small volumes.

by Douglas J. Reed and Terry M. Lambright

Structural adhesive is the material that creates the bond between the active portion of the print cartridge (the printhead) and the passive portion (the cartridge body). Adhesive requirements for the three-color HP DeskJet 500C/DeskWriter C print cartridge are few, but critical. Sufficient bond strength must be provided over all environmental conditions throughout the entire lifetime of the print cartridge (a maximum shelf life of eighteen months and a maximum in-printer service life of six months). This requirement is very similar to the corresponding requirement for the original monochrome ink cartridge introduced with the HP DeskJet printer. However, the three-chamber design of the color cartridge adds the important function of maintaining color ink separation at the interface between the printhead and the cartridge body.

The primary role of the encapsulant is to provide protection for the bonds that are created between the TAB (tape automated bonding) circuit leads and the bond pads on the printhead. The TAB circuit is the device that carries electrical signals from the printer to the resistors on the printhead. The encapsulant also coats and protects the portion of the TAB leads that spans the gap between the edge of the TAB circuit and the edge of the printhead.

Stress Tests

To decrease product development test time, various accelerated stress tests were used to verify the acceptability of the adhesive systems over the projected lifetime. These stress tests exposed the adhesives (and the entire print cartridge in some cases) to mechanical, thermal, and other environmental stresses that were meant to reveal weak points in the design and assembly of the print cartridge. Because the ink is in contact with the adhesives during these tests, chemical interactions with the ink components are also accelerated. The rigorous test regimen used in this project was helpful in understanding the root causes of various failure modes.

Cycle-Time Constraints

To obtain the desired manufacturing cycle time, it was necessary that the structural adhesive and the encapsulant reach full cure in a relatively short time. Material selection was therefore restricted to adhesives that could be rapidly cured by an appropriate energy source. Multiple stations were necessary to obtain the required throughput, but by selecting quick-cure adhesives, cost was minimized by reducing the total equipment needs.

Compatibility with Machine-Vision Inspection Systems

During the manufacturing process, the bead of structural adhesive is inspected by machine vision to verify the presence and the quantity of adhesive in critical regions (see article, page 87). As a result, it was necessary to select a material that has high visual contrast with the pen body so the vision system can process the image. In fact, the final material selection was partially based on this requirement, since the competing material had very little contrast with the black print cartridge.

Mechanical Properties

Mechanical properties are primarily determined by the adhesive type and the degree of cure developed during the assembly process. Because of manufacturing constraints and the asymptotic nature of chemical reactions, it is rarely possible (or even desirable) to obtain 100% cure. In practice, the adhesive is cured to less than 100% because the desired level of mechanical and chemical properties can be obtained under those conditions.

To select the proper structural adhesive for the color print cartridge, several thermoset systems were evaluated. All of them were fast, thermal-cure systems typically used in the electronics industry. The primary differences between the competing systems were the type of catalyst and the level of additives, fillers, and adhesion promoters. These different additives provided a range of adhesion levels, solvent resistance, cure rates, and other characteristics.

A uniform layer of adhesive is more critical for the color print cartridge than for the monochrome application. This is because the dispensed bead on the color cartridge is narrower and therefore the probability is higher that an adhesive defect

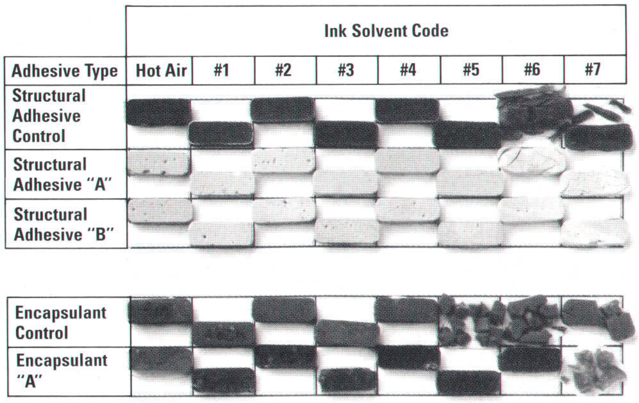


Fig. 1. Adhesive test blocks after accelerated stress tests.

will allow ink leakage between colors or to the external environment. The narrower bead widths also make it necessary to have higher adhesion strength on a unit-area basis. In the three-channel design, no extra room is available to lay down additional beads to provide higher adhesion. As previously mentioned, this initial adhesion had to be maintained as much as possible throughout the life of the print cartridge. Adhesion retention is a direct function of the cleanliness of the substrate (cartridge), the degree of cure, and the effect of long-term interactions with the ink.

The encapsulant materials that we investigated included two different chemical families. The material used on the original monochrome cartridge has some advantages in terms of good adhesion to the printhead. However, its chemical resistance is not great enough to withstand the new ink chemistries introduced with the color print cartridge. The new encapsulant has better chemical resistance even though it is slightly softer and has lower adhesion than the earlier material. This encapsulant survived the rigorous stress tests described above, which destroyed the previous material.

Interaction with Ink Components

Many of the adhesives met all of the performance requirements immediately after cure, but it was crucial to determine how the original material properties were degraded after exposure to inks and to ink components. To evaluate the chemical interactions, small rectangular blocks of each adhesive were poured, cured, and then exposed to ink under various environmental conditions.

As shown in Fig. 1, the test blocks revealed significant information about how the materials performed after immersion. Information gathered from this test and others enabled us to make two critical determinations: (1) Did the ink leach sufficient constituents out of the adhesive to affect ink quality?, and (2) Did the adhesive absorb sufficient ink components to significantly affect the mechanical and dimensional properties of the adhesive?

In some cases, the solvent components of the ink leached polymeric constituents from the adhesive, causing the ink to become contaminated and the adhesive to lose weight. In other cases, weight gain occurred because critical ink components were absorbed by the adhesive and depleted from the ink. In both cases, it proved necessary to evaluate the interactions by performing print quality tests.

Mechanical degradation of the adhesive was also caused by ink interactions. These effects were exhibited as adhesive swelling, loss of hardness, stress cracking, and crazing. Again, the largest factor here was material type. Postimmersion push tests were used to verify adhesion and integrity of the structural adhesive when tested on cartridge bodies.

Dispensing System Design

Applying structural adhesive and encapsulant to the color print cartridge presented significant design challenges compared to the original DeskJet print cartridge application. The trichamber design of the color cartridge and the two-piece assembly process placed more stringent demands on the dispensing systems. This provided the opportunity to advance our capabilities in small-volume dispensing technology and allowed us to build upon our previous manufacturing experience with inkjet products. After searching for outside vendors that could meet the requirements, it became obvious that we would need to design tools in-house to meet the required degree of accuracy. Inkjet technology has continually pushed the limits of standard adhesive technology and dispensing requirements. This new opportunity demanded that we design superior dispense systems that would lend themselves to our unique applications and allow us to leverage those capabilities into future manufacturing processes.

Structural Adhesive Dispensing Requirements

As mentioned earlier, the structural adhesive had the dual role of bonding the printhead to the cartridge and maintaining physical separation between the three ink channels and between the ink channels and the outside world. These functions had to be accomplished in an area that is no larger than that available on the original monochrome cartridge used in the DeskJet family of printers. Compared to previous products, the required bead width was approximately one-third smaller and the placement tolerance was about two-thirds smaller.

Encapsulant Dispensing Requirements

The encapsulation process is also more stringent than its predecessors. Previous applications controlled only encapsulation height and encroachment onto the orifice plate. However, to permit a different sealing mechanism in the printer service station, the color cartridge requires control on all sides of the dispense pattern. In addition to controlling orifice plate encroachment (for proper printer wiping) and height (to prevent crashing into the paper), the color cartridge has the additional constraint of maintaining a "free" area outside the encapsulation zone to allow access by the printer service station. This means that the process specifications are a factor of two tighter than what had previously been required for encroachment as well as requiring dimensional control in five directions (both sides, both ends, and height) instead of the previous two. The final problem for the dispensing system was that the cartridge body is assembled from two pieces—a nosepiece welded to a body. This provided the challenge of an increased tolerance stackup when the part was presented to the dispenser and made it likely that the dispensed surface would not be flat. A background concern affecting all of the others was the knowledge that a steep production ramp would necessitate low cycle times and high reliability.

Tooling Objectives

The primary objectives for the tool included controllable adhesive volume delivery, a multiaxis closed-loop control system, and a system to locate cartridge position. Means had to be provided to locate and calibrate the dispense tip so the system could perform without operator intervention.

Adhesive dispense systems from the electronics and automotive industry were evaluated, site visits were conducted, and trade show information was digested. None of the systems we evaluated would provide the tolerances or multiaxis synchronization that was required. While some of the dispensers in the automotive industry had the desired closed-loop features, the differences in scale were just too large to bridge. It was ultimately concluded that no other applications required the same combination of precision, speed, and small dispense volume.

Production Equipment

A CNC-type controller was selected to provide the required four-axis synchronization. Cartridge location is determined by position sensors, allowing us to locate the part in space in all three axes. Dispense tip location is determined by multiplexing the vision engine that was already in place for online inspection of the dispense pattern. With both the cartridge and the dispense tip located in space, the dispense pattern can be adjusted for each individual cartridge. We selected a controllable dispense mechanism that is force-fed from a pressurized syringe. This solution provides most of the benefits of true positive displacement even though it has the drawback of not accommodating viscosity changes in the adhesive. Fig. 2 shows the adhesive pattern.

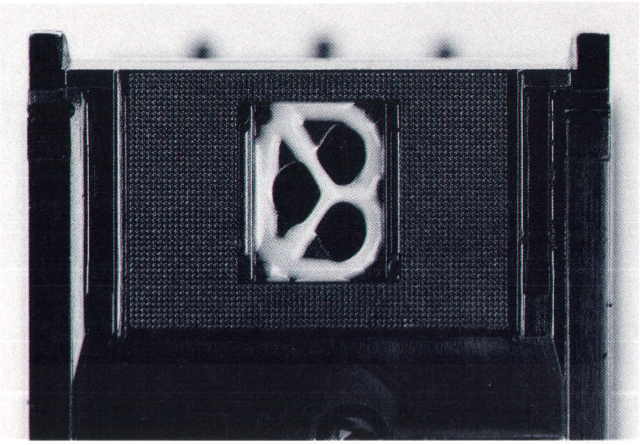


Fig. 2. Adhesive pattern dispensed by the production equipment.

The final system design meets all of the original demands that were imposed by the color print cartridge. It also goes a long way toward advancing our understanding of dispensing technology. Finally, the system provides us with the capability of modifying the design in the future to provide true positive-displacement dispensing for other applications that require similar precision.

Acknowledgments

We would like to thank Melanie Feder for her invaluable assistance in evaluating many of the adhesive systems, providing the samples shown in Fig. 1, and consulting on various aspects of materials compatibility testing. We also want to thank Gary Lutnesky for his direction and help in designing the dispense system.

Machine Vision in Color Print Cartridge Production

In production of the tricolor print cartridges for the HP DeskJet 500C and DeskWriter C printers, machine vision is used for filter stake inspection, adhesive and encapsulant dispenser calibration, structural adhesive inspection, and automatic print quality evaluation.

by Michael J. Monroe

Machine vision can be described as the synthetic acquisition, analysis, and interpretation of images, usually to provide feedback and control for some automated activity. Machine vision has been implemented through the marriage of video camera and display systems to computer technology, and it is often associated with some form of artificial intelligence. Most machine vision applications entail massive data reduction from the millions of bits that represent the images to often a single bit indicating a pass or fail status.

Machine vision is usually used for machine or robot guidance, defect identification and classification, part and assembly alignment, and feature measurement. Automated production environments are the ideal home for machine vision. There it can be applied to relatively simple, repetitive tasks, and cycle time is of critical importance. These applications are distinguished from the more generic digital image processing used in areas such as astronomy, bioscience, and satellite image enhancement, for which the algorithms tend to be very CPU-intensive and execution time is of less importance.

Increased quality is usually the primary motivation for automating a task with machine vision. Machine vision can eliminate the subjectivity often found in manual inspection operations. Quality can also be increased through reduced inspection error rates by eliminating operator fatigue. The improvements in quality that machine vision can help attain may be vital to the long-term competitiveness of a manufacturing operation.

Illumination and Optics

No discussion of machine vision would be complete without emphasizing the importance of illumination and optics. These aspects are critical to nearly all machine vision applications, and indeed the viability of a particular application may hinge upon the design of the illumination and optical systems. Careful design of the illumination system for a particular machine vision application can provide enhancement of contrast or of certain features of interest in the field of view, or it can be used to filter out features that may confuse the algorithm. One obvious advantage of image enhancement through illumination is that it operates on the entire image instantaneously, much faster than any digital image processing. Techniques such as light and dark field illumination can greatly improve the contrast between features of interest and the background. Linear or circular polarization of the illumination or application of on-axis or off-axis illumination can mitigate the effects of specular reflection from metallic or polished surfaces, which can often saturate the imaging sensor within the camera, causing the resultant image to be of little use. In some applications there is no usable contrast between the features of interest and the background, and imaging using illumination in the visible wavelengths is impossible. However, ultraviolet illumination, to which most cameras have reduced sensitivity, can induce some materials to fluoresce in the visible region of the spectrum, thereby providing usable images. Some machine vision applications require only the information that is contained in the outline or silhouette of the object of interest. Backlighting or placing the object between the camera and the illumination source can provide images of such high contrast that they are nearly binary in nature.

Design of the optical system is equally important. CCTV or C-mount lenses, commonly employed with standard video cameras, are often used in machine vision. They are inexpensive, but they provide resolution and contrast that are marginal or inadequate for many applications. Lenses designed for use with standard 35-mm cameras or photoenlarging lenses are somewhat more expensive, but they provide much higher resolution and contrast, and the control of aberrations such as distortion and field curvature is much better. For applications that require extremely high resolution and magnification, micrographic lenses are usually the best choice. The key parameters to be considered during the design of the optical system are the working distance or the distance between the lens and the object, the magnification, and the focal ratio. The working distance and the required magnification are used to determine the focal length of the lensing system, and the focal ratio is a function of the available illumination. Other considerations such as shock and vibration and rigidity of the mounting systems are also very important.

Software

The often crucial nature of the illumination and optical systems design should not overshadow the design of the software that is to be run on the machine vision engine. During the development of most machine vision applications, the majority of the time is devoted to the design, coding, and debugging of this software. The software for a typical application can be broken into five segments. First is image acquisition, which entails synchronization of the video source and then digitization and storage of the image data into frame buffer memory. The most common video source is a standard RS-170 monochrome camera, which transfers the data representing a full frame in 33 milliseconds. After acquisition of the image, some sort of enhancement of the features of interest is usually done. This may consist of high-pass or low-pass convolution filters for treatment of the edges of the features, or morphological filtering, which can eliminate noise pixels in an image. Image segmentation usually follows to provide some means of separating the features of interest from the background or other extraneous parts of the image. Binarization or thresholding of the image is a segmentation technique used to convert an image represented by many gray levels to one consisting only of regions of pure black or pure white. This method also greatly reduces the amount of data that must be processed during the analysis of the image. Feature extraction is then used to derive data on the details of features of interest that will be used during the interpretation phase of the algorithm. Analysis of the gray level sums of the pixels in all of the rows and columns of an image results in profile information such as the position, size, or shape of features. Template or pattern matching uses a previously stored model to locate the position of a matching model in the acquired image. Connectivity analysis provides detailed information on linked areas within a binary image. Morphological analysis can provide data on the shapes of various features of interest, in addition to its usefulness as a noise filter. Finally, all of the acquired data representing the image must be interpreted so that some useful result can be found. This may take the form of calibration or position coordinates that can be used for machine guidance, or it may simply be the pass or fail result of an inspection task.

Machine Vision in InkJet Component Production

Hewlett-Packard has used machine vision for years to improve efficiency and quality in the manufacture of integrated circuits, printed circuit assemblies, calculators, thermal inkjet printheads, and many other products. The Inkjet Components Division has been a leader in the incorporation of this technology into the production processes of its products. Beginning in 1983, machine vision has been used in the final assembly of the HP ThinkJet print cartridge to inspect for defects such as poor structural adhesive placement and leaks. In addition, samples from every print cartridge are analyzed using machine vision to screen for various print quality defects. These machine vision applications were further refined and new ones were developed for the manufacture of the HP DeskJet print cartridge. Machine vision is used to assist very high-precision part alignment and placement machines in the attachment of orifice plates to the thin-film substrates of the printhead assembly and in the placement of the printhead assembly onto the plastic print cartridge body.[1] In addition, a new fully automatic high-speed print quality tester was developed using machine vision to inspect print samples of 100 percent of the print cartridges manufactured.

Color Print Cartridge Production

In the early 1980s, when general-purpose machine vision engines were first made available, many different manufacturers offered products that were similar in function but very different with respect to instruction sets and software development environments. Since that time, there has been a shakeout in the marketplace, and fewer different machines are being offered today. However, the products that remain still tend to be very proprietary in nature, and little if any effort has been made to create the type of standards that exist today in the computer market. This lack of standardization and especially the preponderance of proprietary instruction sets can greatly reduce the efficiency with which new machine vision applications can be developed. Therefore, it is of great benefit to choose one product that has adequate capability for most if not all applications. This benefit of standardization is particularly important in the manufacturing environment, where machine vision applications tend to proliferate and support becomes a significant issue.

For the HP DeskJet 500C/DeskWriter C print cartridge production line we have chosen an engine that offers a range of optional features and capabilities, but uses a common instruction set and software development and debug environment throughout. In a similar vein, a single line of very high-quality photoenlarging lenses has been selected to provide several focal lengths and focal ratios that are suitable for most machine vision applications.

Filter Stake Inspection

Because the printhead for the Deskjet 500C printer is a tricolor printhead it has a separate chamber for each of the cyan, magenta, and yellow primary colors. At the bottom of each of these chambers near the nose of the print cartridge there is a stainless-steel filter screen that is heatstaked to a standpipe. The filter staking process is a critical one, and machine vision has been incorporated to ensure that every print cartridge body coming from the filter staker tool has three filters that are properly staked to the standpipes of the print cartridge body. Each filter is inspected for presence, proper centering on the standpipe, and an adequate weld ring indicating satisfactory attachment to the standpipe. The filter staker consists of an individual staking station for each of the three filters, and each of these stations is equipped with camera, optics, and illumination to enable inspection immediately after completion of the staking operation (see Fig. 1).

The filter stake inspection process is a good example of a machine vision application in which optics and illumination determine the viability of the process. Because of the flow characteristics of the plastic material used in the print cartridge body, relatively little of the material is drawn up into the weave of the filter screen during the heatstaking process. As a result, images that displayed little or no contrast between the weld ring and the filter were all that could be achieved with nearly on-axis illumination. It was found that contrast could be enhanced somewhat by increasing the angle of illumination from the normal to the filter. However, this angle was limited by the shadowing caused by the walls of the print cartridge body. At this point, contrast was still inadequate for reliable imaging, and further development

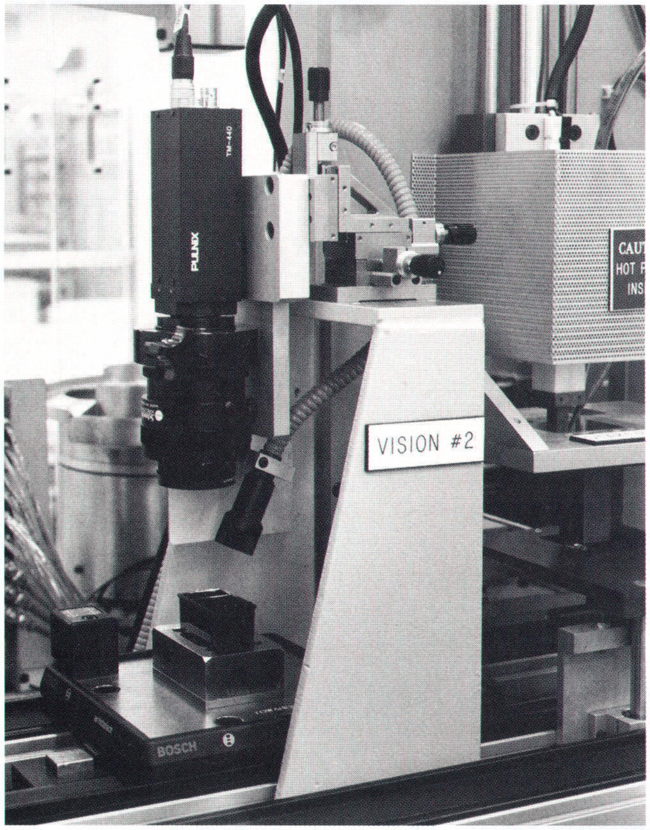


Fig. 1. The filter stake inspection station uses off-axis illumination and polarizing optics.

was required. It was determined that although the filter screen material is relatively rough and provides reflection that is primarily diffuse in nature, there is also a specular component that tends to saturate the imaging sensor in the camera and thereby further reduce contrast. Therefore, linear polarization of the illumination source is used to mitigate the effects of specular reflection and produce images adequate for a robust inspection process. The angular orientation of the filter screens greatly affects the reflectance of the material, and as a result, the images vary from brightness levels too low to be of use to levels so high as to cause saturation of the imaging sensor. In most machine vision applications, the camera's automatic gain control (AGC) is disabled so that the gray levels digitized by the vision engine provide a more accurate representation of the object. However, it was found that the camera itself could be made to cope with the wide variations in brightness simply by reenabling the AGC.

The inspection algorithm is based upon gray-level row and column profiling for determination of filter centering on the standpipe, and gray-level pixel sums and ratios of sums to ascertain filter presence and gauge the continuity of the weld ring. The inspection based upon the analysis of gray levels is very robust and provides false-pass and false-reject rates that are nearly zero. In addition, image acquisition and processing are extremely rapid, and a single machine vision engine can support all three of the inspection stations. At the completion of the inspections, the test results are sent to the PLC (programmable logic controller) that controls the filter staking process, and that data is then transferred to a failure off-load station via pallet code blocks.

Adhesive and Encapsulant Dispensing

The print cartridge for the HP DeskJet 500C printer contains three inks of different colors, and it is critically important that complete separation between these inks be maintained at all points throughout the print cartridge. Any ink mixing will result in contamination of the inks and of the colors that they produce on the printed page. The structural adhesive that bonds the thin-film printhead to the nose of the print cartridge plays a vital role in providing the required ink isolation. Because all three inks are channeled to the same printhead, the tolerances associated with the placement of the pattern of structural adhesive are more restrictive than those of other thermal inkjet print cartridges. As a result, it was recognized that some means of providing an efficient and accurate calibration of the exact position of the needle tip of the adhesive dispenser was required. A similar requirement existed for the encapsulant dispensing process. Machine vision was chosen as an aid to ensure accurate placement of the structural adhesive and encapsulant by providing the dispenser system controller with accurate x, y, and z coordinates of the location of the dispenser needle tip.

Each of the dispenser stations is equipped with two orthogonally positioned cameras and associated optics (Fig. 2). During the calibration process the positioning system moves the dispenser needle to a calibration position so that images from both of the cameras can be captured by the machine vision engine. Through analysis of both of the images the vision engine can determine the x, y, and z coordinates of the dispenser needle tip in pixel space. Then the coordinates in pixel space are converted to coordinates in the coordinate space of the dispenser positioning system using calibration coefficients previously stored in nonvolatile memory of the machine vision engine. This position data is then sent in the form of offsets to the dispenser system positioning controller via an RS-232-C datacomm link. The calibration process is fully automatic and requires operator assistance only for initiation of the procedure. It is normally performed only at the beginning of a shift or when a dispenser needle has been changed.

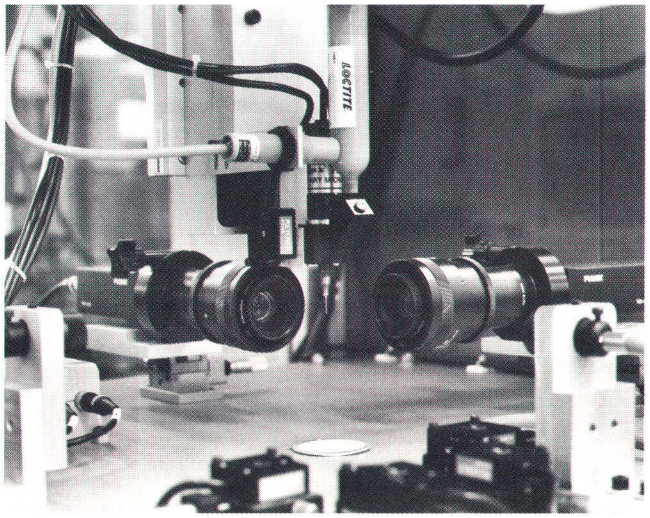


Fig. 2. The adhesive and encapsulant dispensing stations are equipped with two orthogonally positioned cameras using backlighting to determine the position of the dispensing needle tip accurately.

All of the data necessary to determine the precise position of the dispenser needle can be inferred from the profile or silhouette of the needle tip. This provides an ideal opportunity for backlighting or placing the object to be imaged between the illumination source and the camera. Backlighting typically produces an image of extremely high contrast, and often the image is nearly binary with the object appearing dark against a bright background. Since the required accuracy of the machine-vision-based calibration system is very high, a significant amount of magnification is required to achieve the necessary resolution. An appropriate photoenlarging lens with adequate extension was selected to provide the desired field of view.

The machine vision engine uses an edge-enhanced pattern matching technique to locate the dispenser needle tip in the field of view to subpixel accuracy. The template or model describing the salient features of the object to be located is stored in the nonvolatile memory of the vision engine during a setup procedure. A unique model for each of the cameras is stored. These models are then used during each subsequent calibration procedure to determine the locations of the desired features in the two images. Also during the setup procedure, each of the cameras is calibrated using a precision reticle grid placed in precisely the same position as the dispenser needle. Nine intersections on the grid are then used as calibration points so that an array of coefficients can be produced and then used to convert coordinates in pixel space to machine coordinate space. The machine vision engine has dedicated RS-232-C datacomm ports connected to the adhesive dispenser system controller and the encapsulant dispenser system controller. After the offsets in machine coordinate space have been computed by the vision engine, they are sent to the respective controller via these links. Since these calibration procedures are performed on a fairly infrequent basis, speed of execution is not of paramount concern. As a result, the two dispenser systems share one machine vision engine with the structural adhesive inspection station. The automatic calibration process has proved to be robust and to provide precise and repeatable dispenser needle positioning data to the dispenser system controllers.

Structural Adhesive Inspection

Although the machine-vision-based dispenser calibration process provides an efficient means of compensating for variation in needle tip positions, it cannot prevent other possible defects in the structural adhesive pattern. As previously stated, the integrity of this adhesive pattern is vital to the proper functioning of the print cartridge. Therefore, an automatic inspection station was developed using machine vision to ensure that each print cartridge body has an acceptable and accurately placed pattern of structural adhesive before reaching the head attach station. Because the adhesive is very soft and easily disturbed after leaving the dispenser station, it was imperative that a noncontact means of inspection be used. The pattern of structural adhesive for each print cartridge is visually inspected for proper centering and area of the adhesive and for proper centering and area of each of the three channel openings. In addition, the pattern is inspected to ensure that the bead of adhesive is continuous and free of sections that are excessively narrow or wide.

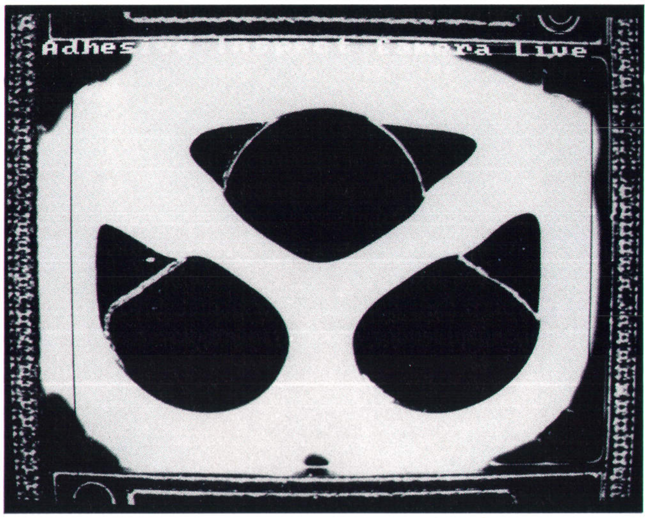
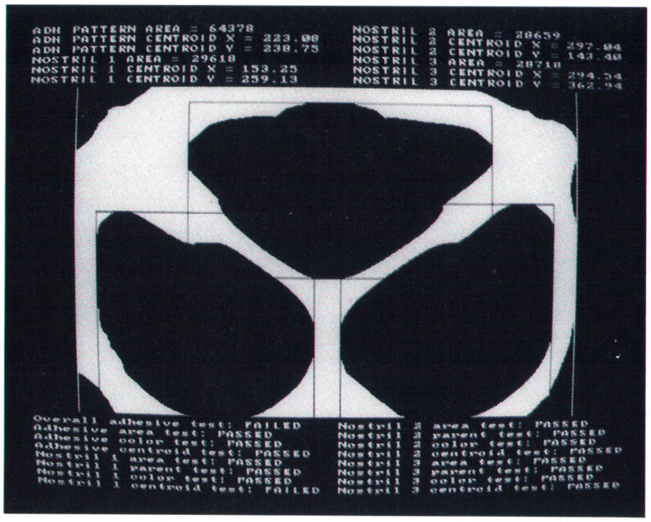


Fig. 3. Unprocessed image of the structural adhesive pattern.

The structural adhesive that was originally to be used provided virtually no contrast when placed on the black plastic nose of the print cartridge and illuminated with light in the visible portion of the spectrum. However, when ultraviolet light was used to illuminate the adhesive, fluorescing compounds contained in it caused it to appear light blue against the black background of the nose. This technique provided some usable contrast, but it was marginal and not likely to succeed with the variabilities commonly found in a production line environment. A new adhesive was selected. The new material provides images of extremely high contrast and allowed the development of a very robust inspection process. Lighting for this application is provided by a standard fluorescent ring lamp, which supplies adequate illumination that is very uniform across the field of view. The standard photoenlarging lens provides excellent, zero-distortion images (Fig. 3).

The most serious defect in the structural adhesive pattern occurs when an insufficient amount of adhesive has been placed by the dispenser station, and a break or discontinuity occurs in the bead. Continuous or connected regions in the image are most effectively evaluated through the application of connectivity analysis. Because the adhesive provides images of extremely high and consistent contrast, binarization and the subsequent connectivity analysis have proved to be a very robust inspection technique. Indeed, the images are of such quality that adaptive thresholding, which is usually required to compensate for variations in contrast and illumination, is not necessary. This results in a decrease in the overall cycle time for the process. Morphological filters are applied to the binary image to remove random noise pixels that could taint the connectivity data, and to reduce the width of the bead image so that portions that represent insufficient adhesive will actually separate and can be easily identified. The database produced from the connectivity analysis furnishes information such as area, centroid, and perimeter of the connected regions found in the image, and a description of the hierarchical relationship between those regions. Much of this data is easily compared with test limits to determine the adequacy of the adhesive pattern (see Fig. 4). However, because of the tight tolerances associated



```
ADH PATTERN AREA = 64378                 NOSTRIL 2 AREA = 28659
ADH PATTERN CENTROID X = 223.00          NOSTRIL 2 CENTROID X = 297.04
ADH PATTERN CENTROID Y = 238.75          NOSTRIL 2 CENTROID Y = 143.48
NOSTRIL 1 AREA = 29618                   NOSTRIL 3 AREA = 28718
NOSTRIL 1 CENTROID X = 153.25            NOSTRIL 3 CENTROID X = 294.34
NOSTRIL 1 CENTROID Y = 259.13            NOSTRIL 3 CENTROID Y = 362.94

Overall adhesive test: FAILED            Nostril 2 area test: PASSED
Adhesive area test: PASSED               Nostril 2 parent test: PASSED
Adhesive color test: PASSED              Nostril 2 color test: PASSED
Adhesive centroid test: PASSED           Nostril 2 centroid test: PASSED
Nostril 1 area test: PASSED              Nostril 3 area test: PASSED
Nostril 1 parent test: PASSED            Nostril 3 parent test: PASSED
Nostril 1 color test: PASSED             Nostril 3 color test: PASSED
Nostril 1 centroid test: FAILED          Nostril 3 centroid test: PASSED
```

Fig. 4. Structural adhesive pattern image after processing and analysis, showing test data and results.

with the adhesive placement and the positioning errors inherent in the pallet lift and index mechanism, it is essential to locate a feature on the nose of the print cartridge with which to compare the adhesive pattern location. The central channel opening in the nose is located using edge-enhanced pattern matching before the connectivity analysis so that its position can be used as a reference in the subsequent tests.

Automatic Print Quality Tester

The quality of the HP DeskJet 500C print cartridges leaving the production line is of the utmost concern. This concern precipitated the development of the automatic print quality tester, on which 100 percent of the print cartridges manufactured are tested. The automatic print quality tester is the final arbiter of print cartridge quality, and it is its responsibility to provide information regarding print cartridge performance so that the defective ones can be culled from those that are acceptable at the end of the production line. The automatic print quality tester is capable of detecting a wide variety of print cartridge defects. It is a fully automatic tester that receives print cartridges seated in nests on a rotary table, fires the print cartridges so that test patterns are printed on paper, and then uses machine vision to examine and evaluate the printed patterns to detect any possible defects in the print cartridges. At the completion of the tests for a given print cartridge, test results in the form of fail codes are both stored locally and sent to a data collection system so that yield and defect summary reports can be prepared. The local database can be interrogated using online utilities, and yield and defect distributions can be displayed in near real time.

The automatic print quality tester uses a relatively straightforward optical and illumination design that incorporates high-quality macroscopic lenses and fiber optic ring illuminators (Fig. 5). The challenge was how best to deal with the cyan, magenta, and yellow primary colors. Standard color imaging techniques using a color video camera with RGB outputs were considered but rejected because frame storage memory requirements were tripled relative to those for monochrome imaging, and most color cameras provide significantly reduced spatial resolution. In addition, before meaningful color image analysis can take place, the contents of the frame buffers that contain the gray levels representing the red, green, and blue video camera outputs must be combined and converted into a more useful color coordinate system such as HSI (hue-saturation-intensity). The time required to make these conversions in software was prohibitive with respect to the overall cycle time of the tester. Because nozzle defects and not hue shifts are the most common defect types, it was decided that true color imaging was not required, and an approach using color filters was taken. If the machine vision engine is presented with images of high enough contrast with the ink appearing dark on the white paper background, it can easily perform the required analyses to determine the health of the print cartridge. The cyan and magenta primaries provide usable contrast without any special optical techniques, but the yellow ink is nearly invisible to the CCD camera. By placing a blue interference filter in front of the camera lens, the yellow test patterns are made to appear dark enough to provide usable contrast in the image. A similar enhancement is afforded by a green interference filter on the camera that is used to image the cyan and magenta test patterns. It greatly increases the contrast of the magenta patterns, but has little effect on the cyan. This technique in effect reduces the number of images that must be acquired for each test pattern from three to two, and it eliminates the necessity of time-consuming color coordinate conversions.

As previously mentioned, the automatic print quality tester ensures the quality of the print cartridges leaving the production line. Special test patterns were developed to reveal the several types of possible print cartridge defects. These test patterns help ensure that the machine vision engine can provide an effective screen for defective print cartridges. These patterns are printed by the print cartridge under test, and after completion of the printing phase, the paper is advanced (Fig. 6) until the test patterns are within the fields of

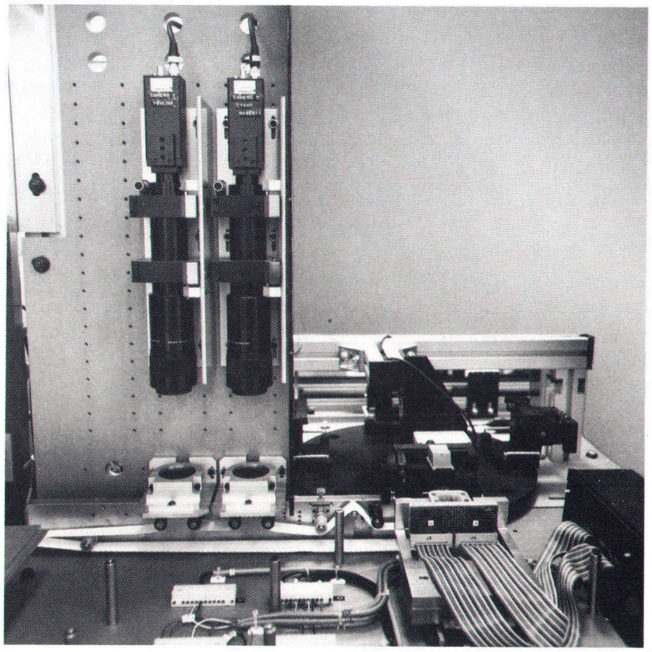


Fig. 5. A view of the automatic print quality tester showing the illuminators, cameras, optics, rotary table, and print cartridge nests.

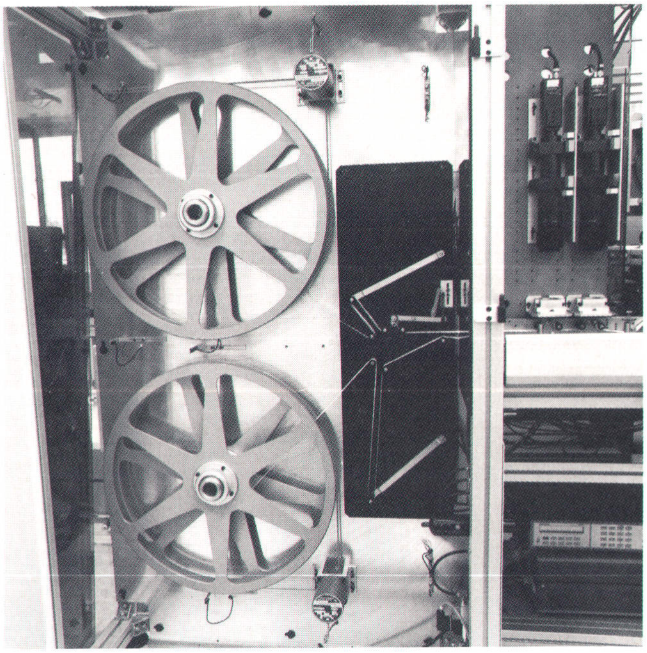


Fig. 6. A view of the automatic print quality tester showing the paper feed and take-up reels and tension arms.

view of the cameras. Signals sent between the paper motion controller and the machine vision engine synchronize the acquisition of all of the test pattern images. The machine vision engine then processes and analyzes the images using techniques such as row and column profiling, edge enhancement, and grey-level pixel sums to identify any defects present. At the completion of the image analysis phase, fail codes that represent any test pattern defects that may have been detected are sent to the HP-UX*-based system controller via an RS-232-C serial data link. Because of the physical separation between the cameras, and for efficiency and throughput considerations, during any given test phase the machine vision engine is evaluating the test patterns from two different print cartridges. It is the responsibility of the system controller to receive and sort the fail codes for these print cartridges properly so that all of the failure information for a particular print cartridge is associated with the bar-code number for that print cartridge. This failure data is then both stored in a local file and sent to the production line data collection system via a local area network. The automatic print quality tester has demonstrated that a fully automatic machine is capable of effective print quality assessment, and that it can ensure that the print cartridges leaving the production line are of the highest possible quality.

Conclusion

The high-volume production of thermal inkjet print cartridges requires many very accurately controlled and repeatable processes. Most of these processes are implemented using well-designed, high-precision automatic machines. Machine vision has played an essential role in the calibration, inspection, and control aspects of many of these processes, and it has helped ensure interprocess quality as well as the quality of the final product. Machine vision is a key ingredient in the further evolution of computer integrated manufacturing. As the technology continues to advance and the cost of its use continues to decrease, an ever-expanding number of applications will be found for machine vision.

Acknowledgments

Many different people deserve recognition for their efforts in the development and integration of machine vision into the production processes for the HP DeskJet 500C/DeskWriter C print cartridge. Thanks to Dennis So for his expertise in the development of the production line control and data systems. For their assistance in the development of the automatic print quality tester, thanks to Tim Hubley, Gary Lutnesky, and Steve Steinfield, its original designers, and Sang Bradley for the development of the system controller software. Thanks also to Bob Conder for the brainstorming that led to the development of the filter stake inspection process, and for his project leadership in keeping all of us pointed in the same direction.

Reference

1. M.C. Huth, et al, "CIM and Machine Vision in the Production of Thermal Inkjet Printheads," *Hewlett-Packard Journal*, Vol. 39, no. 5, October 1988, pp. 91-98.

HP-UX is based on and is compatible with UNIX System Laboratories' UNIX* operating system. It also complies with X/Open's* XPG3, POSIX 1003.1 and SVID2 interface specifications.

UNIX is a registered trademark of UNIX System Laboratories Inc. in the U.S.A. and other countries.

X/Open is a trademark of X/Open Company Limited in the UK and other countries.

HP DeskWriter C Printer Driver Development

Running on the host computer, the driver provides all of the intelligent formatting, rasterizing, color matching, and dithering for this affordable black and color printer.

by William J. Allen, Toni D. Courville, and Steven O. Miller

A printer driver is a program that provides an interface between an application program and a printer. In the Macintosh and Microsoft® Windows environments, the application/driver interface is well-defined. This allows a single driver to serve all applications in a particular environment.

The HP DeskWriter C and DeskJet 500C printers use the same print cartridges and mechanical components to mark the page. Basic print modes and color imaging techniques are the same for both products. To improve clarity, this article focuses on the DeskWriter C (Macintosh) driver. Where appropriate, significant differences in the DeskJet 500C (Microsoft Windows) driver will be pointed out.

To be competitive in the marketplace, a low-cost printer manufacturer must provide drivers for the two most popular windowing environments, Microsoft Windows and the Macintosh operating system. Manufacturers of high-end printers can always include the widely used page description language PostScript® to guarantee support of the printer. However, including PostScript in a low-cost color printer like the HP DeskWriter C would be prohibitive, significantly increasing the price of the product.

The alternative is to build into the printer only the logic necessary to put the dots onto the paper fast enough to keep the mechanism busy. This requires the driver, running on the host machine, to provide all of the intelligent formatting, rasterizing (converting logical graphics objects to a bit image), color matching, and halftoning. This approach, of using the host machine's CPU power to create the raster image to be laid down by the printer, is the one that we take with the HP DeskWriter C.

HP DeskWriter C Printer Driver

The HP DeskWriter C driver is a program that sits between the user, the application, and the operating system as shown in Fig. 1. This diagram shows the major paths of communication and control managed by the driver.

The user creates a document using the application, then chooses Page Setup, causing the application to call the printer driver's Page Setup command. The printer driver puts up the Page Setup dialog box for the user and returns the modified page size and printer attributes to the application. The application can use this information to reformat the document for the new attributes.

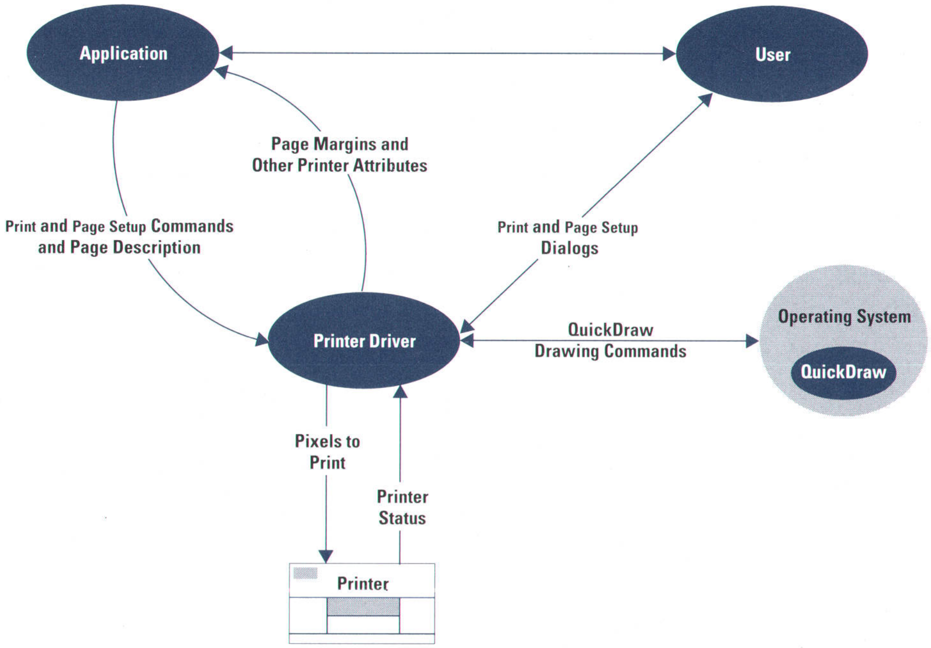

Application
User
Page Margins and Other Printer Attributes
Print and Page Setup Commands and Page Description
Print and Page Setup Dialogs
Printer Driver
QuickDraw Drawing Commands
Operating System
QuickDraw
Pixels to Print
Printer Status
Printer

Fig. 1. Diagram showing the major paths of communication and control managed by the HP DeskWriter C printer driver. QuickDraw, the drawing command language of the Macintosh computer, is similar to GDI (Graphics Device Interface) in Microsoft Windows.

The user then prints the document by selecting Print, causing the application to call the printer driver's Print command. The printer driver puts up the Print dialog box for the user, and when the user is finished making selections the printer driver allows the application to send it a series of page descriptions. These page descriptions are a sequence of imaging commands which describe the page as a series of objects such as text, lines, circles, and raster images (PixMaps). The imaging commands are in QuickDraw, the drawing command language of the Macintosh (similar to GDI, or Graphics Device Interface, in Microsoft Windows).

Now the printer driver will immediately return control of the computer back to the user if the user has enabled background printing, or it will continue processing the print job in the foreground. In any case, the printer driver opens a communication path to the printer (either serial or AppleTalk), determines which print cartridge is installed in the printer, and then proceeds to use a combination of QuickDraw and its own built-in functions to rasterize the page description into a 150-dpi or 300-dpi PixMap.

This PixMap is then adjusted to compensate for the differences between the display and HP printing technology. This may include color matching for the current media, which attempts to make the printed colors appear the same as the colors on the monitor.

The PixMap is finally halftoned, which involves using various patterns of printed dots to simulate all colors that can be produced on the monitor. For instance, since the printer only has cyan, magenta, and yellow inks, it can't directly produce a purple dot. Purple is produced by printing a mixture of red and blue dots in a checkerboard pattern. Each red dot is made by printing a yellow dot on top of a magenta dot and each blue dot is made by printing a magenta dot on top of a cyan dot. This halftoned data is then compressed and transmitted to the printer.

While a communication path is open to the printer, the driver continuously receives status information from the printer so it can report error conditions such as "out of paper" or "wrong print cartridge installed" (e.g., the color print cartridge is installed, but the document only contains black data).

Rasterization

Fig. 2 gives a closer look at how the printer driver rasterizes a page. The transformation from page description to PCL data takes place in several steps. The rasterizing module uses QuickDraw to do much of the work since the page description is in a set of QuickDraw commands anyway, but it replaces some of QuickDraw's functionality where necessary, such as in the rendering of text. Before Apple's System 7.0, QuickDraw did not provide scalable outline fonts, so the HP DeskWriter C rasterizer contains its own outline font renderer called Intellifont™ from AGFA Compugraphic. This provides all Macintosh users with access to high-quality scalable outline fonts for the most commonly used typefaces.

Because most computers have limited RAM, and rasterizing a 300-dpi page can take a lot of memory (up to 32 megabytes to represent a full color page), the driver usually rasterizes a portion of the page at a time in what is called banding. Banding is pictured in Fig. 3. The less available RAM the computer has, the more bands it takes to rasterize the whole page, and the longer it takes to complete the whole page, because there is a fixed overhead in rasterizing each band. Each object must be examined to determine if it falls within the band. If it does then it is rasterized (drawn) into the band. Otherwise, it is ignored (clipped). Objects that cross band boundaries must be rasterized for each band that they touch. For most common documents, banding adds little or nothing to the print time. Only for very complex documents that have a large number of objects, or for very low-memory conditions, which increase the number of bands, does banding significantly affect the time to rasterize the whole page.

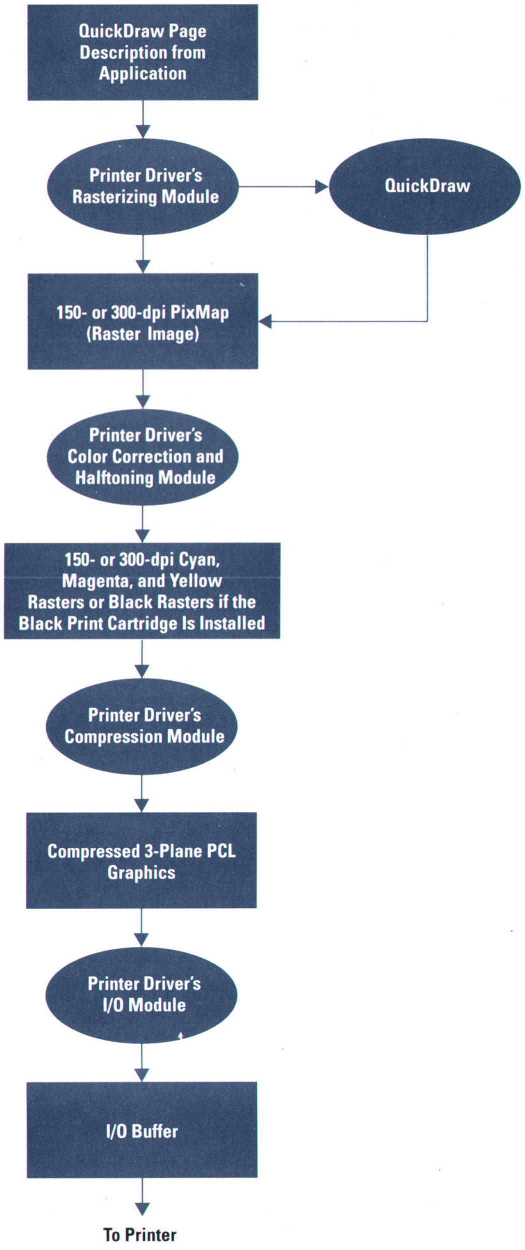

QuickDraw Page Description from Application
Printer Driver's Rasterizing Module
QuickDraw
150- or 300-dpi PixMap (Raster Image)
Printer Driver's Color Correction and Halftoning Module
150- or 300-dpi Cyan, Magenta, and Yellow Rasters or Black Rasters if the Black Print Cartridge Is Installed
Printer Driver's Compression Module
Compressed 3-Plane PCL Graphics
Printer Driver's I/O Module
I/O Buffer
To Printer

Fig. 2. This is the process for rasterizing a page in the HP DeskWriter C printer driver. The rectangles denote data and the ovals denote code.

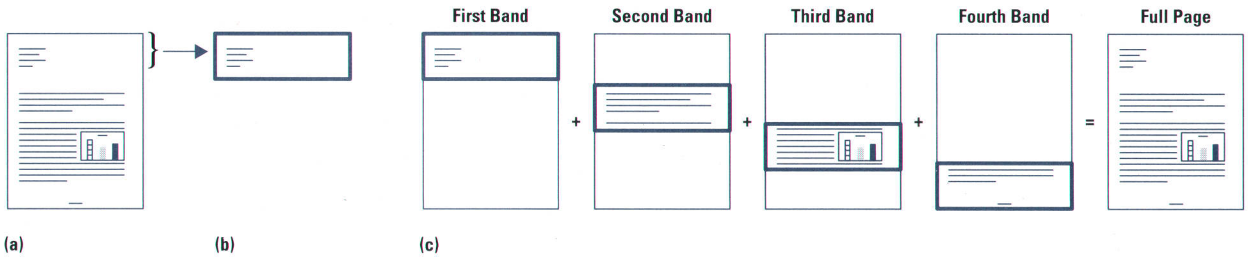


Fig. 3. When the host computer does not have enough available memory to rasterize a complete page, banding is used. (a) This page represents a 256-color document (8 bits per pixel). To rasterize it at 300 dpi would take about 8M bytes of memory. (b) If only 2M bytes of memory are available for rasterizing, then only a quarter of the page can be rasterized at a time. (c) The solution is to use the 2M bytes of memory to rasterize the top quarter of the page. As soon as this band is rasterized and its colors are adjusted, it is dithered, compressed, and transmitted to the printer. While this data is being transmitted, rasterization of the second quarter of the page begins, using the same 2M bytes of memory. I/O buffers ensure that printing continues while successive bands are being rasterized.

User Interface

This product had a wide range of human factors issues and challenges, all of which were analyzed, designed, and tested. The issues included defining a conceptual use model of the product, hardware design, and integrating aspects of color imaging into a simple driver interface.

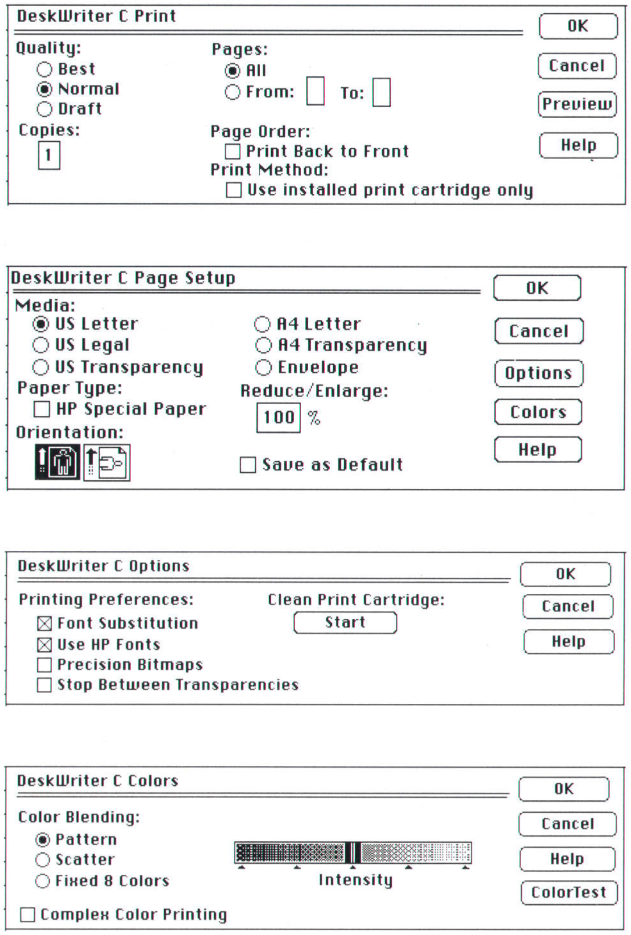
To design a product that meets the requirements of our users, we needed to find out who our users are. The key data point returned from market research was that the HP DeskWriter C should be designed for home and business users who do not currently use color, and who do not want to struggle with their computer or printer. Initial market research and human factors concept testing pointed strongly to the necessity for a user-friendly product with good print quality.

Usability studies and iterations of the user interface followed and continued for the next six months. Usability studies are more detailed than concept testing, and are designed to test users' reactions to specific aspects of a prototyped product. To test the user interface, subjects were asked to perform numerous printing tasks in a number of different scenarios. The users were questioned regarding verbiage, ease of use, and functionality offered. Our goal was to design a usable driver without compromising functionality and without creating confusion for less-technical users. We had three formal usability studies during the product design phase, which required a number of iterations to the driver's interface. The results from each study were published by our human factors engineer along with design recommendations for the next iteration. These design recommendations were based on the users' responses to questions, task completion successes or failures, and other observed data.

Dialog Boxes

Fig. 4 shows the final dialog boxes of the user interface. The first two dialog boxes, Print and Page Setup, are accessed from the menu bar. A key design objective was to make these screens as simple and uncluttered as possible. It was decided that only those functions for which frequent change was required would be in the two main dialog boxes. The more technical functions were put into the Colors and Options screens. This was done to decrease confusion for the typical user who would rarely have use for the added features. The defaults were chosen such that most users would not require the Colors and Options dialog boxes.

Features accessed through the Print dialog box are print quality, copies, page range, page order, and print method. Most of these printing features are common to other Apple drivers. Print Method: Use installed print cartridge only is a feature designed specifically for this product. When this checkbox is on, the alert messages prompting the user to exchange the print cartridge, which normally display when there is a discrepancy between the document content and the installed print cartridge, will not appear. This function is explained in further detail later in this section.



DeskWriter C Print

Quality:
○ Best
● Normal
○ Draft

Copies:
1

Pages:
● All
○ From: □ To: □

Page Order:
□ Print Back to Front

Print Method:
□ Use installed print cartridge only

DeskWriter C Page Setup

Media:
● US Letter
○ US Legal
○ US Transparency
○ A4 Letter
○ A4 Transparency
○ Envelope

Paper Type:
□ HP Special Paper

Reduce/Enlarge:
100 %

Orientation:

□ Save as Default

DeskWriter C Options

Printing Preferences:
☒ Font Substitution
☒ Use HP Fonts
□ Precision Bitmaps
□ Stop Between Transparencies

Clean Print Cartridge:

DeskWriter C Colors

Color Blending:
● Pattern
○ Scatter
○ Fixed 8 Colors

□ Complex Color Printing

Fig. 4. HP DeskWriter C printer dialog boxes.

The Page Setup dialog box features are consistent with other Apple printers in that they allow the user to choose media type, print orientation, and a scaling value. The distinct features offered by the HP DeskWriter C are the HP Special Paper and the Save as Default checkboxes. The HP Special Paper checkbox is selected when printing on special paper. This causes the driver to adjust the way ink is put on the page so it is optimized for HP special paper. Save as Default offers the user the ability to save item states permanently in the Page Setup, Options, and Colors dialog boxes. For example, a user who always prints on U.S. legal-size paper need only select the paper size once, then check the Save as Default box, and click OK to exit the dialog box. US Legal will then be the new default, instead of the factory default, US Letter. This saves the user time by eliminating the need to enter the Page Setup dialog box for each document.

The Options dialog box, accessed by clicking the Options button in the Page Setup box, contains printing preferences and a Clean Print Cartridge button. This button, when selected, attempts to restore print quality if print becomes faint or dots are missing. This is done by activating a print cartridge priming algorithm in the printer.

Users who want more color capability can find it by clicking the Colors button. The color blending selections determine how dots of the three colors (cyan, magenta, yellow) are arranged on the page to create blended colors. Each of these selections produces slightly different results. Pattern produces faster output than Scatter and is recommended for simple solid-color graphics. Scatter takes longer to print and should be used for sophisticated color graphics, such as scanned images and photographs. Fixed 8 allows applications that support color on the Macintosh Plus, SE, Portable, and Classic computers to print the basic eight colors (red, blue, green, cyan, magenta, yellow, black and white). Fixed 8 is the only color blending setting available for these machines.

When the Complex Color Printing checkbox is on, the driver adjusts the printed colors to provide the best match in appearance to the screen. This capability is recommended for complex color graphics such as scanned images, photographs, and complex computer-generated artwork.

The Intensity slider allows the user to select the amount of color ink that is printed on the page. More ink increases the intensity of the images. Users in high-humidity environments may need to move the slider to the right to decrease the amount of ink on the page, since ink bleeding can occur with increased moisture in the air and on the paper. To assist the user in deciding which slider bar location creates the most desirable output, a color test was created. When the user clicks the ColorTest button, a one-page printout is provided that shows the effects of the intensity settings on text, simple graphics, and a complex color image. The ColorTest feature is designed to save the user time by eliminating the need for numerous experimental printouts to determine the best intensity setting.

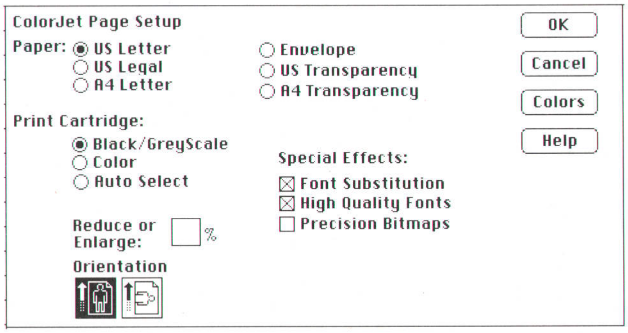
The Colors dialog box was created for the more sophisticated color user. The defaults were chosen such that output will be acceptable for the majority of print jobs. Pattern was chosen for its speed over Scatter and the assumption that most users would probably print simple color graphics. The middle setting in the Intensity slider bar is the default; it is designed to work best in most environments.

Print Cartridge Selection

The dialog boxes described above illustrate the user-driver interaction required to control the printer. In addition, it is necessary for the user to interact with the printer hardware by changing print cartridges. Because this is a completely new task to most customers, numerous prototypes were designed and tested before the final solution was created.

Fig. 5 shows one of the print cartridge swap/page setup prototypes that we explored. The three extra buttons labeled Color, Black/GreyScale and Auto Select were provided to give the user a choice of which cartridge to print with. If the user chose Black/GreyScale, the black print cartridge was expected to be in the printer, and the document would be printed in black. If Color was selected, the color print cartridge would be used. If the expected print cartridge was not in the printer, an alert message would appear prompting the user to insert the correct cartridge. If Auto Select was chosen, the document was scanned, and the user was prompted to insert the black print cartridge if only black was present or to insert the color print cartridge if color was present. With a multipage mixed document (black and color pages), the pages would be ordered so that all the black pages were printed first, followed by the color pages, eliminating the need to swap print cartridges more than once. The user was always given the choice of overriding the swap alerts and continuing to print with the current print cartridge.

Tests of this model produced both positive and negative results. On the positive side, users liked the control of selecting the cartridge type themselves, without getting swap alerts. Choosing Black/GreyScale to print a draft of a color document is one example of the control desired. On the negative side, this model required the users to go into the Page Setup dialog box before printing, which was often forgotten. Most subjects felt this extra step cumbersome and not "Macintosh-like."


ColorJet Page Setup

Paper: ● US Letter ○ US Legal ○ A4 Letter ○ Envelope ○ US Transparency ○ A4 Transparency

Print Cartridge: ● Black/GreyScale ○ Color ○ Auto Select

Special Effects: ☒ Font Substitution ☒ High Quality Fonts □ Precision Bitmaps

Reduce or Enlarge: □ %

Orientation

Fig. 5. A prototype HP Deskwriter C dialog box that was redesigned because user reactions during usability testing were mixed.

The final print cartridge swap/page setup model eliminates print cartridge choices from the Page Setup dialog box completely. Instead, the machine scans for color and displays an alert message when a mismatch between the cartridge and the document is found. With this model, the user need not go into the Page Setup dialog box at all. The final model is both easy to follow and gives users the control they want. Instead of having the user select which print cartridge to use, the driver is always in auto select mode. The driver scans the document for color. If none is found, it verifies installation of the black print cartridge in the printer. If the color cartridge is in the printer, the user is alerted to change the print cartridge. The same is done if a color document is being printed and the black cartridge is installed. If the user is printing a document with both black and color pages, the driver firsts prints the pages that can be printed by the installed print cartridge. When printing of those pages is complete, the driver alerts the user to swap the print cartridge, then continues printing the remaining pages. This minimizes the number of user interactions. The control provided by the former design is offered with this model by the Use current print cartridge only checkbox in the Print dialog box. When this box is checked, the driver will print all pages using the current print cartridge in the printer, and will not prompt the user for print cartridge swaps.

Print Modes

In general, print quality settings allow the user to trade print quality for print speed. Draft mode is fastest, and has the additional benefit of saving ink. Mode selection affects the resolution at which the page is rasterized and the timing and placement of the dots that are printed onto the page. The printer's firmware is responsible for managing the mechanism, printhead, and low-level dot control.

Imaging Resolution. Most of the time, HP DeskWriter C drivers operate at 300 dpi. For faster throughput at the expense of resolution, 150-dpi imaging is available as draft mode. Rasterizing at 150 dpi versus 300 dpi speeds up printing in several ways. First, the driver only has to draw objects at half the resolution. Second, at 150 dpi, it takes only one fourth as many bands to rasterize a page as it does at 300 dpi, because each band at 150 dpi represents four times as much page area as it does at 300 dpi (memory requirements grow as the square of the resolution). Third, the driver has one quarter the amount of data to halftone, compress, and transmit.

Dot Timing and Placement. The print cartridge contains three colors of ink: cyan, magenta, and yellow. For each primary color, sixteen nozzles are allocated. The nozzles for each primary are separated vertically (Fig. 6). This means cyan ink is always fired onto dry media. Magenta comes next, possibly overprinting cyan, and yellow follows.

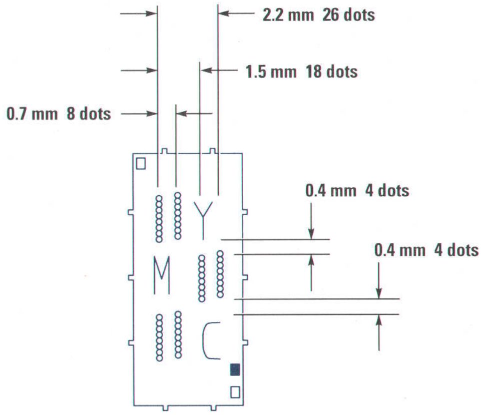


Fig. 6. HP DeskWriter C/DeskJet 500C printer orifice plate viewed from the paper.

In general, the printer can lay 16 rows of each color down at a time and then advance the paper 16/300 inch to lay down the next 16 rows of color. However, there are times when it is advantageous to lay the ink down more slowly while distributing the ink in a single raster row among several different nozzles in the printhead. This process is known as shingling (see Fig. 7). Using shingling, several separate print passes are used to lay the ink down, printing only a fraction of the dots in each raster row during each print pass. The paper is advanced slightly and then more of the pixels representing that row are laid down. The printer supports both 50% shingling and 25% shingling. 50% shingling lays down every other pixel in the row on the first pass and then the remaining pixels in the row on the second pass using a different nozzle of the print head. 50% shingling takes twice as many print passes to print a page and about 50% more time. It doesn't take twice as long, since every other dot is skipped during each pass, allowing the printer to increase the carriage speed without exceeding the maximum nozzle refire rate. Fig. 7 illustrates how the cyan ink will be printed with 50% shingling.

25% shingling is also available. It uses four times as many passes, laying down 25% of the ink in each raster row at a time.

Shingling provides several advantages. Printing each raster in several passes gives the ink a chance to dry before the adjacent dots are laid down onto the page. This is especially important when inks of two different colors are placed next to each other. Shingling allows the first color to partially dry and minimizes bleed (mixing) between the two colors. This is very important for printing on transparency media. Laying the ink down too quickly causes it to bleed and form puddles.

Shingling also distributes the printing of any single raster between two or more nozzles. This is useful for hiding inconsistencies between nozzles, such as a weak or missing nozzle, since it distributes the missing or weak dots among several rasters. A weak nozzle is not very noticeable when 25% shingling is used. Shingling also hides errors in paper feed accuracy in a similar manner.

When draft mechanical quality is selected, the printer will skip every other dot. This not only halves the amount of ink used to print the image, but also allows the printer to increase the carriage speed for the same reason it can when shingling.

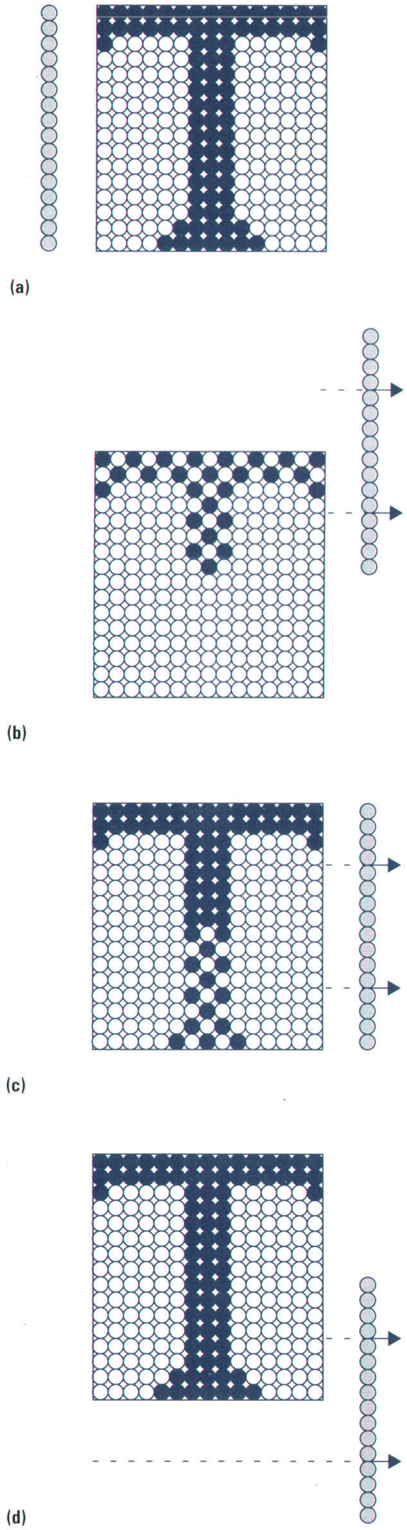


Fig. 7. Shingling is the process of laying down the ink dots in several passes, printing only a fraction of the dots in each raster row during each pass. (a) The 16 cyan nozzles spaced vertically 1/300 inch apart and a character to be printed. (b) During the first pass of the printhead, the first 8 cyan nozzles print half of the top part of the character. (c) During the second pass the first 8 cyan nozzles print half of the bottom part of the character while the second 8 cyan nozzles print the remaining dots in the top part of the character. (d) During the third pass of the printhead the second 8 cyan nozzles print the remaining dots in the bottom half of the character.

Resolution, shingling mode, and mechanical quality are mixed in various combinations to provide the user with three different quality modes for each of three types of media:

	Media Type		
Print Mode	**Plain Paper**	**HP Special Paper**	**Transparency**
Draft	150 dpi, no shingling, draft quality	150 dpi, no shingling, draft quality	Not supported
Normal	300 dpi, no shingling, normal quality	300 dpi, 50% shingling, normal quality	300 dpi, 50% shingling, normal quality
Best	300 dpi, 50% shingling, normal quality	300 dpi, 25% shingling, normal quality	300 dpi, 25% shingling, normal quality

Media. HP DeskWriter C and DeskJet 500 C printers are designed to provide good print quality on plain paper, special paper, and transparency media. The special paper and transparency media are specifically designed to work in the printers. They are available from Hewlett-Packard as CX JetSeries CutSheet Paper and LX JetSeries Transparency Film.

The ink performs differently on the three supported media types. When the user informs the driver which type of media has been selected, the driver takes this into account, and makes appropriate adjustments in the amounts of each ink deposited on the media.

Delivering high print quality on a wide range of plain papers is very difficult. As previously mentioned, print quality varies as a function of paper type and environmental conditions. The black print cartridge is the same one used by the HP DeskJet and DeskWriter printers. It is optimized to perform well over a wide range of plain papers. Similarly, the color print cartridge is designed to perform well on plain papers as well as on special paper. During development, print quality was tested on many different types of plain paper. From this large group, a small set of papers, each representing a significant subset of the universe of papers, was selected. Creating this manageable set of papers for initial evaluation of various inks, print cartridge architectures, and print modes proved valuable. Promising combinations could then be tested against the larger set to ensure robust performance.

Because we can control the design of the special paper, it is adjusted to accommodate the print cartridge. The special paper is coated on one side. The coating causes most of the colorants in the inks to be deposited near the surface of the paper after the inks dry. Concentrating the colorant near the surface increases the saturation of colors. The coating also reduces the sensitivity to environmental conditions. When using special paper, the user is less likely to need to adjust the intensity slider to accommodate environmental extremes.

The transparency medium consists of a plastic substrate and a special coating. The coating accepts the ink. Without the coating, ink would puddle up on the substrate and run together, forming large muddy pools. Like the special paper, the transparency is designed around the print cartridge.

Color Halftoning. Applications communicate with the driver through a stream of QuickDraw commands. These commands specify 24 bits of color for each 300-dpi pixel on the page. The printer can only place three bits of information at each 300-dpi print grid position—one bit for each primary ink. Color halftoning is the process that reduces the information from 24 to 3 bits per print position. By carefully controlling the placement of colored dots, myriad different colors are produced. Cyan, magenta, and yellow are halftoned independently. Cyan is reduced from 8 bits per pixel to 1 bit per pixel. Similar reductions occur for magenta and yellow.

Halftoning increases the effective color depth of the printer. The mechanism can put one of eight different combinations of the primary inks (cyan, magenta, yellow, red, green, blue, black, or white) on the paper at each 300-dpi print position. With halftoning, many more than eight colors can be produced. The increased color depth comes at the expense of spatial resolution. The halftoning techniques used in the HP DeskWriter C driver preserve 300-dpi edges. Spatially, the human visual system is much more sensitive to the edges of an object than to color shifts within the the object. Because the edges are preserved, the reduction in spatial color resolution is not offensive, and the increase in color depth allows complex images, such as photographs, to be reproduced well by the printer.

The user can choose between two color blending (halftoning) algorithms: *pattern* and *scatter*. Pattern is a dispersed ordered dither (Bayer's), while scatter uses a form of error diffusion.[1]

Pattern. The pattern algorithm is the default halftoning algorithm. It is a form of ordered dither. It does a good job with almost all types of data, but is best suited to simple graphics composed of large homogeneous regions. Computationally, it is much simpler than error diffusion. An 8 × 8 threshold matrix defines the halftoning pattern. Many different patterns could be used; the best pattern depends on the type of image being printed and on personal taste. To keep the user interface simple, only one pattern is offered: Bayer's dither. This pattern is very good at preserving fine detail. Preservation of detail is especially important when halftoning text that is not being printed at full intensity.

There are 64 cells in the 8 × 8 threshold matrix. Each cell contains an 8-bit threshold. The page is logically tiled with the matrix. Each pixel's 8-bit value is compared with its corresponding position in the matrix. If the pixel's value is greater than the threshold, a dot is fired at that position; otherwise, no dot is fired. This reduces the 8-bit information for the pixel to 1 bit (fire or don't). The process is repeated for all pixels on the page.

In an 8 × 8 area on the page, the number of dots fired can be anywhere from zero to 64. This means a total of 65 (counting none) different amounts of any primary color can be used to fill an area. The original data is 8-bit, representing 256 levels. Because only 65 patterns are used to represent 256 levels, visible contouring may occur when the specified amount of ink gradually varies over a large region of the page.

An example 4 × 4 threshold matrix is shown below. In this case, the pixel values would be scaled between 0 and 16 before halftoning. If a pixel's value were greater than the corresponding threshold matrix entry, that dot would be fired, otherwise it would not be fired.

4 × 4 Threshold Matrix

0	8	2	10
12	4	14	6
3	11	1	9
15	7	13	5

For example, assume a large homogeneous region were being halftoned, and the scaled values in the region were all 8. The resulting pattern of dots fired would be a checkerboard. This is because every other element in the threshold matrix is less than 8.

An obvious feature of the ordered dither is the geometric artifact that is visible when homogeneous regions are halftoned. At 300 dpi, the artifact is small and usually not objectionable. A shortcoming is the inability to represent 256 separate levels. This means a continuously varying gradient will be broken up into bands (or contours) of pixels that all map onto one of the 65 levels.

Scatter. For the scatter halftoning algorithm, a version of the Floyd-Steinberg error diffusion algorithm[1] is used. Unlike ordered dither, error diffusion is not restricted to 65 patterns. A pixel is examined, and if its 8-bit value is greater than 128, a dot is fired. The difference between the specified value and 255 is the error that is produced by putting down a full drop of ink. This error is diffused among four neighboring pixels, reducing their specified values slightly (see Fig. 8). If the original pixel's 8-bit value is less than 128, no ink is fired, and the error is simply the pixel's value. In this case, the distributed error increases the values of the neighboring pixels.

Pixels are processed from from left to right along each raster. Rasters are processed from top to bottom. The error from each pixel is broken up into four parts, which are distributed

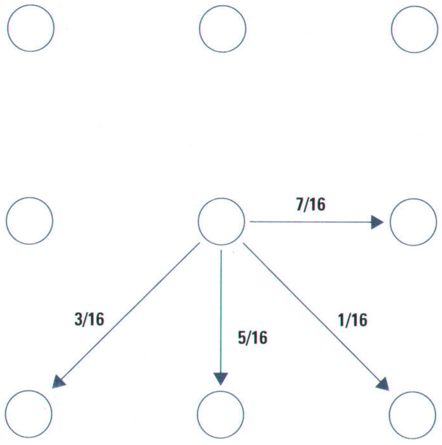


Fig. 8. Error diffusion to neighboring pixels in the scatter halftoning algorithm.

to neighboring pixels. Some noise is added to the error terms to break up artifacts that tend to appear in error diffused data. The average value of the noise is 0, so the image is not lightened or darkened.

Error diffusion does not suffer from contouring. Color gradients are printed as smoothly varying regions of increasing dot density.

Error diffusion can produce artifacts. These are most likely to be visible in large homogeneous regions. It is best suited for complex images like photographs and sophisticated presentation graphics. Hence, it makes a good compliment to the ordered dither algorithm. Error diffusion requires more computation than pattern dither, and can degrade throughput. Pattern dither was chosen as the default halftoning algorithm because it is faster, and is best suited for common types of output.

Grey Balancing

In theory, an equal mixture of cyan, magenta, and yellow inks would produce a neutral (white, grey, or black) color. In practice, this does not occur. The particular inks used by the HP DeskWriter C and DeskJet 500C printers, when mixed in equal proportions, typically produce a color with a slight greenish cast. To compensate for this effect, grey balancing is performed before halftoning.

Grey balancing reduces the amount of cyan used for neutral and near-neutral colors. Decreasing cyan increases the relative amounts of magenta and yellow. Magenta and yellow inks together make red, which is the opponent color of green. One can think of the cyan reduction as an increase in red, which compensates for the greenish cast. Unfortunately, reducing the amount of ink on the page makes the color lighter, so the cyan reduction must be balanced against the loss in darkness. Experimentation showed that a cyan:magenta:yellow ratio of 2:3:3 produces good neutral and black colors over a wide range of papers.

The adjustment is made by first computing the saturation of the color.[2] The following equation yields a value for S between 0 and 1:

$$S = \frac{(\max(c, m, y) - \min(c, m, y))}{\max(c, m, y).}$$

In this equation, c, m, and y represent the amounts of cyan, magenta, and yellow inks, respectively. Larger values of S indicate more saturated colors, while 0 indicates a neutral color.

The amount of cyan ink is adjusted based on S. If S is 0, cyan ink is reduced to two thirds of its original value. If S is 1, cyan is unaffected. Cyan ink is adjusted linearly between 100% and 67% for intermediate values of S.

Intensity Slider

Possibly the greatest challenge in creating a plain paper color printer is delivering high-quality output over a range of environmental conditions (temperature and humidity) and on a range of media, from copier paper to high-quality cotton bond. Two problems occur when fully saturated colors are printed on some types of paper, or at high humidities. Color bleed occurs when adjacent colors run into each other and mix. Ink can even bleed through the paper and appear on the back side. This can occur even if shingling is used. Bronzing occurs when too much ink is laid down on the paper and the dye sits on top of the paper fibers rather than soaking in. This overabundance of ink crystalizes on the surface and the crystals reflect light. This causes a shiny reflective surface that actually gets lighter as more ink is laid down.

As it turns out, when these problems arise, reducing the amount of ink used in printing the colors usually results in higher-quality output. Use too much ink and you get the problems described above; use too little, and the output looks dull and washed out.

Other factors affect the user's perception of quality and color accuracy, such as room lighting, computer display variations, personal taste, and the type of data being printed such as line art versus an image. Rather than attempt to characterize all of these factors, we supply a control that is analogous to the brightness control on a television set. This is called the intensity slider.

The intensity slider is a five-position slider available under the Options dialog box described above. It allows the user to control the intensity (saturation) of the colors on the page. This is done by controlling how much ink is used to create any particular color, which affects how saturated the color appears as well as other print quality factors described previously. At the slider's lowest setting, it can reduce the amount of ink used to generate a color by as much as 70%. This reduction is applied to the 24-bit data in the rasterized band just before halftoning. At the slider's highest setting, no reduction in ink volume occurs. The degree to which ink is reduced is a nonlinear function dependent on how much ink was specified in the first place. The percentage reduction is larger for more saturated colors, because these colors are most affected by problems associated with media type and environmental conditions.

Edge Enhancement

One unfortunate side effect of the intensity slider is that colors that would normally be a solid area fill of ink are now created with a dithered pattern of dots. This is even true for black printed with the color cartidge. At the minimum setting, about one third of the maximum posssible volume of ink is used. This may produce the best-quality black for area fills, but the edges of black characters will be rough in appearance and may have color halos.

The driver implements a simple edge enhancement algorithm. If a pixel is black and one or more of its four nearest neighbors (north, south, east, or west) are white, it is considered an edge pixel. Edge pixels are always printed with one drop each of cyan, magenta, and yellow. They are never depleted.

Fig. 9 shows how edges are enhanced.

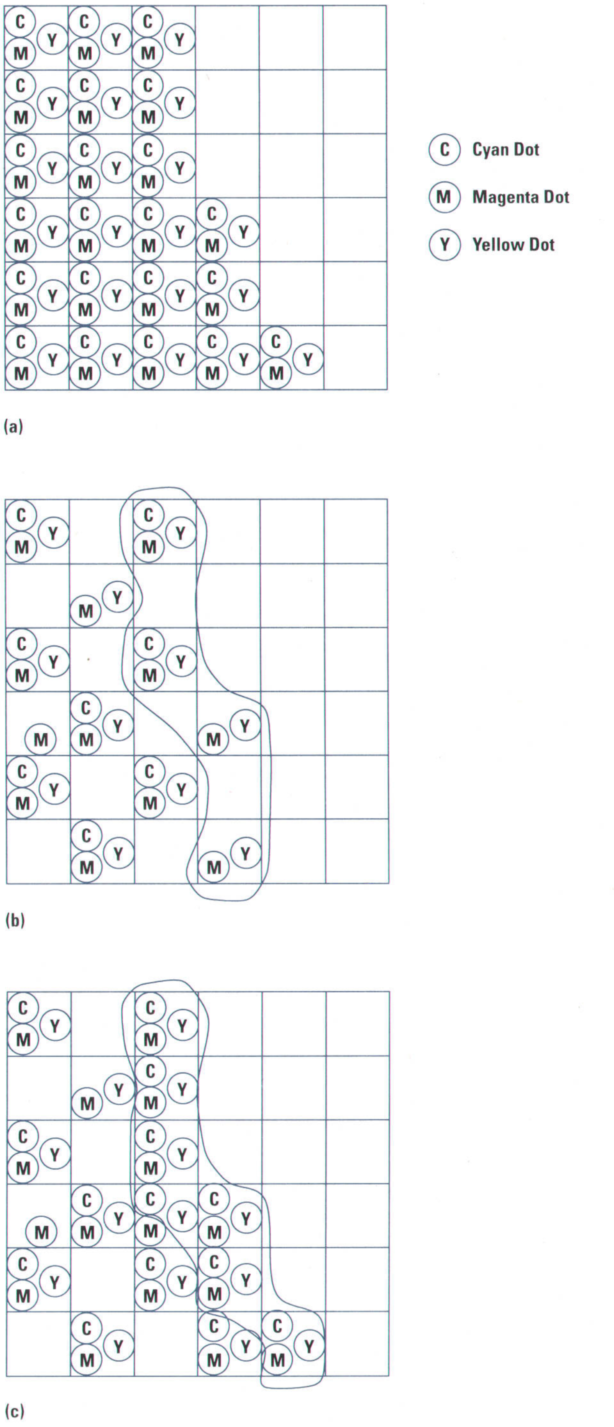


Fig. 9. Edge enhancement in the HP DeskWriter C printer. (a) The edge of a black character with no saturation reduction or color correction. (b) The edge of a black character with color correction and a low setting of the intensity slider. The center of the character will appear black, but the edge is rough because of missing pixels. Also, because unequal amounts of cyan, magenta, and yellow are used to produce black, the edge has a reddish hue or halo because of the absence of cyan dots. If this were the lower right portion of a character, the rightmost pixel of the serif would be totally missing. (c) With edge enhancement, every pixel that is adjacent to a white pixel is printed with cyan, magenta, and yellow dots. This creates a smooth edge without unwanted hues.

Selection of Printed Colors

The color halftoning techniques previously described allow the printer to produce myriad colors. The task of determining which colors to print seems simple, but turns out to be complex. At first glance, one might suggest putting colors on paper that exactly match the colors on the monitor. In practice, this produces surprising results. The situation is further complicated by the type of image printed. Processing a scanned image of a person's face and a bar chart in the same manner may not be a good idea.

The goal is to not surprise users when they receive the output. To this end, the HP DeskWriter C driver allows the user to choose between two color selection paradigms. The default is optimized for simple business graphics, but usually gives acceptable results with any type of output. The Complex Color Printing checkbox biases color selection for scanned photographs, sophisticated presentation graphics, and other complex images.

Chroma-Based Selection (Default). The default color selection model places a high priority on the chroma (colorfulness judged relative to a neutral area of similar lightness) of colors. On CRTs, chroma increases when the beam intensity of one or two of the electron guns is increased. This also produces a lighter color. On paper, chroma is increased by putting down more of one or two primary inks; this darkens the color. In the chroma-based selection scheme, priority is given to chroma over lightness. When printing simple business graphics, such as bar and pie charts, customers usually desire solid, high-chroma colors. The fact that printed colors appear significantly darker than monitor colors is not objectionable; it is usually desirable.

Colors are passed to the driver in RGB format. Red modulates the amount of cyan ink, green modulates magenta ink, and blue controls yellow. More red on the screen means less cyan on the paper. The cyan, magenta, and yellow inks can be thought of as "negative red," "negative blue," and "negative green," respectively.

A one-dimensional correction function is independently applied to each primary before it is halftoned. This function accounts for two factors. First, the perceived intensity of the monitor is not linearly related to electron gun voltage, and second, the perceived darkness on the paper is not linearly related to the amount of ink deposited in a given area.

When high-chroma colors are darkened by the driver, they are not as distinguishable as their counterparts on the monitor. This is not generally a problem, but some applications are capable of producing fairly complex business graphics. For example, in a three-dimensional bar chart, the top, front, and side of a bar can be colored with different tones of the same hue. This gives the illusion of depth on the screen. If the three faces of the bar don't have the correct tones on paper, the sensation of depth is reduced. In this case, appearance-based color selection may give superior results.

Appearance-Based Selection. Scanned photographs and other sophisticated images can be distorted objectionably by chroma-based color selection. In these images, differences in lightness often convey depth information. Artificially darkening high-chroma colors can make the output look unnatural. HP DeskWriter C users can instruct the driver to optimize color selection for these types of output.

A white patch on a monitor is not the same color as white paper. This can easily be seen by holding a piece of paper next to a CRT. White viewed on the monitor will (probably) have a bluish cast. The bluishness is not usually noticeable because the human visual system is remarkably adept at accommodating a wide variety of white points. Through green, grey, and rose colored sunglasses, snow looks white, grass looks green, and other objects appear as expected. This is the phenomenon of color constancy. The visual system adapts to the monitor's white. Only when another suitable reference white is placed in its proximity does the monitor appear bluish.

If colors on paper were selected to match the monitor exactly, monitor white would have to be printed as a pale blue tint. However, the eye has difficulty in accepting tinted paper as white. Because the HP DeskWriter C printer is a 300-dpi binary printer, pale blue must be created by a sparse scattering of relatively large blue dots. Monitor white is homogeneous, while the paper pale blue version would appear textured.

A better approach is to accept the fact that unmarked paper and the monitor appear white when considered independently. Colors are selected for printing so they appear, relative to unmarked paper, the same as the monitor's colors appear relative to the CRT's white point.

Gamut Issues. The gamut of a color device is the set of all colors it can reproduce. Typical monitors have a larger gamut than the HP DeskWriter C printer. This means they can display colors that cannot be reproduced by any combination of the printer's cyan, magenta, and yellow inks. A device's gamut can be modeled as a three-dimensional solid. The shape is irregular, but roughly resembles a lumpy football. The idea is to compress the monitor's gamut so that it fits inside the printer's gamut. Some compromises must be made when choosing a gamut compression algorithm, and although objectionable distortion is controllable, it cannot be eliminated altogether.

Characterize Monitor. To match colors on a monitor, it must be known how the colors on the monitor appear to the viewer. A spectroradiometer can be used to measure the colors produced by various intensities of red, green, and blue on a CRT. The device measures the amount of radiant energy emitted by the monitor as a function of visible wavelength. The system is well-behaved, and a reasonably simple model can be used to predict colors accurately once a few constants have been determined.

Apple's Macintosh color monitors all use Sony Trinitron CRTs. The 13-inch Apple monitor was chosen as the target monitor for the HP DeskWriter C driver. Other sizes of Trinitron CRTs are available from Apple and other vendors. Other brands of CRTs are also available. However, the 13-inch Trinitron heavily dominates our target customer's environment.

Characterize Printer. The HP DeskWriter C printer was characterized by printing a sample consisting of hundreds of small patches of different colors. Each patch contained a known percentage of cyan, magenta, and yellow dots. These were measured with a spectrophotometer. This instrument measures the amount of light reflected at various wavelengths across the visible spectrum. From this data, perceived colors can be calculated.

The HP DeskWriter C and DeskJet 500C printers are designed to be used with HP special paper, HP transparency media, and a wide variety of plain papers. The specified special paper was used to characterize the printer. Transparency color selection is always chroma-based, so rigorous color characterization was not required.

Two sets of samples were created, one using the pattern halftoning technique and one using the scatter technique. Each set was printed on special paper and plain paper, resulting in four sets of samples overall.

The choice of plain paper proved to be very challenging. Good print quality is needed across a wide selection of papers. Patches of eight colors were measured on over sixty varieties of plain paper. With this information in hand, a single "representative" plain paper was selected. This good-quality 25% cotton bond was used for the plain-paper characterization.

The selection of a single target plain paper simplified analysis. The user interface is straightforward; it allows the user a simple three-way selection: plain paper, special paper, or transparency. Although print quality may not be optimized for any given plain paper, it is well-controlled on almost all of them. When color accuracy is critical, HP special paper is recommended.

Acknowledgments

Our thank you list is endless, but the following people deserve individual recognition: Dave Parks, Mr. Martin O. Nicholes, Roopa Pathak, Damon Schaefer, and Dan Weeks.

References

1. R. Ulichney, *Digital Halftoning*, The MIT Press, 1987.
2. J. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer Graphics Principles and Practice*, 2nd Edition, Addison Wesley, 1987.

An Interactive User Interface for Material Requirements Planning

For planners and buyers in the manufacturing business environment, HP MRP Action Manager is an online, interactive tool that automates many of the traditional paper-intensive activities of material requirements planning.

by Alvina Y. Nishimoto, William J. Gray, and Barbara J. Williams

HP MRP Action Manager for NewWave is an enhancement to HP Materials Management (HP MM) software. HP MM provides an online materials management tool for production planners and buyers.[1] The HP MRP Action Manager operates within the HP NewWave environment[2] to provide an easy-to-use, interactive user interface between a personal computer and the HP 3000 computer.

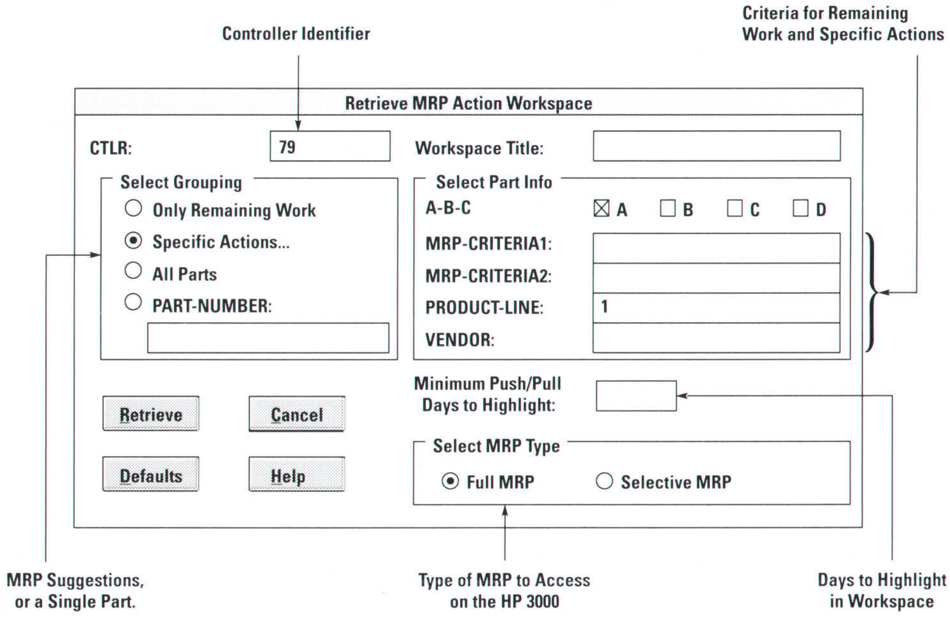
In a typical manufacturing environment, planners and buyers receive MRP (materials requirements planning) reports after running a regularly scheduled MRP program. These reports, called action reports, show suggested actions for work orders and purchase orders to meet future demand for inventory. The suggested actions include changes to pull in and push out dates,* cancellation of existing orders, and suggestions for new orders.

Planners and buyers typically have to wade through mountains of paper reports to determine the suggestions they would like to act on. Then, they make any changes on paper and manually enter these changes into an MRP package such as HP MM.

HP MRP Action Manager software automates this process by providing the following features:

- Dynamic selection control. Each planner or buyer reviews the entire action report or a subset of the action report, filtering the information displayed by selecting some combination of MRP actions (pull in, push out, etc.), part numbers, user-defined items, product line, and vendor. This feature eliminates the unnecessary printing of MRP reports, resulting in reduced printing costs and increased time savings. Fig. 1 shows the Action Manager window for selecting MRP information. The selections shown in the window indicate that the user is selecting parts for controller 79 that have the ABC code A in product line 1. Fig. 2 shows the window for selecting a specific action to be performed on orders. This window appears when the Specific Actions... command is selected from the selection window shown in Fig. 1.
- Online review and approval of MRP actions. Planners and buyers can approve MRP actions with or without changes, cancel MRP actions, or hold MRP actions for later review.

* Push out indicates that an open order's due date should be rescheduled farther in the future, and pull in indicates that an open order's due date should be rescheduled with an earlier due date.



Controller Identifier

Criteria for Remaining Work and Specific Actions

Retrieve MRP Action Workspace

CTLR: 79

Workspace Title:

Select Grouping

○ Only Remaining Work

● Specific Actions...

○ All Parts

○ PART-NUMBER:

Select Part Info

A-B-C ☒ A □ B □ C □ D

MRP-CRITERIA1:

MRP-CRITERIA2:

PRODUCT-LINE: 1

VENDOR:

Minimum Push/Pull Days to Highlight:

Select MRP Type

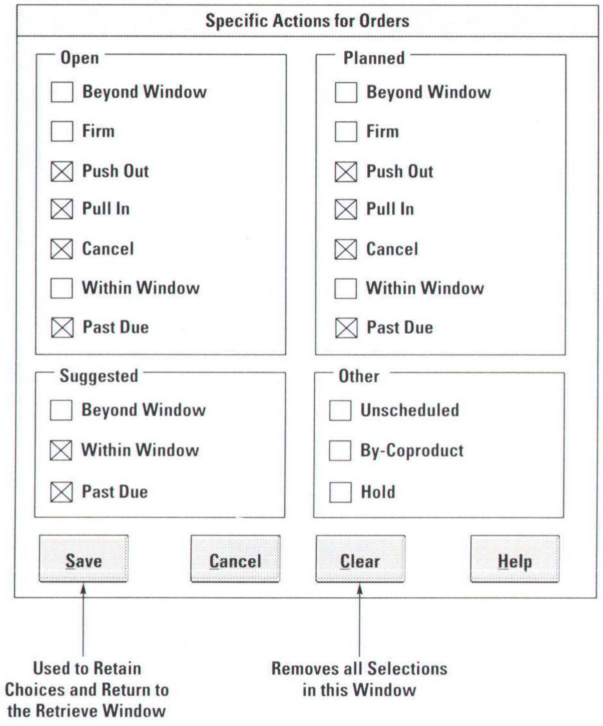
● Full MRP ○ Selective MRP

MRP Suggestions, or a Single Part.

Type of MRP to Access on the HP 3000

Days to Highlight in Workspace

Fig. 1. An HP MRP Action Manager window for selecting MRP information.



Specific Actions for Orders

Open
- ☐ Beyond Window
- ☐ Firm
- ☒ Push Out
- ☒ Pull In
- ☒ Cancel
- ☐ Within Window
- ☒ Past Due

Planned
- ☐ Beyond Window
- ☐ Firm
- ☒ Push Out
- ☒ Pull In
- ☒ Cancel
- ☐ Within Window
- ☒ Past Due

Suggested
- ☐ Beyond Window
- ☒ Within Window
- ☒ Past Due

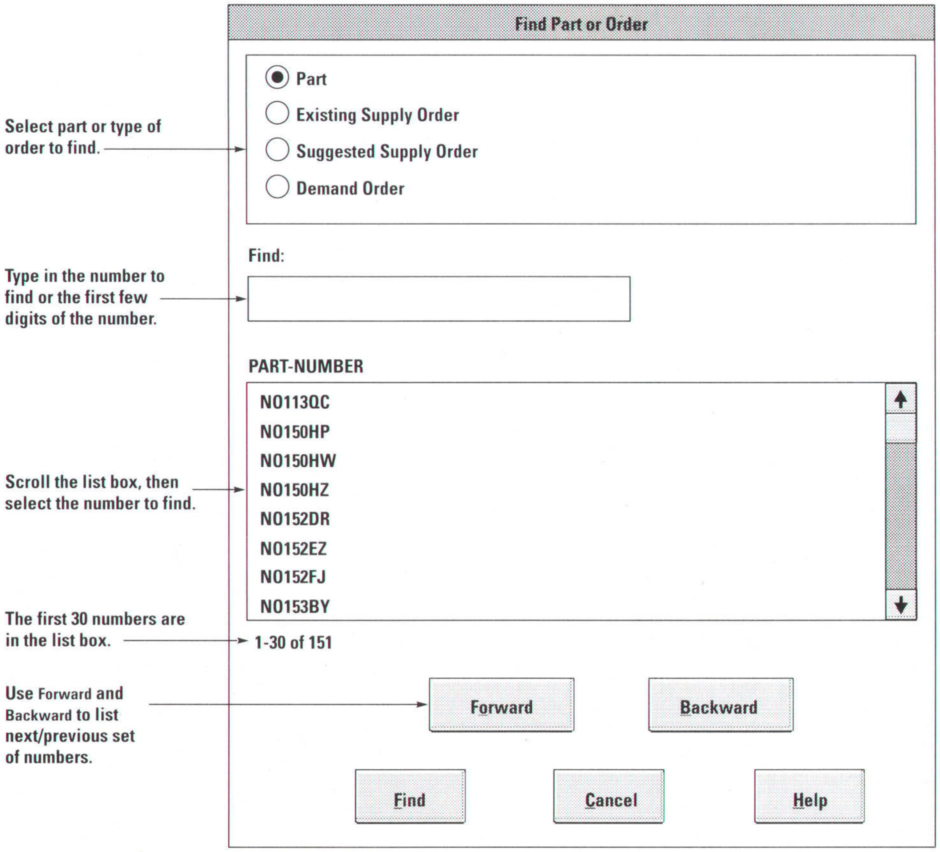
Other
- ☐ Unscheduled
- ☐ By-Coproduct
- ☐ Hold

Used to Retain Choices and Return to the Retrieve Window

Removes all Selections in this Window

Fig. 2. An HP MRP Action Manager window for selecting a specific action to be performed on orders.

- Online recalculation of projected inventory. Recalculation of projected inventory occurs when order quantities and dates change. This interactive "what-if" analysis immediately shows the planner the impact of changes on inventory levels. If the impact is not desirable, the planner can reset the changed selections to the original values suggested by MRP Action Manager.
- Automated order additions and changes. Order additions and changes are transferred to HP MM for automatic update through the batch interface. This feature eliminates the need to reenter the data manually through HP MM online data-entry screens.
- Direct part or order find. Using wildcard searches and selection windows, planners identify and quickly display specific parts or orders (Fig. 3).
- Online order pegging. By using online pegging, the planner can examine the upward and downward links in a bill of material, level by level, to find out where the demand is coming from that creates a requirement for an order, or to see the impact a supply order has on lower-level components.
- Workspace comments. The planner can create comments in the MRP action workspace reports to annotate the decision-making process.
- Optimized system resources. HP MRP Action Manager uses a cooperative client/server interaction between PCs and the HP 3000 computer for optimum use of system resources.
- Ease of use. Action Manager has a graphical user interface that takes advantage of the flexible features of the HP NewWave environment and provides online help with browsing and searching capabilities (see Fig. 4).



Find Part or Order

- ● Part
- ○ Existing Supply Order
- ○ Suggested Supply Order
- ○ Demand Order

Find:

PART-NUMBER

N0113QC
N0150HP
N0150HW
N0150HZ
N0152DR
N0152EZ
N0152FJ
N0153BY

1-30 of 151

Select part or type of order to find.

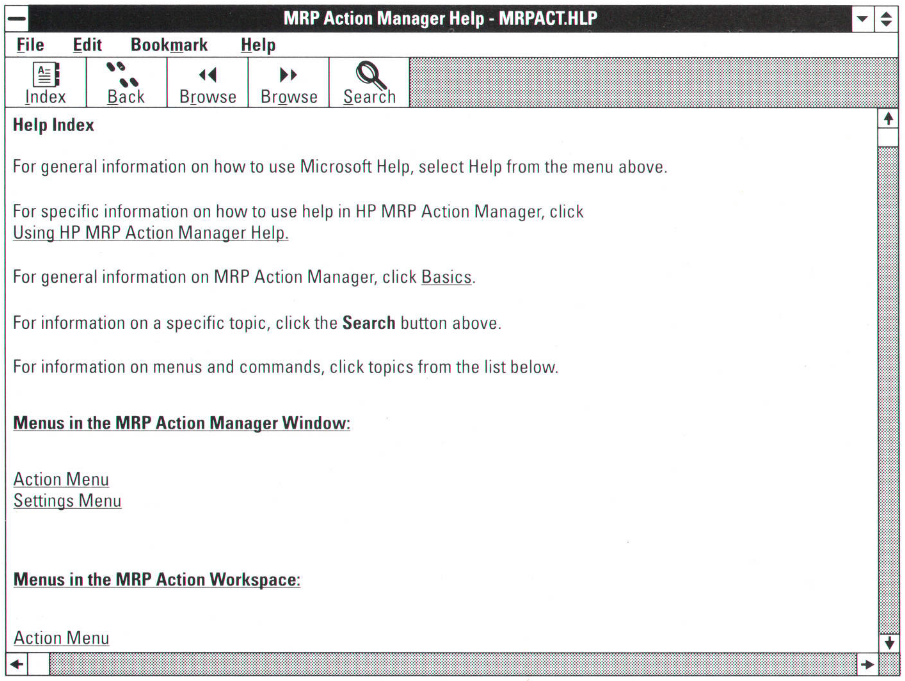
Type in the number to find or the first few digits of the number.

Scroll the list box, then select the number to find.

The first 30 numbers are in the list box.

Use Forward and Backward to list next/previous set of numbers.

Fig. 3. The Action Manager window for quickly finding an order or a part.



Help Index

For general information on how to use Microsoft Help, select Help from the menu above.

For specific information on how to use help in HP MRP Action Manager, click Using HP MRP Action Manager Help.

For general information on MRP Action Manager, click Basics.

For information on a specific topic, click the **Search** button above.

For information on menus and commands, click topics from the list below.

Menus in the MRP Action Manager Window:

Action Menu
Settings Menu

Menus in the MRP Action Workspace:

Action Menu

Fig. 4. A help window in HP MRP Action Manager, which uses the NewWave help facility.

With HP MRP Action Manager, the planner can analyze MRP data and execute MRP suggestions more quickly and effectively than by using the traditional printed reports and manually entering data into HP MM. The process a planner would use with the HP MRP Action Manager might involve:

- Selecting subsets of MRP output from the NewWave interface on the PC
- Analyzing suggested changes and new orders
- Taking advantage of NewWave's cut and paste functionality to incorporate portions of the MRP report into memos and spreadsheets
- Forwarding necessary order changes and additions to HP MM for automatic processing.

Action Manager Architecture

The architecture of the HP MRP Action Manager is based on a client/server model. The client (a PC running NewWave) communicates over the network with the server (an HP 3000 running the MPE/iX operating system). This architecture is shown in Fig. 5.

PC Architecture. Operating in the NewWave environment allows Action Manager to provide a user with all of the benefits of the NewWave user interface. A planner or buyer can easily perform the tasks necessary to complete an MRP job and get access to other NewWave applications such as NewWave mail and object storage.

The PC network requirements are HP ThinLan, HP StarLan 10, or HP EtherTwist and any Microsoft® Windows supported networking package. Because of the amount of data and the transmission rate, serial support is not provided. Communication between the PC and the HP 3000 is handled by a set of routines provided by an HP client/server developer's kit called HP Cooperative Services. HP Cooperative Services is a package of library modules and executable programs that provide the PC application developer with the tools necessary to write PC-to-HP 3000 client/server applications easily. The HP Cooperative Services developer's kit includes intrinsics for connection and session management, remote HP Image database management system calls, remote MPE calls, and user-defined remote procedure calls.

The HP Cooperative Services routines were chosen because they required the least amount of coding by the HP MRP Action Manager development team. They also provide transparent access* to the network, and an interface to the HP Image database on the HP 3000. Since these routines are also used in other HP products that require high-speed network transfers, we felt that the performance of these routines would already be tuned for the HP 3000. Not having to code the networking calls saved a tremendous amount of time and lowered the risk of running into technical problems later.

HP 3000 Architecture. The HP Cooperative Services calls embedded in the HP MRP Action Manager allow it to access HP 3000 files and databases. Furthermore, remote procedure calls from the PC to MPE/iX routines enable the HP 3000 to accomplish the heavy workloads and free the PC for other work while a retrieval takes place.

The Action Manager software on the HP 3000 uses the application customizer and application monitor software systems.[3,4] The application customizer provides a method for customers to tailor HP MM to individual needs and the application monitor automates many of the day-to-day administrative functions usually performed by an operations staff. These customizable technology software tools[5] implement much of the detail work involved in opening databases and files, handling terminals, and updating screens.

The HP MRP Action Manager architecture is designed to do most of the processing on the HP 3000 to take advantage of the processing power of that machine. The PC provides the planner with a graphical user interface and online MRP calculations for immediate feedback on the actions taken.

Microsoft is a U.S. registered trademark of Microsoft Corp.

* No need for the application to be concerned about the type of network installed.

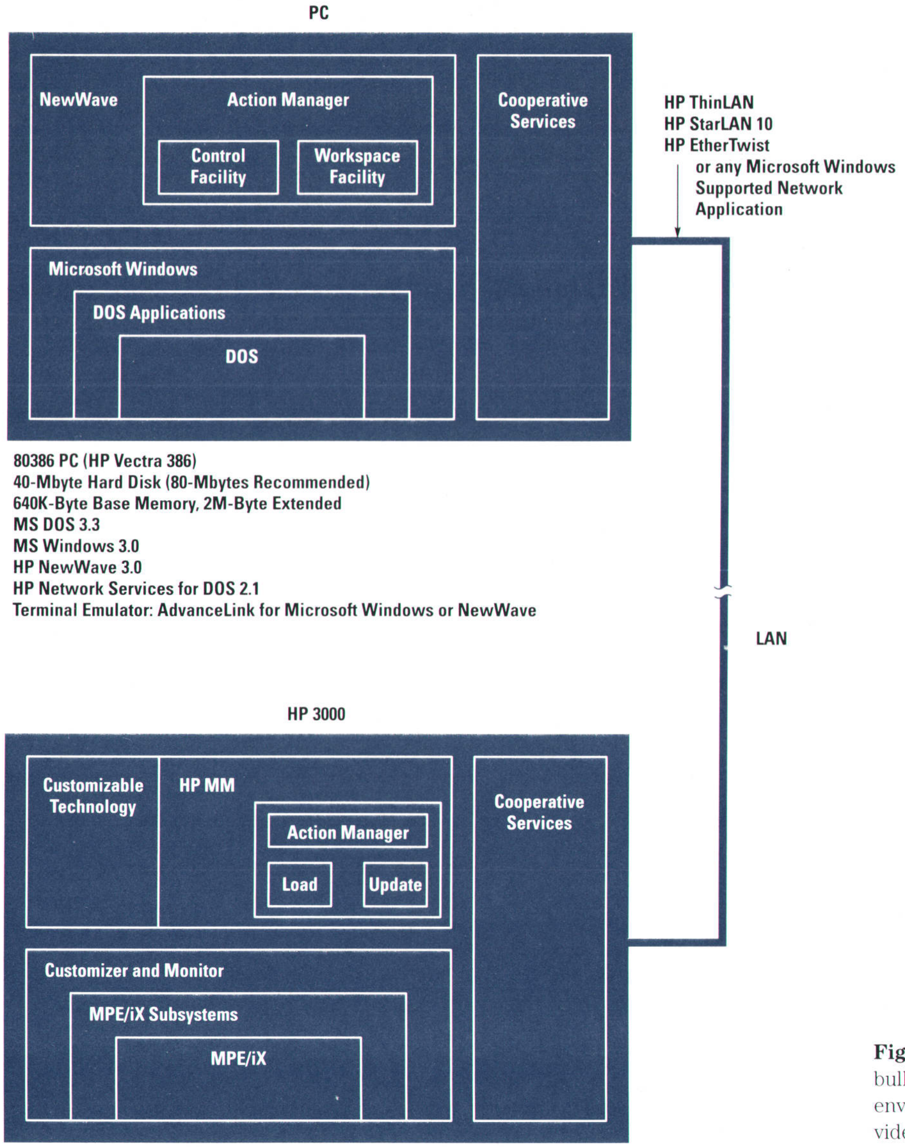


Fig. 5. Action Manager architecture. The bulk of the processing is done in the HP 3000 environment and the user interface is provided by the PC.

This organization became especially important when dealing with the data customization feature on the HP 3000. All formatting of the MRP workspace is done on the HP 3000 to avoid having to code for the PC all the same customizer intrinsics used on the HP 3000. The MRP load program on the HP 3000 creates layout information about the workspace so that the PC knows where the information it needs is stored, and when altered, how the planner wants the information displayed on the PC. Other information such as significant literals, search item names, and field names are also created by the MRP load program for the PC to access.

Action Manager Tasks

The following sequence describes Action Manager's tasks and data flows between the PC and HP 3000 environments. The sequence numbers correspond to the numbers in Fig. 6.

1. An MRP batch job is run on the HP 3000. The job creates MRP output files, and if requested, generates MRP reports. The MRP output files contain MRP suggestions, actions, requirements, and part information.

2. The HP MRP Action Manager load program reads the MRP output files, strips the MRP output, places statistics and indexes in the MRP database, and creates a separate report file for each controller identifier (planner or buyer).

3. When the planner activates the Action Manager from the NewWave desktop, the settings, the controller identifier, and the selection criteria are defined. The creation of the Action Manager workspace is also initiated.

4. The Action Manager control facility uses the MRP database to identify the parts specified in step 3, and then retrieves corresponding action report data from the action report files on the HP 3000.

5. The control facility formats the workspace data and uses the NewWave object management facility[6] to create a new Action Manager workspace object.

6. With the new workspace, the planner reviews, changes, and approves MRP suggestions. The NewWave desktop is used to switch back and forth between HP MM screens and workspace windows so that the user has access to all the relevant data for faster decision making.

7. The planner initiates batch transfer of all actions from the currently active workspace to HP MM on the HP 3000. The

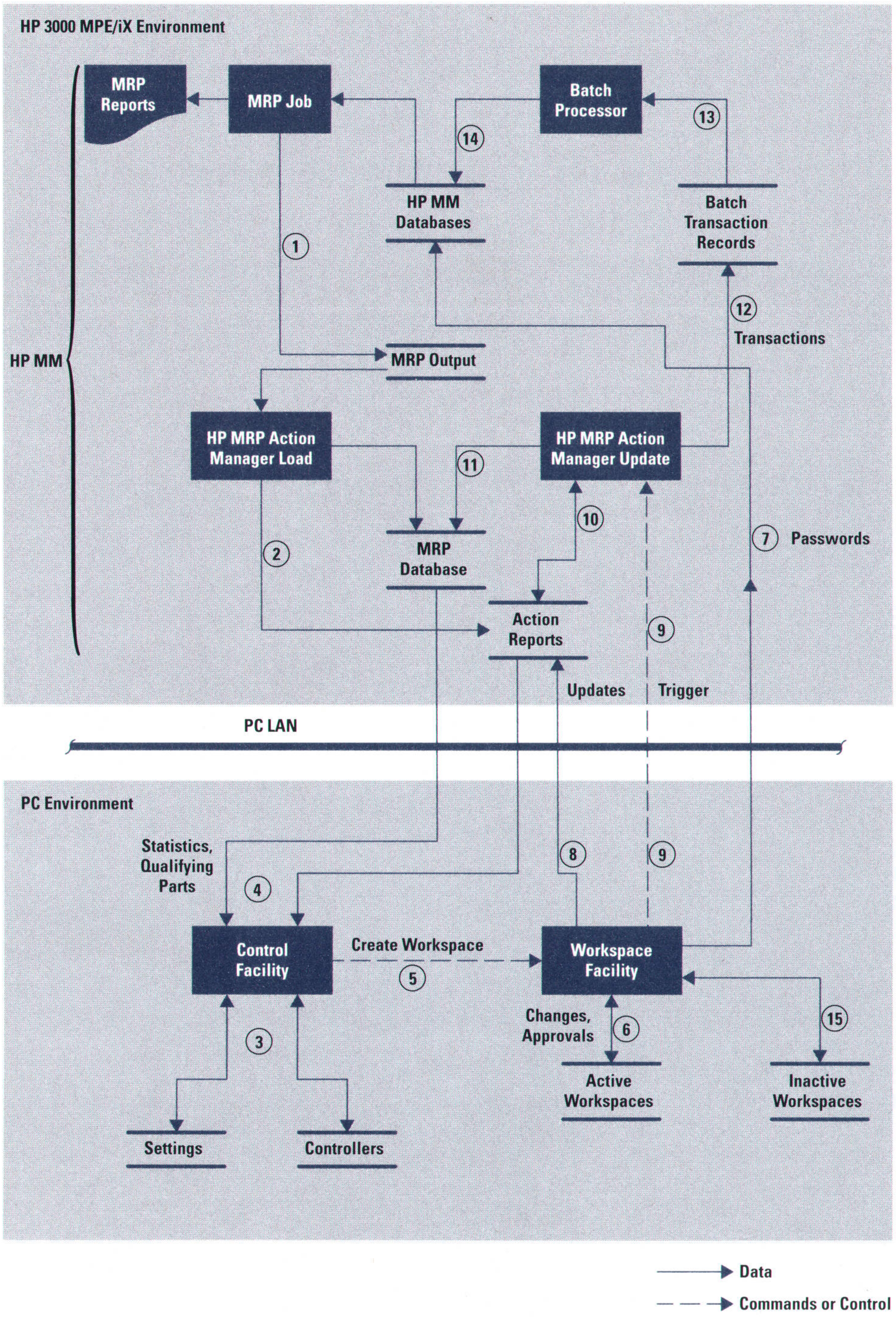


Fig. 6. Action Manager tasks and data flows between the PC and the HP 3000 environment. Note that Action Manager runs as part of HP MM on the HP 3000 system.

planner is prompted for the controller password to authorize the transfer, and the password is validated against the HP MM database.

8. The Action Manager updates the action report file on the HP 3000 with the marked actions (e.g., pull in, push out).

9. The Action Manager completes the transfer, then triggers the Action Manager update program on the HP 3000.

10, 11, 12. The Action Manager update program scans the action report file for approved changes and additions that need processing, updates the MRP database with changes, then creates batch transaction records and sends them to the HP MM continuous background batch processor.

13, 14. The background batch processor processes the batch transactions, completing the HP MM updates.

15. The planner archives the MRP action workspace. The data in the workspace can still be reviewed, but changes and approvals can no longer be done.

Development Environment

For the HP MRP Action Manager software running on the PC, two software development tools were used to generate the early prototypes. The first tool, CASE:W by CaseWorks Inc., was used to create the main windows (menus and menu items) and to generate the Microsoft Windows code. The code generated is highly structured and well-commented, which made it easy to determine where to insert the functionality required to make Action Manager work. The other tool, WaveMaker, which is an internal development tool, was used to read the CASE:W generated code and insert the calls to the NewWave OMF (object management facility), API (application program interface), agent, and help facilities.[2]

With the use of these tools we were able to create working prototypes very quickly and present them to our development partners and project team members for feedback. When the visual and usability aspects of Action Manager were refined, traditional methods were employed to complete the PC-side development. The software revision controller on the HP 3000 was used to handle revision control.

For the HP 3000 development environment, the application customizer and the application monitor mentioned earlier were used. These two software tools are used by application designers to develop generalized software systems for HP 3000 systems. The application programmer uses the application customizer intrinsics to manipulate data, since crucial information about data formats and fields is not known until run time. (Users can modify data formats and fields.)

While the application customizer intrinsics make customizable applications more manageable, the parameter lists and code to load the information can make coding more difficult. A layer of routines on top of the application customizer intrinsics, called customizer technology intrinsics, are used by the programmer to make it easier to develop the application code.

Action Manager developers also took advantage of reusable code originally developed for the HP JIT (just in time) product.[7] HP JIT placed many of the common calls to the underlying customizer technology tools into utility subroutines. Being able to develop the HP 3000 portion of the Action Manager with reusable code helped us to be more productive and ensure better quality, consistency, and standardization in the code.

HP MRP Action Manager Project Management

When the investigation for the HP MRP Action Manager project started, we discovered that in addition to the challenges presented by the technical issues, we also had to define clearly the goals of the product to complete the project on time. As is typical of many projects these days, these requirements meant that we had to make some changes to our traditional development process.

Purpose and Direction

To ensure that we had a stated goal for the project and at the same time addressed the technical issues facing us, during the investigation phase the development team focused on technical issues and the project manager worked with the team to develop a purpose and direction statement. This statement documented the key aspects that were critical for our project's success. The following are the key areas identified and some of the subsequent actions we took to address them.

- The product must meet the customers' needs as simply as possible. We wanted to ensure that the functionality of the product would encourage the user to prefer to use HP MRP Action Manager instead of doing the work manually. We believed that if some aspect of the user's job was more difficult or cumbersome with Action Manager, the user would abandon using the product and revert to doing the job manually.
- The features in Action Manager must satisfy the top requests from the installed base of HP MM customers and HP field representatives. We worked to ensure that the features we implemented in Action Manager truly satisfied customers' requests. We also focused more attention on these features, since we knew that our customers had some particular interest in them.
- There must be minimal impact to the existing HP MM product. In past projects, we found that new enhancements (software changes) to HP MM often introduced defects to the existing software. We agreed to focus on minimizing changes to HP MM to eliminate the support headaches that often followed a software enhancement release.
- The product should support HP's cooperative computing strategy. As corporate citizens, we felt it was important to be consistent with the direction HP was heading in software. We formulated the product idea to take advantage of client/server technology and to use NewWave as a basis.

The process of developing the statement was more critical than the statement itself. It was in our discussions about our purpose and direction that we came to a common understanding of our goals.

One simple action was neglected, however. We failed to share our purpose and direction goals with other project teams in the organization. Their understanding of our project's goals was very different. This resulted in some issues later, especially during the integration phase when our project and other projects became interdependent (because we were all completing our software for one combined release).

Customer Focus and Understanding

Some of the team members were experts on the targeted end users of Action Manager. Others had little or no knowledge of the customers' functionality needs (but had technology expertise instead). Throughout the project, we took advantage of every possible opportunity to show the latest prototype or actual product software to a customer, sales representative, or field engineer. This constant exposure to customers served two very important purposes. First, it continued to educate every team member so that we could make better product decisions. Second, it gave us a sense of whether we were on the right track with our product. Initially, we had to make some major adjustments in our ideas for implementing the product, but these changes became less and less significant as the project progressed.

Managing Dependencies

This project required high-quality software components from other organizations within HP. We were dependent on HP NewWave, HP Cooperative Services, NetIPC, NS services, and the Microsoft Windows Toolkit. When we started the project, NewWave 2.1 was the latest version available. We had to convert to NewWave 3.0 in the middle of our project since it was key for us to release with that version. We also had to transition from NetIPC networking software to NS services. Thus, we were very dependent on other organizations to meet our schedules. We were able to tap into the expertise of engineers in these other organizations and resolve issues associated with these dependencies when they arose.

Development Process

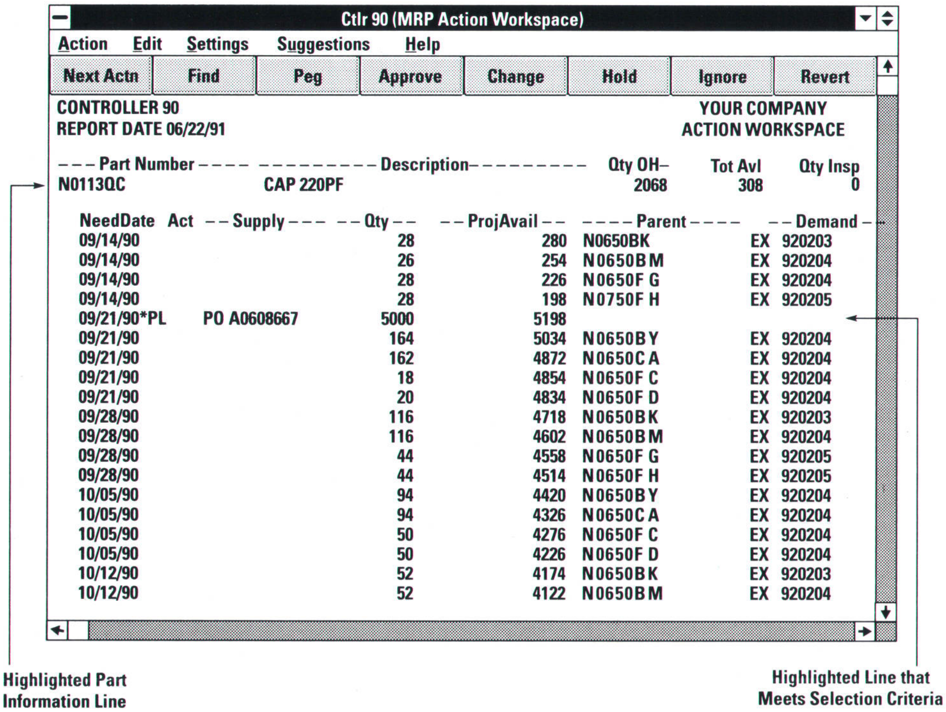
Since Microsoft Windows and the NewWave environment were new technologies for us, we did not use every step of the traditional software development process to meet the objective of producing a high-quality product in a short time.

Development Partners. Throughout the development of the product, we used development partners (both internal and external to HP) to help in the design of the product. For example, HP MM internal and external customers provided feedback on functionality, usability, and reliability, and current product engineers and support engineers provided feedback on maintainability and supportability.

Prototyping. Each prototype completed was presented to our development partners for review. Because of the tools mentioned above, we were able to make changes to the software while the development partners were sitting right there with us. The formal presentations of the product were supplemented with many informal presentations between each development cycle. The most important aspect of this prototyping phase was that we were never afraid to throw away code to incorporate the requested functionality.

Key Features. Many of the key features incorporated into Action Manager were based on user feedback and our observations of our partners' working environments. Some of these features included:

- Filtering action reports to allow planners to see those parts that are the highest-priority. We observed our partners using a variety of methods to reduce the mountains of paper produced for a traditional MRP report. Most often they had to resort to manual methods, thumbing through listings and paper-clipping the most important parts of the report. MRP Action Manager provides planners, through their chosen filters, with the ability to see only those parts of the MRP report that interest them.
- Adding usability improvements in choosing filtering criteria. Once we gave our partners the ability to choose their own filters, we found that the dialog boxes used for selecting these filters were cumbersome and confusing. We originally presented the dialog boxes in Figs. 1 and 2 as one dialog box. This presentation created a lot of confusion for users. Subsequent revisions attempted to eliminate choices that the planner rarely used. This process proved fruitless, since almost every combination was of some importance. After many iterations and usability evaluations, we chose the final set of dialog boxes in which the grouping titled Only Remaining Work represented the planner's most likely choice, but the grouping titled Specific Actions... allowed the planner to customize this grouping. By changing to two dialog boxes, we eliminated the clutter of the one combined dialog box, while allowing everyone to select the final combination.
- Highlighting each planner's highest-priority actions. We observed that every printed action report was highlighted with those actions most important to that planner. In the product, we filter the reports based on the planner's selections and highlight only those actions on the report that they have chosen (see Fig. 7).
- Providing comments to annotate decisions. Every planner we visited had an action report with notes scribbled in the margins. To provide this ability in the Action Manager, we added the capability for users to add annotations with each key decision they make on the action report.
- Providing the ability to archive an action report. Planners often commented on how they kept past action reports nearby for easy reference in case they had a question about a past decision they had made. We provided in Action Manager the ability to archive a workspace with the annotations so that planners could easily reference those workspaces and annotations when needed.


Ctlr 90 (MRP Action Workspace)
Action
Edit
Settings
Suggestions
Help
Next Actn
Find
Peg
Approve
Change
Hold
Ignore
Revert
Highlighted Part Information Line
Highlighted Line that Meets Selection Criteria

```
CONTROLLER 90                                                        YOUR COMPANY
REPORT DATE 06/22/91                                              ACTION WORKSPACE

---Part Number----  ----------Description---------   Qty OH-     Tot Avl    Qty Insp
N0113QC             CAP 220PF                           2068         308           0

  NeedDate   Act  --Supply---   --Qty--   --ProjAvail--   ----Parent----    --Demand--
  09/14/90                           28             280   N0650BK          EX  920203
  09/14/90                           26             254   N0650BM          EX  920204
  09/14/90                           28             226   N0650F G         EX  920204
  09/14/90                           28             198   N0750F H         EX  920205
  09/21/90*PL     PO A0608667      5000            5198
  09/21/90                          164            5034   N0650BY          EX  920204
  09/21/90                          162            4872   N0650C A         EX  920204
  09/21/90                           18            4854   N0650F C         EX  920204
  09/21/90                           20            4834   N0650F D         EX  920204
  09/28/90                          116            4718   N0650BK          EX  920203
  09/28/90                          116            4602   N0650BM          EX  920204
  09/28/90                           44            4558   N0650F G         EX  920205
  09/28/90                           44            4514   N0650F H         EX  920205
  10/05/90                           94            4420   N0650BY          EX  920204
  10/05/90                           94            4326   N0650C A         EX  920204
  10/05/90                           50            4276   N0650F C         EX  920204
  10/05/90                           50            4226   N0650F D         EX  920204
  10/12/90                           52            4174   N0650BK          EX  920203
  10/12/90                           52            4122   N0650BM          EX  920204
```

Fig. 7. Highlighting items that meet selectiion criteria.

Testing. Throughout the development of the product, we used formal design reviews, code inspections, and testing to contribute to the overall correctness of the code in each module. In the area of testing the product, we used a combination of formal, ad hoc, and usability testing.

Developed by an engineer familiar with the design of the product, the formal testing process included test scenarios with documented expected results. Engineers tested those modules that they had not developed. This strategy revealed defects that the author of a module, because of familiarity with the code, would often miss.

Ad hoc and usability tests were informal tests done by an engineer not familiar with the design of the product but very familiar with MRP and the planner's job. In many cases, ad hoc testing proved more effective than formal testing in finding defects in the product. With little knowledge of the internals of the product, the ad hoc tester could concentrate on testing what a planner expected from the product without being biased because of familiarity with the product design.

Acknowledgments

We would like to thank Mike Levy, the R&D section manager on this project. We would also like to thank product team members Rosario Delatorre, Gary Erickson, Bob Garland, Pamela Hinz, Bill Manasco, Dorota Ornatkiewicz, Michelle Petersen, Dave Tucker, and Mike Winter. Special thanks to Frank Blakely for his assistance with HP Cooperative Services. Our thanks to the many HP divisions and customers who gave us their active participation and feedback in the development of this product.

References

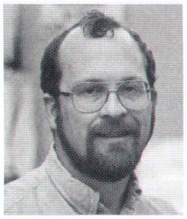
1. N. C. Federman and R. M. Steiner, "An Interactive Material Planning and Control System for Manufacturing Companies," *Hewlett-Packard Journal*, Vol. 32, no. 4, April 1981, pp. 3-12.
2. *Hewlett-Packard Journal*, Vol. 40, no. 4, August 1989, pp. 6-64.
3. L. E. Winston, "A Novel Approach to Computer Application System Design and Implementation," *Hewlett-Packard Journal*, Vol. 32, no. 4, April 1981, pp. 13-18.
4. B. D. Kurtz, "Automating Application System Operation and Control," *ibid*, pp. 19-22.
5. J. L. Malin and I. Bunton, "HP Maintenance Management: A New Approach to Software Customer Solutions," *Hewlett-Packard Journal*, Vol. 36, no. 3, March 1985, pp. 4-10.
6. J. A. Dysart, "The NewWave Object Management Facility," *Hewlett-Packard Journal*, Vol. 40, no. 4, August 1989, pp. 17-23.
7. R. K. Bhargava, et al, "New Methods for Software Development: System for Just-in-Time Manufacturing," *Hewlett-Packard Journal*, Vol. 37, no. 6, June 1986, pp. 11-17.

Authors

August 1992

6 Series 700 Workstations

Andrew J. DeBaets



Andy DeBaets was project manager for the processor board of the HP 9000 Model 720 computer and for the Model 720/730 system I/O board. He received his BS degree in electrical engineering and computer science from the University of California at Berkeley in 1979 and joined HP's General Systems Division in 1980. He has served as manufacturing engineer for the HP 125 computer, design engineer for the HP 120 computer, I/O design engineer for the HP 150 computer, and project manager for the power supply of the HP 9000 Model 815 computer. A native of Alameda, California, he is married, has one child, and enjoys golf, Vietnamese food, Peter Weir movies, and watching his son grow.

Kathleen M. Wheeler

Kathy Wheeler is an R&D project manager with HP's Entry Systems Division. She joined HP's Personal Computer Group Operation in 1985. She has been a software designer, a firmware designer for PA-RISC workstations, project manager for the firmware of the HP 9000 Models 815 and 808 computers, project manager for the firmware and processor hardware of the HP 9000 Models 720 and 730 computers, and program manager for the Model 730. She received her BS degree in electrical engineering and computer science from the University of California at Berkeley in 1979. Before coming to HP she was a hardware designer for Floating Point Systems and NCR and a firmware designer for GRiD Systems Corp. Kathy was born in Wallace, Idaho. She is married and has two children. Her interests include photography, reading, wine tasting, golf, and running.

12 VLSI Circuits

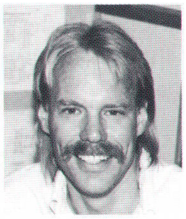
Mark A. Forsyth

Mark Forsyth was born in Minnetonka, Minnesota and attended the University of Minnesota, graduating with a BSEE degree in 1978. He joined HP's Calculator Products Division that same year. Now a project manager in the Systems Technology Division engineering systems lab, he has worked on HP's first 32-bit VLSI CPU chip, four generations of PA-RISC VLSI designs, and the PCX-S CPU chip and cache memory design for the HP 9000 Series 700 computers. He has coauthored several papers on CPU and computer design and is named as an inventor on three pending patents on RISC CPU design. He's a member of the IEEE. Mark's interests include flyfishing and cross-country skiing. He is married and has two children.

Craig A. Gleason

Craig Gleason is an engineer/scientist at HP's Systems Technology Division, specializing in CMOS I/O circuit design and cache memory timing design. With HP since 1984, he has contributed to the design of NMOS III and CMOS26 PA-RISC chipsets and to the VLSI CPU and cache memory system designs for the HP 9000 Series 700 computers. His work has resulted in pending patents for a CMOS output driver and cache memory timing. Born in Plainfield, New Jersey, he lived in Belgium and France from age 1 to age 17, and received his BSEE degree from Rensselaer Polytechnic Institute in 1984. Craig is married. His hobbies include writing and recording industrial music, skiing, golf, ice hockey, softball, cooking, and wine.

Leith Johnson

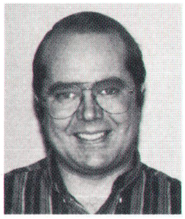


Leith Johnson received his BSEE degree in 1978 from the University of Nevada at Reno. He also joined HP's Calculator Products Division that year. He has done production engineering for the HP 9845 computer, processor and I/O design for HP 9000 computers, and high-end memory system design. For the HP 9000 Series 700 workstations, he was responsible for the memory and system bus controller architecture and design. Leith is an engineer/scientist with the HP Systems Technology Division. He is a coauthor of three technical papers and is listed as an inventor on seven pending patents, mainly on digital circuits and memory system design. In 1988 he received an MSCS degree from Colorado State University. He is a native of Seattle, Washington and enjoys snowboarding, boardsailing, golf, and mountain climbing.

Steven T. Mangelsdorf

Systems Technology Division R&D project manager Steve Mangelsdorf was the HP project manager for the VLSI floating-point coprocessor chip for the HP 9000 Series 700 computers. With HP since 1983, he has served as a design engineer and project manager for various VLSI chips used in PA-RISC processors, including bus interface units, cache controllers, and floating-point controllers. A member of the IEEE, he specializes in computer architecture, performance analysis, and VLSI design. He is a coauthor of three technical papers and is named as an inventor on a bus synchronizer patent. Born in El Paso, Texas, he received his BS degree in electrical engineering from Princeton University in 1982 and his MSEE degree from Stanford University in 1983. He enjoys hiking, bicycling, music, science fiction, and computer games.

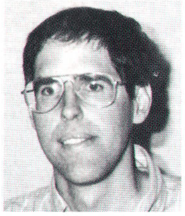
Thomas O. Meyer



Tom Meyer was project manager for the memory and system bus controller for the HP 9000 Model 720/730/750 computers. Now with the Systems Technology Division, he has been with HP since 1977. He has designed remote console firmware and memory expansion boards for the HP 250 computer and the power supply for HP 9000 Series 500 computers, and has served as project manager for the power supply and motherboard for the HP 9000 Model 825 computer and as project manager for the HP 9000 Model 835 computer. He grew up in Custer, South Dakota and attended the South Dakota School of Mines and Technology, graduating with a BSEE degree in 1977. He is married and enjoys sailing, woodworking, hiking, and skiing.

23 ECL Clock Circuits

Frank J. Lettang



Entry Systems Division development engineer Frank Lettang specializes in high-speed digital design, RF analog design, and integrated circuit design. With HP since 1980, he has contributed to the design of various HP 9000 Series 800 computers and to the clock system design of the HP 9000 Model 720/730/750 workstations. Frank grew up in Yorba Linda, California. He received his BS degree in electrical engineering and computer science from the University of California at Berkeley in 1980 and his MS degree in electrical engineering from Stanford University in 1983. He is married and has two daughters. In addition to his home and family, he likes hiking, camping, and bicycling.

26 Core I/O Subsystem

Daniel Li

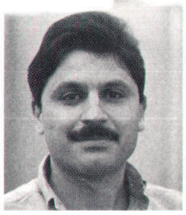
Daniel Li joined HP's Systems Technology Division in 1988 and was involved in the definition and development of the core I/O subsystem for the HP 9000 Model 720/730/750 computers. He's now a firmware designer at the Entry Systems Division. Daniel received his MSEE degree from the University of Wisconsin at Madison in 1981. Before coming to HP he worked on the 80386 processor design at Intel Corporation and developed a multiprocessor system and a vector processor for two startup companies. He was born in Taichung, Taiwan. He is married, has two sons, and enjoys tennis and skiing.

Audrey B. Gore

Author's biography appears elsewhere in this section.

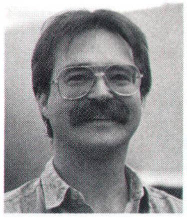
34 Design Verification

Ali M. Ahi



With HP since 1987, Ali Ahi has been involved with floating-point coprocessor design and system verification of the HP 9000 Series 700 workstations, first at the Information Technology Group and now at the Entry Systems Division. Before coming to HP, he designed special-purpose processors at the Fairchild Research Center. He holds BS and MS degrees in electrical and computer engineering from the University of California at Santa Barbara, and has been an instructor in the extension program of the University of California at Berkeley. He is married and enjoys reading, hiking, and racquetball.

Gregory D. Burroughs

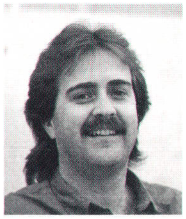


Greg Burroughs contributed to the design verification of the HP 9000 Series 700 computers. A native of San Francisco, California, he received his BS and MA degrees in mathematics in 1978 and 1979 from the University of California at Riverside and his MS degree in computer science from the University of Wisconsin at Madison in 1981. With HP since 1981, he has worked on computational geometry, VLSI design tools, microprocessor and floating-point design, validation, and testing, performance analysis of CPUs, floating-point coprocessors, and I/O systems, and workstation design verification. He is the author or coauthor of eight technical papers on CAD software, hardware performance, hardware testing and validation, and software engineering. Greg is a member of the Mathematics Association of America. His leisure activities include ballroom dancing, volleyball, and singing in a church choir.

Audrey B. Gore

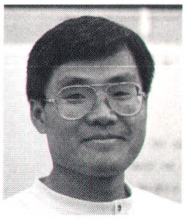
As a member of the system verification team for the HP 9000 Series 700 computers, Audrey Gore developed tests, analyzed simulation results, and maintained the model database. She received her BSEE degree from Brown University in 1988 and joined HP's Systems Technology Division as a hardware engineer, developing a bus analyzer card for the HP 9000 Model 815 computer. She is currently doing system verification at the Entry Systems Division. Audrey is originally from Tappan, New York. She plays oboe with the Santa Clara University orchestra and English horn with the Peninsula Symphony. She also plays acoustic guitar and enjoys writing and recording songs.

Steve W. LaMar



Steve LaMar has done design verification and testing for various floating-point, CPU, and system designs including the HP 9000 Series 700 workstations and the HP 9000 Model 815 CPU chip. A native of Fresno, California, he received his BS degree in computer science engineering from California State University at San Jose in 1985. He has been with HP since 1984 when he joined HP Laboratories, and is now with the Entry Systems Division. He is married, has a daughter, and enjoys skiing, tennis, and weightlifting.

Chi-Yen R. Lin



Robert Lin is a software development engineer with HP's Entry Systems Division. He was a member of the design verification team for the HP 9000 Series 700 workstations and for the HP 9000 Model 815 computer. Before joining the HP Systems Technology Division in 1987, he developed graphics system software for Daisy Corp. A native of Taiwan, he received his BS degree in fishery from the College of Marine Science and Technology. In 1983 he received MS degrees in oceanic science and computer-aided design from the University of Michigan. He is married and has one child.

Alan L. Wiemann

Entry Systems Division project manager Alan Wiemann has been working with PA-RISC systems since he joined HP Laboratories in 1984. He managed the system verification team for the HP 9000 Series 700 workstations. A native of North Dakota, he received his BSEE degree in 1980 from the University of Connecticut and his MSEE degree in 1981 from Stanford University. He has coauthored technical papers on computer architecture and processor design. Before coming to HP he designed digital telephone systems and components at Bell Laboratories. In addition to computer architectures and system verification, Alan's interests include scuba diving, hiking, bicycling, photography, gardening, and travel.

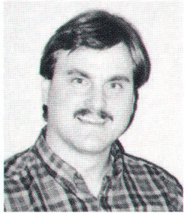
43 Model 720/730 Mechanical Design

John P. Hoppal

John Hoppal is a technical contributor with HP's Entry Systems Division. He received his BS degree in mechanical engineering from Colorado State University and joined HP in 1979. He has done mechanical design for several generations of

HP 9000 computers including the Series 500, the Models 825 and 835, and the Models 720 and 730. Two pending patents on the disk tray and EISA tray concepts for the Models 720 and 730 have resulted from his work. Born in Sterling, Colorado, John is married and has two sons. He enjoys bicycling, hiking, traveling, and his family.

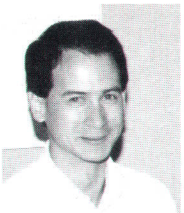
Arlen L. Roesner



A native of the Glacier Park area of Montana, Entry Systems Division mechanical product designer Arlen Roesner received his BSME degree from Montana State University and joined HP's Fort Collins Systems Division in 1984. He has done manufacturing engineering for the HP 9836 and HP 9000 Models 825, 835, and 845 computers and mechanical design for the HP 9000 Models 720 and 730 workstations. He is married, has one child, and enjoys fishing and woodworking.

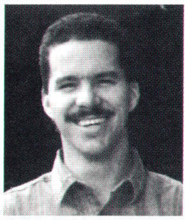
49 Series 700 Manufacturing

Kevin W. Allen



Kevin Allen received his BSME and MSME degrees in 1978 and 1979 from the University of California at Davis. After joining HP's Desktop Computer Division in 1979, he worked in R&D for four years and as a process engineer for five years, then became process engineering manager. In that capacity, he was responsible for implementation of the surface mount process for HP 9000 Series 700 computer manufacturing at HP's Colorado Computer Manufacturing Operation. A member of the International Society for Hybrid Microelectronics, he has coauthored several articles on surface mount pick-and-place equipment and surface mount process implementation. He is married, has two daughters, and plays softball and basketball.

Samuel K. Hammel



Kelley Hammel is a new-product engineer at HP's Colorado Computer Manufacturing Operation. For the HP 9000 Series 700 workstations, he was responsible for introduction to manufacturing, design for test, test strategy and implementation, and test controller and data collection system development. Before coming to HP, he was a design engineer for Mohican Enterprises. A native of Portsmouth, Virginia, he received his BSEE degree in 1988 from Ohio State University and his MSEE degree in 1991 from Stanford University. He is married, has a daughter, enjoys volleyball and model railroading, and serves as youth director at his church.

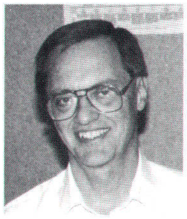
Anna Marie Hargis

Process development engineer Anna Hargis joined HP's Colorado Computer Manufacturing Operation in 1988. A native of Cincinnati, Ohio and a 1987 Ohio State University graduate with a BS degree in industrial and systems engineering, she had previously done manufacturing consulting with Arthur Anderson and Company. At HP, she has done process engineering for through-hole printed circuit board assembly, and equipment layout and process design for the surface mount production line for HP 9000 Series 700 workstations. She is a coauthor of an article on surface mount assembly.

Paul Roeber

As manufacturing engineering manager at HP's Exeter Computer Manufacturing Operation, Paul Roeber's contribution to the HP 9000 Series 700 workstations was the establishment of a new process for the release of software products. Paul came to Apollo Computer Systems in 1982 from Data General Corp., where he was software manufacturing engineering manager. A native of Boston, Massachusetts, he holds a BS degree in industrial engineering and an MBA degree, both from Northeastern University. He is married, has two children, and enjoys volleyball, travel, boating, and skiing. He also serves on the board of incorporators of the Franklin Children School and as head trustee for the Bay Street Parking Trust.

Spencer M. Ure



Spence Ure is manufacturing development engineering manager at HP's Colorado Computer Manufacturing Operation. As new product engineering manager for the HP 9000 Series 700 workstations, he was responsible for the worldwide manufacturing introduction of these products. Previously he had served in a similar capacity for the HP 9000 Models 825 and 835 and HP 3000 Series 925, 935, and 945 computers. With HP since 1965, when he joined the Colorado Springs Division, he also served as production engineering manager for the HP 9000 Series 500 and was in R&D for nine years. Born in Duluth, Minnesota, he received his BS degree in mechanical engineering from the University of Utah in 1965. Spence is married and has four children and one grandchild. He is involved in cub scouting, boy scouting, and church activities, and his interests include backpacking, photography, birding, military history, and railroading.

55 Series 700 Entry Level

John A. Dykstal

John Dykstal received his BSEE degree from the University of Wisconsin at Madison in 1988 and his MSEE degree from Stanford University in 1989. He's a hardware design engineer at HP's User Interface Technology Division, which he joined in 1989. He has contributed to the hardware design of the TurboVRX graphics subsystem and the graphics subsystem for the HP 9000 Models 705 and 710 workstations. John is a native of Madison, Wisconsin. His interests include skiing, softball, volleyball, and basketball.

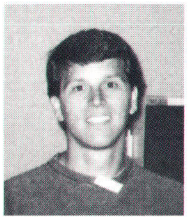
Craig R. Frink

Craig Frink is an engineer/scientist at the Advanced Systems Division of the HP Workstation Business Unit. He has served as system architect and lead designer for the Apollo DN3000, DN4000, DN3500, and DN4500, the HP Apollo DN5500, and the HP 9000 Models 705 and 710 computers. He has published papers on virtual cache architecture and design, system-level simulation, and workstation design. He joined Apollo Computer Systems in 1984 and holds a BSM degree in management from Lesley college. His interests include hockey, running, and archery.

Robert J. Hammond

User Interface Technology Division project manager Bob Hammond was project manager for the multimedia audio system for the HP 9000 Models 705 and 710 computers. He joined Apollo Computer Systems in 1987 and has been a software engineer and project manager for various Apollo graphics subsystems. Before joining Apollo, he was a software engineer with Compugraphic Corp. Bob was born in Allentown, Pennsylvania. He received his BS degree in biochemistry from the University of Connecticut in 1980 and studied electrical engineering and computer science at the University of New Hampshire. He is married, has three children, and enjoys hockey, basketball, scuba diving, and fishing.

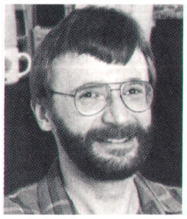
Don C. Soltis, Jr.



Don Soltis joined HP's Fort Collins Systems Division in 1985, served as a production engineer and as a manufacturing development engineer, and then became an R&D hardware designer. He contributed to the design of the I/O ASIC for the HP 9000 Models 425e and 475 computers and the voice-quality audio system for the HP 9000 Models 705 and 710 computers. He's now with the HP Measurement and Control Systems Division. Don received his BSEE and MSEE degrees from Colorado State University in 1984 and 1988. He is married and has three children. His interests include fly-fishing, bicycling, skiing, and camping.

64 Color Print Cartridge

Daniel A. Kearl



Dan Kearl is a senior architect at HP's Inkjet Components Division with interests in thin-film and microelectronic process development. He contributed to the DeskJet/DeskWriter color print cartridge architecture and testing. A native of Bellevue, Washington, he graduated from Washington State University in 1980 with a BS degree in physical metallurgical engineering and joined HP's Loveland Technology Center the same year. He has done process development for IC assembly and packaging, for thin-film deposition for both precision resistor networks and inkjet print cartridges, and for thermal inkjet barrier film imaging. He serves on advisory boards for local school and college technology programs, and has a special interest in "antique technology"—steam engines, gas engines, tractors, and hydropower—collecting, restoring, and running the "old iron", and collecting books of the era. His other interests include greenhouse gardening, sailing, old motor boats, and beer and wine making.

Michael S. Ard

Mike Ard is an R&D project manager at HP's Vancouver Division. He was project manager for development of the firmware and electronics for the DeskJet 500C and DeskWriter C printers. He received his BS and MS degrees in computer science from Brigham Young University in 1975 and 1978 and joined HP's Data Systems Division in 1976, moving to the Vancouver Division three years later. Mike was born in St. Anthony, Idaho. He is married and has seven children. He coaches youth basketball and is involved in church activities.

69 Ink & Cartridge Development

Craig Maze

Craig Maze helped develop the inks for the DeskJet/DeskWriter color print cartridge. With HP since 1978, he has also worked on liquid crystal displays for calculators, IC process development, and print cartridge architecture. He received his BS degree in chemical engineering from Purdue University in 1959 and his PhD in chemical engineering from Iowa State University in 1970. Before coming to HP, he worked on IC packaging and liquid crystal displays for Motorola and on aerospace materials engineering for Martin-Marietta. A member of the American Chemical Society and the American Institute of Chemical Engineers, he is named as an inventor on two patents related to liquid crystal displays.

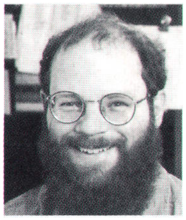
Loren E. Johnson

An ink chemist at HP's Inkjet Components Division, Loren Johnson helped develop the inks for the DeskJet/DeskWriter color print cartridge. He has also worked on inks for previous DeskJet and ThinkJet products, on etch process development for the ThinkJet printhead, on ThinkJet ink manufacturing, and on evaluation of thermal printer papers. He joined HP at the Vancouver Division in 1981. A native of Rapid City, South Dakota, he received his BS and MS degrees in chemistry from the South Dakota School of Mines and Technology in 1978 and 1980. Six patents on inkjet ink compositions have resulted from his work. Loren is married and has a daughter. His interests include hiking, fishing, computer gaming, xerobotany, drawing, science fiction, pottery, and cross-country skiing.

Daniel A. Kearl

Author's biography appears elsewhere in this section.

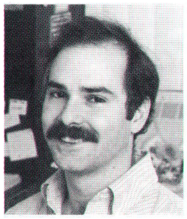
James P. Shields



Jay Shields is an R&D chemist at HP's Inkjet Components Division. With HP since 1988, he was on the ink development team for the DeskJet 500C/DeskWriter C color print cartridge. He received his BS degree in chemistry from Bradley University in 1982 and his PhD degree in analytical chemistry from Oregon State University in 1987. Before coming to HP, he was a research chemist at Dow Chemical. He has published several articles in the area of analytical plasma spectroscopy. Jay is married and enjoys woodworking, wine making, and a variety of outdoor sports.

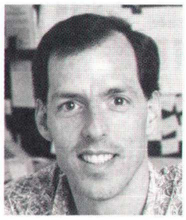
77 Automated Assembly

Lee S. Mason



Lee Mason is a mechanical engineer at HP's Inkjet Components Division, specializing in design for manufacturability, machine design, and automated assembly. Born in Pittsburgh, Pennsylvania, he received his BS degree in mechanical engineering from the University of California at Davis in 1981 and joined HP's Corvallis Division the same year, serving as an R&D and production engineer for the HP 75C handheld computer. From 1984 to 1987, he designed turnkey robotic assembly systems at Intelledex Inc. After returning to HP, he did tool design and process development for the DeskJet/DeskWriter black and color print cartridges. In 1992 he received an MBA degree from the University of Oregon. Lee is married and has two daughters. Now that his MBA studies are completed, he hopes to have more time for gardening, bicycling, water skiing, and family activities.

Mark C. Huth



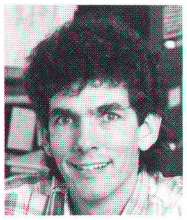
Manufacturing development engineer Mark Huth came to HP's Corvallis Division in 1981, shortly after receiving his BSME degree from the Virginia Polytechnic Institute. He has developed automated print cartridge manufacturing processes for the ThinkJet, DeskJet, and PaintJet printers, and did process development and tool design for the DeskJet/DeskWriter color print cartridge. His professional interests include automated manufacturing and machine vision. Mark is married and has two sons. He is a rock climber and shares interests in soccer and sumo wrestling with his sons.

84 Adhesive Technology

Douglas J. Reed

Doug Reed is a manufacturing development engineer at HP's Inkjet Components Division. For the DeskJet/DeskWriter color print cartridge, he worked in R&D on sealing surface development and in manufacturing on the adhesive dispense process and tooling. He also helped develop the adhesive dispense process and did tooling modification for the DeskJet black print cartridge. Doug received his BSME degree from Oregon State University in 1987 and joined HP the same year. He is a member of the ASME and the Society of Manufacturing Engineers. Born in Elmira, New York, he is married, has three children, and is expecting a fourth. He is involved in church leadership and enjoys fly-fishing, reading, and writing.

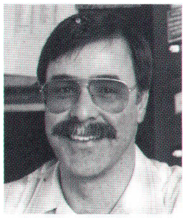
Terry M. Lambright



Now with HP's Inkjet Components Division, mechanical development engineer Terry Lambright joined the HP IC Business Division in 1984 as an incoming materials engineer, later working on the encapsulation of TAB circuits for HP 20 Series calculators. For the DeskJet/DeskWriter color print cartridge, his responsibilities included liaison with the printer development team, adhesive selection, and TAB circuit design. Before coming to HP he worked for five years in the nuclear industry on materials decontamination and consolidation, and three years in the aerospace industry on the manufacture of titanium tubing. He's the author of a paper on titanium alloy tube fabrication. Born in Galion, Ohio, he attended the University of Arizona, receiving a BS degree in metallurgical engineering in 1974 and an MS in materials science in 1976. He is married, has two children, is involved in church leadership, and holds the rank of captain in the U.S. Air Force Reserve. His interests include marquetry, woodcarving, reading, and working with international students.

87 Machine Vision

Michael J. Monroe



Mike Monroe is a manufacturing development engineer at HP's Inkjet Components Division. A specialist in optical design, machine vision, and control systems, he joined the HP Corvallis Division in 1979. He has served as a production and service engineer for desktop computers and as an electronic tooling engineer for portable computers and handheld calculators, and developed the machine vision systems for DeskJet/DeskWriter color print cartridge production. Before joining HP he worked on the Viking Mars probe at the Jet Propulsion Laboratory (NASA) and on electronic systems for missile launching submarines. He was also associated with a computer design consulting firm. He received his BSE degree in 1971 from the University of California at Los Angeles, specializing in electronics engineering. A member of the IEEE, he was born in Salt Lake City, Utah and served in the Air National Guard for six years, attaining the rank of sergeant. He is married, has two daughters, and enjoys making telescopes, metalworking, and woodworking.

93 DeskWriter C Driver

William J. Allen

Will Allen was one of the developers of the software drivers for the DeskJet 500C and DeskWriter C color inkjet printers. Born in Lafayette, Indiana, he received his BS degree in computer science from Purdue University in 1982. After joining HP in Colorado in 1983, he developed firmware for the HP 16500 and 1630G logic analyzers, then left HP briefly in 1987 to join a startup company in Oregon, where he developed automated test equipment software. Rejoining HP a few months later at the Vancouver Division, he served as a product support engineer, and in 1989 moved to the R&D lab, where he has worked on printer motion control firmware and software drivers. Will is married, has three children, and coaches a youth soccer team. His other interests include travel, bicycling, swimming, table tennis and track work at auto races.

Toni D. Courville

Software development engineer Toni Courville received her BS degree in computer science from Portland State University in 1988. With HP's Vancouver Division since 1988, she has worked on firmware for the DeskJet 500 and DeskWriter printers, and helped develop the driver for the DeskWriter C. She is a native of Boise, Idaho and enjoys bicycling, skiing, and other outdoor activities.

Steven O. Miller

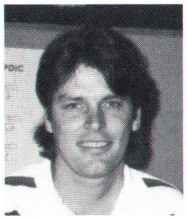
Steven Miller has done software and firmware engineering for HP's Vancouver Division since 1985, contributing to the firmware and driver design of the RuggedWriter, DeskWriter, and DeskWriter C printers. A native of Bellevue, Washington, he received BS degrees in computer engineering and computer science from Oregon State University in 1985. A patent on a data compression scheme names him as an inventor. Steven is married, has two children, and enjoys boardsailing, golf, skiing, and white-water rafting.

103 MRP Action Manager

Alvina Y. Nishimoto

An engineer/scientist at HP's Software Technology Division, Alvina Nishimoto is responsible for the development, enhancement, and support of manufacturing software products. She joined HP in 1978 after receiving both her BS and MS degrees in industrial engineering from Stanford University that same year. Her professional interests include software reuse and software development processes. Alvina was born in Honolulu, Hawaii.

William J. Gray



A development engineer on the HP MRP Action Manager project at HP's Manufacturing Productivity Division (MPD), Bill Gray is currently a development engineer at HP's Professional Services Division. He joined MPD in 1983 after receiving a BS in computer science from North Carolina State University that same year. Bill was born in Macon, Georgia, is married, and has one child. His recreational activities include golf, bowling, soccer, and racquetball.

Barbara J. Williams

Currently a software manufacturing engineering manager at HP's Scientific Instruments Division, Barbara Williams was the project manager for the HP MRP Action Manager project at HP's Manufacturing Productivity Division. Barb joined HP's Computer Systems Division in 1982. She worked as a development engineer on the HP PM/3000 and HP MM/3000 products and as a project manager for the A.02.00 release of the HP Purchasing enhancements project. While attending college she worked as a software engineer in a summer intern program at Bell Laboratories. She has a BS degree (1982) in computer science from Washington State University and she is certified in production and inventory management (CPIM) by the American Production and Inventory Control Society. Managing people, processes, and projects are her main professional interests. She served as a United Way loaned executive for HP in 1991. Born in Wenatchee, Washington, she is married and expects her first child in the summer. Sewing, aerobics, swimming, tennis, camping, waterskiing, gardening, and volleyball are among her recreational activities.

Hewlett-Packard Company, P.O. Box 51827
Palo Alto, CA 94303-0724

HEWLETT-PACKARD
JOURNAL

August 1992 Volume 43 • Number 4

**Technical Information from the Laboratories of
Hewlett-Packard Company**

Hewlett-Packard Company, P.O. Box 51827
Palo Alto, California, 94303-0724 U.S.A.

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

CHANGE OF ADDRESS: To subscribe, change your address, or delete your name from our mailing list, send your request to Hewlett-Packard Journal, P.O. Box 51827, Palo Alto, CA 94303-0724 U.S.A. Include your old address label, if any. Allow 60 days.